

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Analisis Penelitian Terdahulu

Tabel di bawah ini menyajikan ringkasan referensi penelitian sebelumnya yang mengaplikasikan metode *Convolutional Neural Network* (CNN) dari Algoritma *You Only Look Once* (YOLO).

Tabel 2.1 Tinjauan Referensi Penelitian Terdahulu.

NO	PENULIS	TAHUN	METODE	JUDUL	HASIL
1	Ke Sun, Yu-jie Zhang, Si-yuan Tong, Chang- bao Wang (Anhui Normal University)	2022	YOLO-v5 (You Only Look Once)	Study on Rice Grain Mildewed Region Recognition Based on Microscopic Computer Vision and YOLO-v5 Model	Penelitian ini bertujuan untuk mengembangkan metode deteksi butiran beras yang terinfeksi jamur dengan cepat dan tanpa merusak. Pertama, sampel mikrograf butiran beras yang terkontaminasi oleh <i>Aspergillus niger</i> , <i>Penicillium citrinum</i> , dan <i>Aspergillus cinerea</i> diambil, dan area yang terinfeksi ditandai. Berdasarkan struktur mikro yang dibuat, tiga model YOLO v5 dikembangkan untuk

					<p>mengidentifikasi area yang terkontaminasi oleh <i>Aspergillus niger</i>, <i>Penicillium citrinum</i>, dan <i>Aspergillus cinerea</i>. Selanjutnya, hubungan antara rasio luas jamur dan jumlah koloni dianalisis. Hasil penelitian menunjukkan bahwa model YOLO-v5 yang diusulkan mencapai akurasi sebesar 89,26%, 91,15%, dan 90,19% dalam mendeteksi area yang terinfeksi oleh <i>Aspergillus niger</i>, <i>Penicillium citrinum</i>, dan <i>Aspergillus cinerea</i> pada mikrograf set validasi. Proporsi area yang terkontaminasi oleh ketiga jenis jamur tersebut secara logaritma berkorelasi dengan logaritma jumlah koloni (TVC), dengan koefisien determinasi masing-masing adalah 0,7466, 0,7587, dan 0,8148. Hasil penelitian ini</p>
--	--	--	--	--	---

					memberikan kontribusi sebagai referensi untuk penelitian lebih lanjut mengenai metode deteksi cepat jenis beras berbasis teknologi MCV.
2	Muhammad Juman Jhatial, dkk. (Sukkur IBA University)	2022	YOLO-v5 (You Only Look Once)	Deep Learning – Based Rice Leaf Diseases Detection Using YOLO-v5	<p>Pertanian padi memiliki peran sentral dalam ekonomi Pakistan sebagai penyedia kebutuhan pangan. Daun tanaman padi sering kali diserang oleh berbagai penyakit seperti busuk daun bakteri, bercak coklat, hawar daun, dan busuk busuk. Dalam penelitian ini, kami berusaha membangun model deteksi daun padi yang sederhana dan optimal menggunakan YOLOv5, sebuah model deep learning yang telah diperbarui ke versi terbaru YOLO.</p> <p>Performa dan akurasi deteksi target YOLOv5 terbukti lebih unggul dibandingkan dengan YOLOv3 dan YOLOv4.</p>

					<p>Meskipun sistem deteksi ini memiliki banyak aplikasi, penelitian ini secara khusus berfokus pada pertanian dengan tujuan meningkatkan produktivitas tanaman melalui deteksi dini penyakit tanaman, terutama penyakit daun pada tanaman padi. Model ini berhasil mengidentifikasi dan mendeteksi penyakit pada daun padi dengan menggunakan dataset gambar daun yang diunduh dari platform Kaggle, terdiri dari 400 gambar daun yang terinfeksi penyakit. Pada penelitian ini, Google Colaboratory digunakan sebagai platform untuk melatih, memvalidasi, dan menguji model deteksi penyakit daun padi. Semua langkah yang diperlukan dilakukan untuk memastikan</p>
--	--	--	--	--	--

					identifikasi dan deskripsi penyakit daun padi yang memadai. Hasil eksperimen menunjukkan bahwa model deep learning yang dikembangkan dengan melakukan 100 siklus pelatihan memberikan kinerja terbaik, dengan nilai presisi, recall, dan mAP masing-masing mencapai 1.00, 0.94, dan 0.62.
3	Nova Eka Budiyanta, dkk. (Universitas Katolik Indonesia Atma Jaya)	2021	YOLO (You Only Look Once)	Sistem Deteksi Kemurnian Beras Berbasis Computer Vision	Penelitian ini bertujuan untuk implementasi sistem deteksi kemurnian beras pada campuran pengotor sebagai parameter kualitas beras dalam proses pengendalian kualitas. Sistem yang dikembangkan berbasis computer vision dengan kamera sebagai sensor utamanya. Data gambar yang diperoleh dari kamera diolah untuk mengidentifikasi beras bersih dan beras yang

					<p>terkontaminasi dengan pengotor. Pengidentifikasian dilakukan menggunakan metode You Only Look Once (YOLO) v3. Secara keseluruhan, sistem pengenalan objek ini berhasil. Proses pelatihan model mampu mengurangi nilai loss, dengan nilai loss mencapai 1,89 pada iterasi ke-1.000 dan 0,16 pada iterasi ke-15.000, sementara pengulangan diminimalkan secara signifikan. Dalam pengujian menggunakan proses observasi, model ini memberikan hasil yang baik dengan akurasi rata-rata sebesar 86,11%.</p>
4	<p>Alfath Daryl Alhajir (Universitas Pembangunan Nasional “Veteran”)</p>	2021	<p>YOLO (You Only Look Once)</p>	<p>Implementasi Metode Object Detection dengan Algoritma YOLO (You</p>	<p>Standar SNI-6128: Tahun 2020 merinci kualitas beras ke dalam tiga kategori, dengan mempertimbangkan komponen mutu beras seperti beras utama, butir pecah, menir,</p>

				<p>Only Look Once) Untuk Deteksi Beras dan Objek Asing</p>	<p>butir merah/hitam, butir rusak, butir berkapur, benda asing, dan komponen mutu gabah. Beras utama diidentifikasi sebagai komponen mutu yang diinginkan, sedangkan sisanya dianggap tidak diinginkan. Adanya kebutuhan mendesak untuk mengembangkan sistem atau algoritma yang dapat memisahkan beras dari komponen yang tidak diinginkan menjadi jelas. Pengenalan objek, sebagai tugas pembelajaran mesin, mampu mengenali dan mengklasifikasikan objek dalam gambar, dan dapat diimplementasikan dalam sistem pemisahan beras. Algoritma YOLO, yang berbasis Convolutional Neural Network (CNN), dipilih karena sederhana dan dapat dengan mudah diadaptasi ke berbagai</p>
--	--	--	--	--	--

					<p>kasus penggunaan. Algoritma ini difokuskan pada pemrosesan gambar atau visi komputer. Dalam penelitian ini, implementasi algoritma sebagai sistem deteksi diskriminasi beras berhasil mencapai akurasi rata-rata 0,9015 mAP pada dataset pelatihan dan 0,5944 pada dataset pengujian.</p> <p>Pemanfaatan metode seperti augmentasi data dan pencarian grid secara signifikan meningkatkan kinerja model.</p>
5	Kiki Ahmad Baihaqi, dkk. (Universitas Buana Perjuangan Karawang)	2021	YOLO-v3 (You Only Look Once)	Application of Convolutional Neural Network Algorithm for Rice Type Detection Using YOLO-v3	<p>Beras, sebagai makanan pokok yang kaya energi, memiliki berbagai jenis yang umumnya dijual di toko beras, termasuk beras IR42, beras pera, beras gluta, dan beras pandan wangi.</p> <p>Namun, banyak masyarakat, terutama kaum milenial, masih kurang familiar dengan keempat jenis beras tersebut.</p>

					<p>Oleh karena itu, penelitian ini bertujuan untuk memperkenalkan warung nasi sebagai upaya untuk mengurangi kekecewaan para pedagang beras. Metode yang digunakan dalam penelitian ini adalah YOLO (You Only Look Once) v3 untuk mengidentifikasi varietas padi. Implementasi pengenalan citra menggunakan YOLO v3 diuji dengan 12 pola, mencakup berbagai objek citra digital seperti 4 jenis beras, 4 butir beras, dan 3 jenis beras yang berbeda. Hasil pengujian menunjukkan bahwa tercapai tingkat keberhasilan 100% dalam mengidentifikasi objek citra tersebut.</p>
6	Alfath Daryl Alhajir, dkk. (Universitas Pembangunan	2021	CNN (Convolutional Neural	Sistem Pendeteksian Objek Beras dan Benda	Standar SNI 6128:2020 mengklasifikasikan kualitas beras ke dalam tiga kelas berdasarkan komponen-

	<p>Nasional “Veteran”)</p>		<p>Network) Berbasis Keras</p>	<p>Asing Berbasis Keras dan Google Colab</p>	<p>komponen mutu seperti beras kepala, butir patah, butir menir, butir merah/hitam, butir rusak, butir kapur, benda asing, dan butir gabah. Beras kepala diidentifikasi sebagai komponen mutu yang paling diinginkan, sementara sisanya dianggap tidak diinginkan. Oleh karena itu, diperlukan suatu sistem yang dapat memisahkan beras dari komponen yang tidak diinginkan. Penelitian ini mengembangkan sistem berbasis deep learning untuk mendeteksi beras dan benda asing dengan mengimplementasikannya menggunakan framework deep learning keras melalui platform Google Colaboratory dengan menggunakan bahasa pemrograman Python. Sistem ini diharapkan dapat</p>
--	--------------------------------	--	--	--	---

					<p>memberikan performa terbaik dan nantinya dapat diintegrasikan sebagai subsistem rekognisi dalam sistem pemisahan beras. Subsistem ini menerima input dari sensor kamera dan menghasilkan informasi yang berguna untuk diproses oleh subsistem aktuator. Hasil penelitian ini adalah sistem yang mampu mendeteksi objek dalam citra dengan kategori beras dan benda asing.</p>
7	<p>Siti Rahmah Danur Amiril (Universitas Islam Indonesia)</p>	2020	<p>CNN (Convolut ional Neural Network)</p>	<p>Implementasi Algoritma Convolutional Neural Network Pada Klasifikasi Penyakit Padi Melalui Citra Daun</p>	<p>Padi atau nama lain dalam Bahasa latin nya adalah Oryza Sativa merupakan salah satu tanaman budi daya terpenting didalam peradaban khususnya di negara Indonesia. Di negara ini padi merupakan komoditas utama bagi kehidupan dalam pemenuhan kebutuhan bahan pokok pangan. Menurut data dari Badan Pusat Statistik</p>

					<p>(BPS) pada tahun 2020 terkait jumlah penduduk Indonesia saat ini yang mencapai 269.600.000 jiwa, secara tidak langsung membuat kebutuhan akan tanaman ini juga semakin meningkat. Proses penanaman pada padi biasanya dihadapkan dengan berbagai kendala baik dari luar (hama) maupun kendala dari dalam (penyakit). Penyakit yang paling sering menyerang tanaman padi di Wilayah Indonesia ini adalah Bacterial Leaf Blight, Brown Spot dan Leaf Spot. Oleh sebab itu, penelitian ini bertujuan untuk mengklasifikasi penyakit padi berdasarkan citra daunnya yang telah terserang penyakit menggunakan teknik Deep Learning dengan metode Convolutional Neural Network (CNN), untuk memudahkan</p>
--	--	--	--	--	--

					<p>dalam melakukan pendeteksian dini terhadap penyakit padi agar tidak menular dan menjadi semakin parah. Metode CNN mampu menghasilkan tingkat akurasi yang signifikan karena memiliki kedalaman jaringan dan telah banyak di aplikasikan pada data gambar. Berdasarkan dari hasil klasifikasi yang dilakukan, didapatkan tingkat akurasi sebesar 91,7% dengan arsitektur terbaik menggunakan parameter scenario perbandingan dataset 90% size 100x100 piksel, kernel 3x3, learning rate 0,01, optimizer Adam, epoch 150 dan batch size 30.</p>
8	N. Prakash, R Rajakumar, N. Leela Madhuri, M. Jyothi, A.	2022	Algoritma Convoluti onal Neural Network	Image Classificatio n for Rice Varieties Using	Padi adalah tanaman dengan pertumbuhan tercepat di India dan seiring bertambahnya populasi, permintaan beras

	<p>Pavitra Bai, M. Manjunath, K. Gowthami</p>		<p>(CNN)</p>	<p>Deep Learning Models</p>	<p>juga meningkat. Sebagian besar negara Asia menanam beras dan mengekspornya ke seluruh dunia. Bergantung pada budaya makanan masyarakat, berbagai jenis beras dibudidayakan. Pada saat yang sama, kualitas makanan menjadi prioritas, jadi kami menggunakan teknik computer vision untuk mengekstraksi karakteristik kualitas beras. Produk dianalisis sifat fisiknya menggunakan teknik image processing seperti Visual Geometry Graph (VGG16) dan Vanilla CNN (alias Vanilla Neural Network) untuk mengidentifikasi fitur tekstual dan fitur gambar gabah beras. VGG16 terdiri dari arsitektur CNN 16 lapis. Itu dapat melatih jutaan set data dan mencapai tingkat akurasi tertinggi. Selain itu, model lain</p>
--	---	--	--------------	---------------------------------	---

					<p>yaitu vanilla neural network merupakan perluasan dari model regresi linier . Namun, vanilla CNN memiliki lapisan tersembunyi ekstra antara input dan output untuk membantu perhitungan ekstra. Teknik pemrosesan gambar telah digabungkan dengan jaringan saraf untuk memberikan akurasi yang lebih tinggi dalam model pelatihan daripada proses manual. Varietas beras Jasmine, Basmati, Arborio, Ipsala, dan Karacadag adalah lima jenis citra beras berbeda yang digunakan untuk klasifikasi. Masingmasing varietas ini memiliki 15.000 gambar, dimana total 75.000 gambar digunakan dalam pelatihan dan pengujian. Pengklasifikasi citra terbaik dipilih berdasarkan akurasi</p>
--	--	--	--	--	---

					terbaik. Hasil model yang diusulkan mempertimbangkan kinerja yang lebih baik dalam menentukan varietas padi.
--	--	--	--	--	--

2.1.2 Analisis Penelitian Saat Ini

Setelah mencermati beberapa penelitian terdahulu mengenai metode dan objek yang digunakan dalam penelitian ini, maka dapat dilihat perbedaan antara penelitian ini dengan penelitian sebelumnya yang disajikan pada tabel dibawah ini.

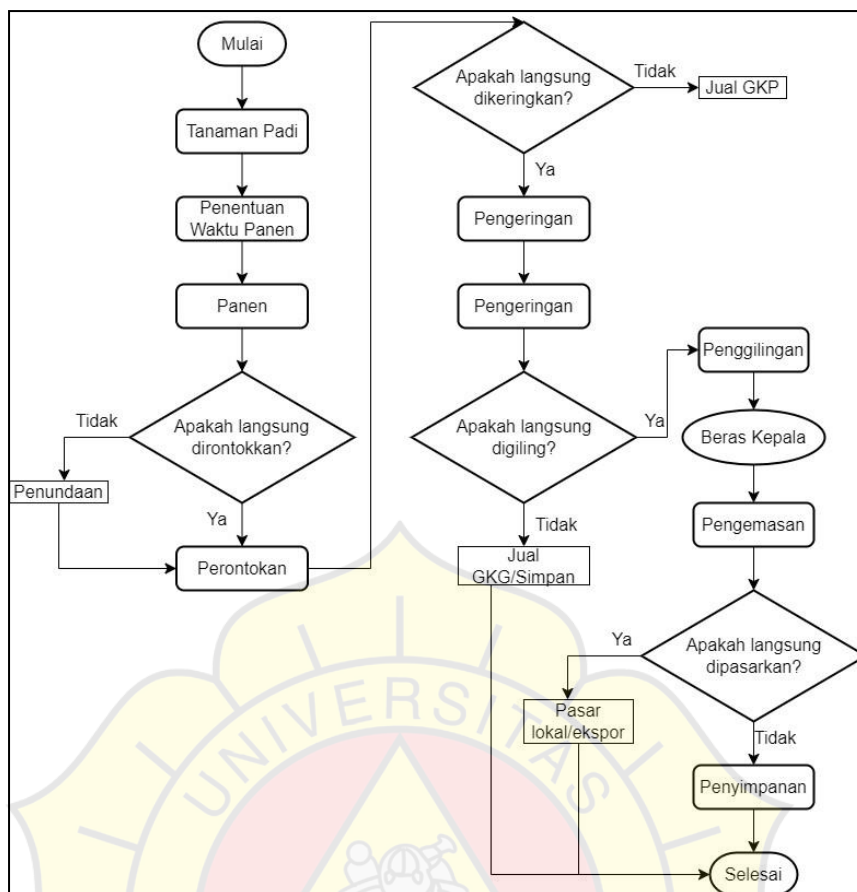
Tabel 2.2 Tinjauan Penelitian Saat Ini.

NO	PENULIS	TAHUN	METODE	JUDUL	HASIL
1	Shilvi Yanti Safitri	2023	CNN (Convolutional Neural Network) Arsitektur YOLO (You Only Look Once)	Sistem Identifikasi Deteksi Beras dan Objek Asing Menggunakan Metode Convolutional Neural Network (CNN)	Pada penelitian ini, pelatihan model dilakukan dengan menggunakan 30 epochs, batch size 10, dan ukuran gambar 640 piksel. Total dataset terdiri dari 1000 gambar, yang dibagi menjadi 500 gambar untuk pelatihan, 250 gambar untuk validasi, dan 250 gambar untuk pengujian. Terdapat lima kelas objek deteksi, yaitu butir

					<p>beras, butir batu, butir gabah, hewan kutu, dan butir merah/hitam. YOLOv5 mencapai kinerja terbaik dengan akurasi sebesar 95%, menggunakan nilai ambang batas (threshold) sebesar 0.6. Hasil tersebut menghasilkan Confusion Matrix 1.00, skor F1-Confidence Curve 0.795, Presisi Confidence Curve 0.953, Presisi Recall Curve 0.995 mAP @0.5, dan Recall-Confidence Curve 1.00.</p>
--	--	--	--	--	---

2.2 Proses Pascapanen Padi

Menurut riset yang dilakukan oleh Sigit Nugraha pada tahun 2018, aktivitas pascapanen melibatkan serangkaian kegiatan seperti pemanenan, perontokan, pengangkutan, pengeringan, penggilingan, penyimpanan, dan pemasaran. Kerugian hasil yang kritis terjadi terutama pada tahap pemanenan, penumpukan hasil panen padi sementara, dan proses perontokan padi untuk menghasilkan gabah. Evaluasi tingkat kerugian hasil pada kegiatan pascapanen di tiga ekosistem menunjukkan penurunan yang signifikan, yaitu dari 20,51% menjadi tingkat susut hasil sebesar 13,35% pada ekosistem lahan irigasi.



Gambar 2.1 Alur Proses Penanganan Pascapanen Padi

Berdasarkan penelitian yang dilakukan oleh Sri Lestari dan kolega pada tahun 2021, data dari Biro Pusat Statistik pada tahun 1996 menunjukkan bahwa kehilangan total hasil (susut bobot) dari berbagai tahap panen dan pascapanen padi setelah panen masih mencapai angka sebesar 20,42%.

Tabel 2.3 Susut Bobot Tahapan Panen dan Pascapanen Padi.

No	Tahapan Kegiatan	Susut Hasil (%)
1	Pemanenan	9,52
2	Perontokan	4,78
3	Pengangkutan	0,19
4	Pengeringan	2,13
5	Penggilingan	2,19
6	Penyimpanan	1,61
	Total	20,42

Klasifikasi kelas mutu beras telah ditetapkan berdasarkan peraturan Menteri Pertanian RI Tahun 2017, sebagaimana tertera dalam **Tabel 2.4**.

Tabel 2.4 Klasifikasi Kelas Mutu Beras (Menteri Pertanian, 2017)

No	Komponen Mutu	Satuan	Kelas Mutu	
			Medium	Premium
1	Derajat Sosoh (minimal)	%	95	95
2	Kadar Air (maksimal)	%	14	14
3	Beras Kepala (minimal)	%	75	85
4	Butir Patah (maksimal)	%	25	15
5	Total butir beras lainnya (maksimal), terdiri atas butir menir, merah, kuning/rusak, kapur	%	5	0
6	Butir Gabah (maksimal)	(Butir/100g)	1	0
7	Benda lain (maksimal)	%	0,05	0

Agar dapat menetapkan klasifikasi kelas mutu beras sesuai dengan penjelasan di atas, diperlukan pengolahan panen dan pascapanen produk beras yang hati-hati dan akurat. Hal ini bertujuan untuk memastikan bahwa hasil kerja sesuai dengan standar kelas mutu beras premium. Oleh karena itu, disarankan untuk melaksanakan langkah-langkah terencana guna meningkatkan proses pengolahan tanaman padi yang ada agar dapat ditingkatkan.

2.2.1 Parameter Pengukuran

Setiap contoh butiran beras, kerikil (batu), butiran gabah, hewan (kutu), dan butiran beras merah/hitam yang diambil sejumlah 100 gram dengan dilakukan 10 kali percobaan atau ulangan. Proses pertama melibatkan penggunaan gambar secara individual dari setiap sampel data untuk melatih algoritma YOLO-v5. Sebagai langkah pengujian, terdapat 10 gambar sampel dan 10 video sampel yang digunakan untuk menguji kemampuan algoritma dalam mengenali objek atau tidak. Setelah tahapan ini berhasil diselesaikan, pengujian dilakukan secara real-

time menggunakan kamera web.

2.2.2 Dokumentasi Proses Panen dan Pascapanen Padi

Pada tanggal 29 Maret 2022, penulis melakukan observasi penelitian pada Bapak Hj. Burhan, tempat beras atau padi diolah, dipanen dan digiling disana dengan mesin yang masih sederhana untuk selanjutnya diperdagangkan oleh penduduk/masyarakat.



Gambar 2.2 Sawah dan Hasil Panen Padi

Pada saat observasi penulis melakukan pengambilan gambar lahan pertanian Bapak Hj Burhan, melihat para petani sedang bergotong royong mengerjakan pekerjaannya.



Gambar 2.3 Proses Pengangkutan Padi Ke Dalam Mesin Giling

Selanjutnya petani melakukan pengangkutan padi ke dalam mesin penggiling untuk dilakukan proses penggilingan agar menjadi gabah yang akan dikeringkan atau dijemur dibawah sinar matahari langsung.



Gambar 2.4 Proses Pemisahan Padi

Pada gambar dibawah ini merupakan petani yang sedang mengeringkan gabah yang telah selesai digiling menggunakan mesin penggiling untuk dimasukkan kedalam karung, lalu diantar ketempat penggilingan gabah menjadi beras.



Gambar 2.5 Proses Pengeringan Gabah

Setelah itu, padi yang sudah dimasukkan kedalam karung dibawa oleh petani menggunakan motor untuk selanjutnya di proses penggilingan gabah menjadi beras.



Gambar 2.6 Proses Pengangkutan Gabah Ketempat Penggilingan

Setelah sampai ketempat produksi beras, tugas selanjutnya yaitu pegawai beras melakukan penggilingan gabah padi menjadi beras untuk selanjutnya diperjualberikan kepada toko agen beras.



Gambar 2.7 Proses Penggilingan Gabah Menjadi Beras

Pada gambar dibawah merupakan proses memasukkan beras yang sudah digiling kedalam karung penyimpanan, untuk sisa gabah penggilingan bisa

dijadikan dedak atau sekam yang dapat bermanfaat bagi masyarakat serta mendatangkan keuntungan bagi pemilik penggilingan dan para petani.



Gambar 2.8 Tempat Penyimpanan Beras

2.3 Beras

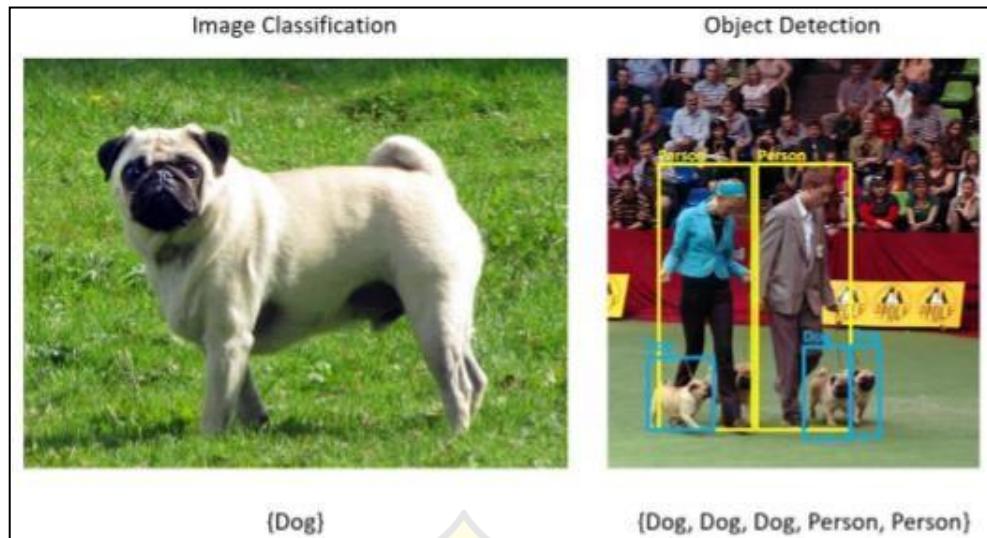
Menurut penelitian yang dilakukan oleh Alfath Daryl Alhajir dan kolega pada tahun 2021, beras merupakan hasil utama dari proses penggilingan padi jenis *Oryza Sativa*. Beras tersebut terbentuk dari seluruh lapisan sekam yang telah terkelupas, bersama dengan seluruh atau sebagian lembaga, lapisan dedak, dan bekatul yang telah dipisahkan. Bentuk beras dapat berupa butir beras utuh, beras kepala, beras patah, atau menir, sesuai dengan Standar Nasional Indonesia (SNI) tahun 2020. Di Indonesia, beras memiliki peran krusial sebagai sumber makanan utama, dengan total produksi mencapai 314,10 ribu ton pada tahun 2020 menurut Badan Standardisasi Nasional (BSN).

Proses pengolahan beras dimulai dari gabah padi, yaitu butir padi yang sekamnya belum terkelupas. Proses ini melibatkan tahap-tahap pembersihan, pengeringan, penggilingan, dan pemolesan untuk menghasilkan beras sosoh, sesuai dengan standar SNI tahun 2020. Meskipun beras sudah dipisahkan dari sekam, dedak, dan bekatul, kemungkinan masih ada impuritas atau benda asing

seperti kerikil, beras hitam, beras gabah, atau beras patah yang mungkin terbentuk atau tidak terproses pada tahap sebelumnya. Oleh karena itu, beras dapat menjalani proses pemrosesan tambahan melalui berbagai metode, seperti proses color sorting yang memisahkan beras berdasarkan warna, proses destoning yang memisahkan beras dari batu, atau proses grading yang memisahkan beras berdasarkan ukuran, sesuai dengan standar SNI tahun 2020.

2.4 *Object Detection*

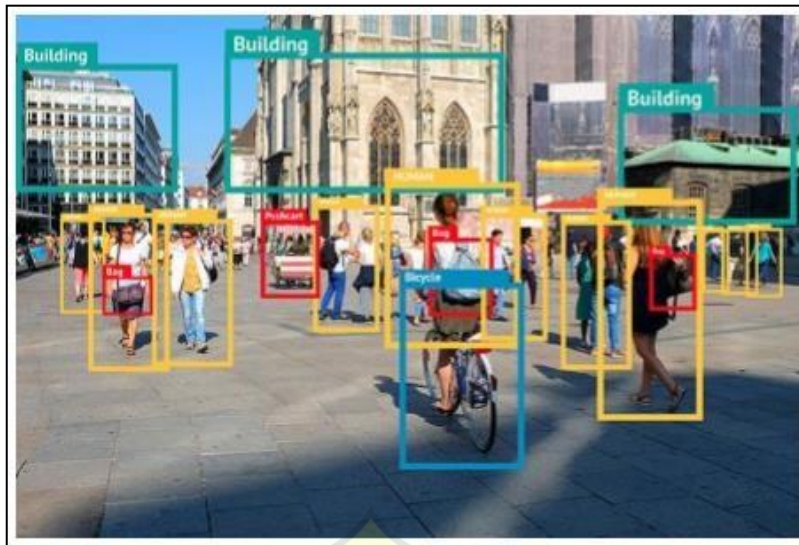
Menurut Haris Nasrullah M. pada tahun 2022, deteksi objek, atau yang dikenal sebagai object detection, merupakan suatu konsep dalam computer vision yang merupakan perkembangan dari klasifikasi gambar. Tujuan dari deteksi objek adalah untuk menemukan dan mengklasifikasikan objek-objek yang terdapat dalam sebuah gambar atau video. Dibandingkan dengan klasifikasi, deteksi objek memiliki kemampuan untuk mendeteksi lebih dari satu objek secara simultan dan memberikan informasi mengenai lokasi objek dalam gambar. Sebuah detektor objek akan menghasilkan daftar objek yang berhasil terdeteksi, disertai dengan informasi tentang kelas objek, probabilitas, dan koordinat objek untuk masing-masing objek yang terdeteksi (Vasilev dkk, 2019; Zhao dkk, 2018).



Gambar 2.9 Perbedaan Klasifikasi dan Deteksi Objek

Sumber : *Detect stop signs in images with model builder-learn.microsoft.com*

Pada **Gambar 2.9**, Klasifikasi hanya mampu mengidentifikasi kelas suatu objek pada gambar, sementara dalam deteksi objek, banyak objek dapat terdeteksi sekaligus dengan kelas yang bervariasi. Dalam proses deteksi objek, detektor objek akan memberikan daftar objek yang terdeteksi, mencakup informasi mengenai kelas objek (seperti sendok, laptop, ponsel, atau benih) serta probabilitas atau skor keyakinan dalam rentang 0 hingga 1. Skor ini mencerminkan seberapa yakin detektor objek dalam menentukan keberadaan dan lokasi objek yang terdeteksi. Selain itu, detektor juga menyediakan informasi mengenai koordinat wilayah di mana objek terdeteksi, ditampilkan dalam bentuk kotak pembatas yang disebut sebagai bounding box, sebagaimana terlihat pada **Gambar 2.10**.



Gambar 2.10 Bounding Box Pada Deteksi Objek

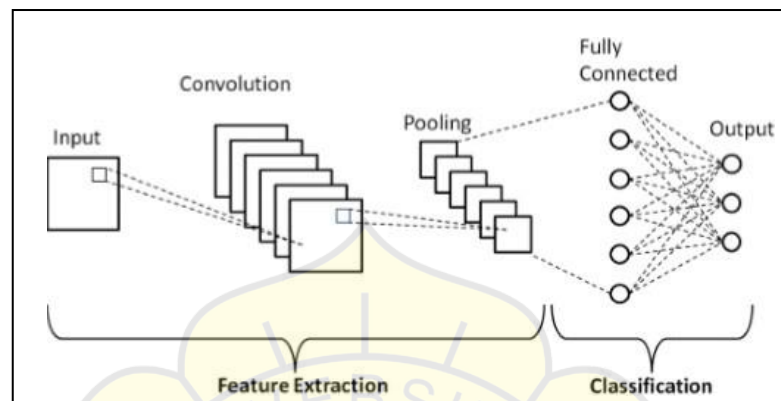
Sumber : *Object Detection Technology – How it Works and Where is it Used?* – DataScienceCentral.com

2.5 *Convolutional Neural Network (CNN)*

Menurut Haris Nasrullah M. pada tahun 2022, *Convolutional Neural Network (CNN)* merupakan salah satu algoritma yang paling populer yang digunakan dalam bidang *Deep Learning*. CNN merupakan suatu metode yang berasal dari *Multi-Layer Perceptron (MLP)* dan dirancang untuk memproses data dalam bentuk grid dua dimensi. Penggunaan CNN umumnya ditujukan untuk mengklasifikasikan data yang telah diberi label menggunakan pendekatan pembelajaran berbasis supervisi. Algoritma ini terbukti sangat efektif dalam menemukan pola dalam gambar dan mengidentifikasi objek yang terdapat di dalamnya.

Secara umum, CNN terdiri dari dua tahap utama, yaitu ekstraksi fitur atau feature learning dan klasifikasi dengan menggunakan lapisan fully connected (Ghosh dkk., 2020). Pada tahap ekstraksi fitur, gambar input mengalami proses ekstraksi untuk mempelajari nilai-nilainya. Nilai-nilai ini kemudian diubah

menjadi vektor dan diteruskan ke tahap klasifikasi. Pada tahap klasifikasi, model jaringan saraf CNN akan digunakan untuk mengklasifikasikan objek berdasarkan kelasnya. Komponen utama dalam model CNN melibatkan lapisan konvolusi, lapisan pooling, fungsi aktivasi, dan lapisan fully connected (Indolia dkk., 2018).



Gambar 2.11 Arsitektur *Convolutional Neural Network* (CNN)

Sumber : Katole dkk., 2015

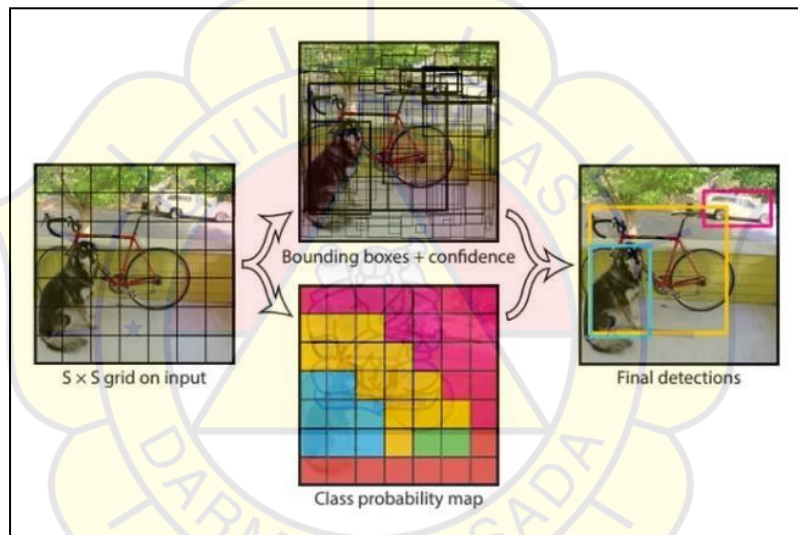
Pada ilustrasi **Gambar 2.12**, proses ekstraksi fitur dalam CNN melibatkan beberapa lapisan tersembunyi atau *hidden layer*, di mana output dari lapisan konvolusi pertama menjadi input bagi lapisan konvolusi berikutnya. Pada tahap klasifikasi, terdapat lapisan fully connected dan fungsi aktivasi yang menghasilkan output dalam bentuk hasil klasifikasi (Katole dkk., 2015).

2.6 Algoritma *You Only Look Once* (YOLO)

Menurut Haris Nasrullah M. pada tahun 2022, YOLO atau You Only Look Once adalah algoritma deteksi objek yang terkenal karena mencapai tingkat akurasi yang tinggi dan mampu beroperasi secara real-time. Algoritma "You Only Look Once" mengimplikasikan bahwa hanya diperlukan satu perambatan maju melalui satu jaringan saraf tunggal untuk melakukan klasifikasi dan meramalkan kotak pembatas untuk objek yang terdeteksi. Hal ini mengoptimalkan kinerja deteksi dan memungkinkan kecepatan yang lebih tinggi dibandingkan dengan

menggunakan dua jaringan saraf terpisah untuk deteksi dan klasifikasi objek secara terpisah.

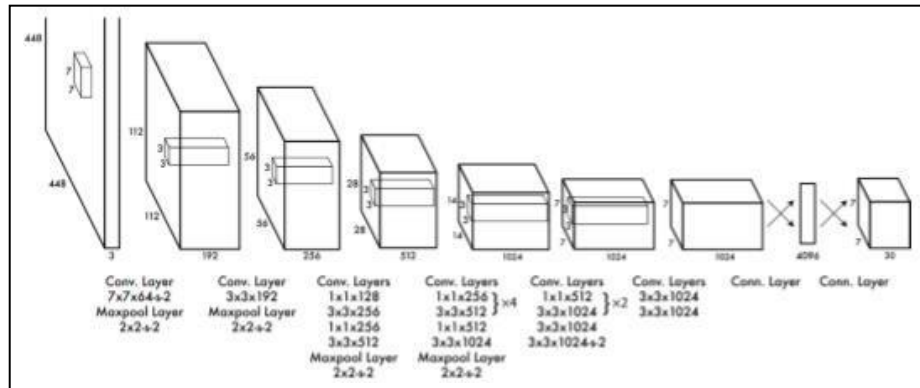
YOLO membawa pendekatan yang inovatif, di mana ini merupakan jaringan saraf konvolusi (CNN) yang mampu melakukan deteksi objek secara real-time. Pendekatan ini melibatkan satu jaringan saraf yang diterapkan pada seluruh gambar, lalu gambar tersebut dibagi menjadi beberapa wilayah. Setiap objek memiliki kotak pembatas yang diberi bobot berdasarkan probabilitas yang telah diprediksi.



Gambar 2.12 Ilustrasi Cara Kerja *You Only Look Once* (YOLO)

Sumber : Redmon dkk., 2016

Pada **Gambar 2.12** citra dibagi menjadi *grid* atau kisi dengan dimensi $S \times S$ dan untuk setiap sel pada kisi memprediksi B *bounding box*, nilai confidence untuk setiap kotak, dan nilai probabilitas untuk setiap kelas C . Prediksi ini diwakili sebagai tensor $S \times S \times (B * 5 + C)$ (Redmon dkk, 2016). Secara umum, arsitektur YOLO digambarkan sebagai *convolutional layer* yang berlapis – lapis.



Gambar 2.13 Arsitektur *You Only Look Once* (YOLO)

Sumber : Redmon dkk., 2016

Jaringan deteksi YOLO terdiri dari 24 lapisan konvolusi yang diikuti oleh 2 lapisan fully connected. Lapisan konvolusi 1x1 secara bergantian mengurangi dimensi ruang fitur dari lapisan sebelumnya. Lapisan konvolusi ini dilatih menggunakan klasifikasi ImageNet dengan setengah resolusi (gambar input 224x224) dan kemudian resolusinya ditingkatkan menjadi 448x448 untuk tujuan deteksi (Redmon dkk, 2016).

2.6.1 Deskripsi YOLO v5

Menurut A Levina pada tahun 2021, YOLO v5 diperkenalkan sebagai generasi kelima dalam model pengenalan objek dan dirilis pada bulan April 2020. Secara umum, struktur arsitektur model ini tidak jauh berbeda dari versi YOLO sebelumnya. YOLO v5 ditulis dalam bahasa pemrograman Python, berbeda dengan pendahulunya yang ditulis dalam bahasa C. Keputusan ini memudahkan proses instalasi dan integrasi dengan perangkat IoT. Selain itu, komunitas PyTorch yang mendukung YOLO v5 lebih besar daripada komunitas darknet, menunjukkan bahwa PyTorch memiliki potensi pertumbuhan dan kontribusi yang lebih besar di masa depan.

Meskipun YOLO v4 dan YOLO v5 ditulis dalam bahasa pemrograman yang berbeda dan menggunakan dua kerangka kerja yang berbeda, membuat perbandingan kinerja yang akurat antara keduanya menjadi sulit. Walaupun demikian, seiring berjalannya waktu, YOLO v5 terbukti lebih efektif daripada YOLO v4 dalam beberapa situasi dan mendapatkan dukungan dari komunitas computer vision bersama dengan YOLO v4. Seperti yang telah disebutkan, struktur arsitektur YOLO v5 memperkenalkan inovasi terbaru yang mirip dengan YOLO v4, sehingga secara teoritis tidak ada perbedaan signifikan antara keduanya.

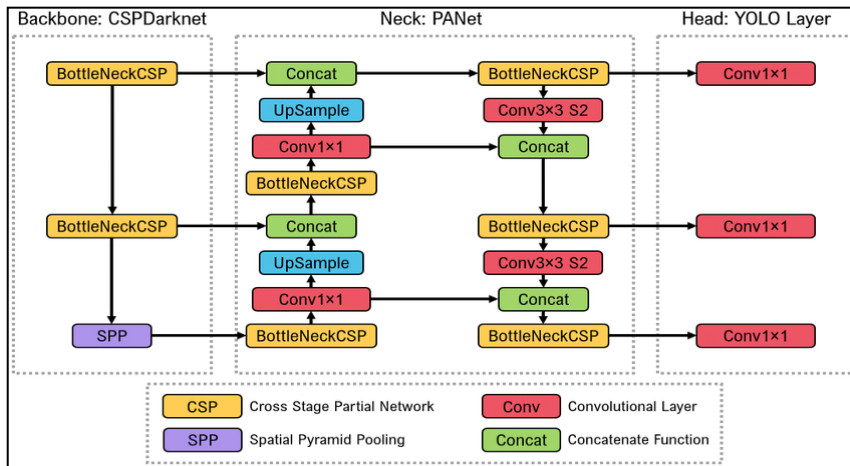
Pengembang model YOLO v5 tidak merinci model dalam makalahnya, melainkan hanya mempublikasikan repositori di Github dan melakukan pembaruan di sana. Dengan menganalisis struktur kode pada file .yaml, terlihat bahwa model YOLO v5 menggunakan struktur backbone yang difokuskan dan jaringan CSP pada bagian neck, yang menggunakan SPP dan PANet. Pada bagian head, model ini mengadopsi head YOLO v3 dengan menggunakan GIoU-loss. Glenn Jocher juga mengusulkan integrasi proses pemilihan anchor box ke dalam YOLO v5. Dengan demikian, pada jaringan YOLOv5, tidak perlu mempertimbangkan parameter ini pada dataset mana pun sebagai input, memungkinkan model untuk secara otomatis mempelajari anchor box yang optimal untuk dataset tersebut dan menggunakannya selama proses pelatihan.

2.6.2 Arsitektur YOLO v5

Menurut A Levina pada tahun 2021, YOLO v5 memperlihatkan perbedaan signifikan dari versi sebelumnya. YOLO v5 menggantikan Darknet dengan PyTorch sebagai kerangka kerjanya. Model ini menggunakan CSP Darknet 53 sebagai bagian backbone, yang menyelesaikan masalah pengulangan informasi gradien pada backbone yang besar. Dengan menggabungkan perubahan gradien ke dalam peta fitur, hal ini tidak hanya mengurangi kecepatan inferensi dan meningkatkan akurasi, tetapi juga mengurangi ukuran model dengan mengurangi jumlah parameter.

Pada YOLO v5, Path Aggregation Network (PANet) digunakan sebagai neck untuk meningkatkan arus informasi. PANet memanfaatkan Feature Pyramid Network (FPN) baru yang mencakup beberapa lapisan bottom-up dan top-down, meningkatkan propagasi level rendah fitur dalam model. Penggunaan PANet juga membawa peningkatan dalam lokalisasi pada lapisan bawah, yang berdampak positif pada akurasi lokalisasi objek.

Head pada YOLO v5 tetap konsisten dengan YOLO v4 dan YOLO v3, menghasilkan tiga keluaran peta fitur yang berbeda untuk mencapai prediksi skala multi. Pendekatan ini membantu meningkatkan efisiensi dalam prediksi objek dari ukuran kecil hingga besar dalam model. Prosesnya dimulai dengan menyajikan gambar ke CSP Darknet 53 untuk ekstraksi fitur, dilanjutkan dengan pengolahan di PANet untuk fungsi fitur. Akhirnya, lapisan YOLO menghasilkan hasil deteksi. Gambaran arsitektur YOLO v5 dapat dilihat pada gambar yang telah diberikan.



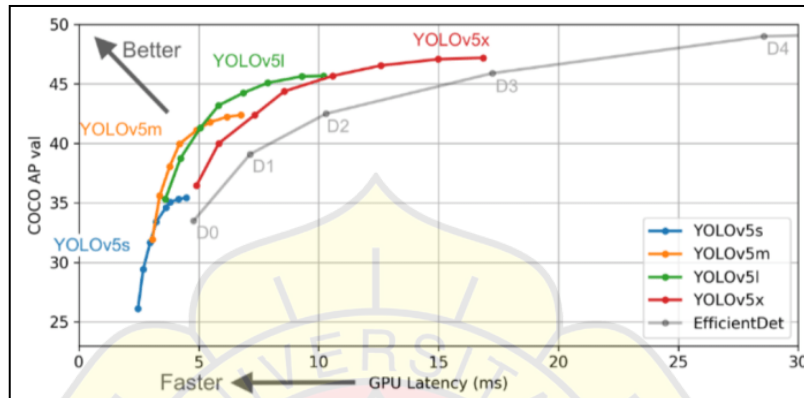
Gambar 2.14 Arsitektur YOLOv5

Sumber : A Levina, 2021

Conv menunjukkan lapisan konvolusi. C3 terdiri dari tiga lapisan konvolusi dan modul yang melibatkan berbagai hambatan. Spatial Pyramid Pooling (SPP) merupakan lapisan pooling yang dimanfaatkan untuk mengatasi batasan ukuran tetap pada jaringan. Upsample digunakan dalam melakukan upsampling pada fusi lapisan sebelumnya di node terdekat. Concat adalah lapisan pengiris yang berfungsi untuk mengiris lapisan sebelumnya. Tiga Conv2d terakhir adalah modul deteksi objek yang diterapkan di head dari jaringan..

2.6.3 Tipe YOLO v5

Menurut A Levina pada tahun 2021, YOLO v5 memiliki beberapa varian dengan perbedaan dalam kecepatan deteksi dan performa mAP. Berikut adalah ilustrasi perbandingan antar jenis YOLO v5.



Gambar 2.15 Performa Tipe Model YOLOv5

Sumber : Jurnal A Levina, 2021

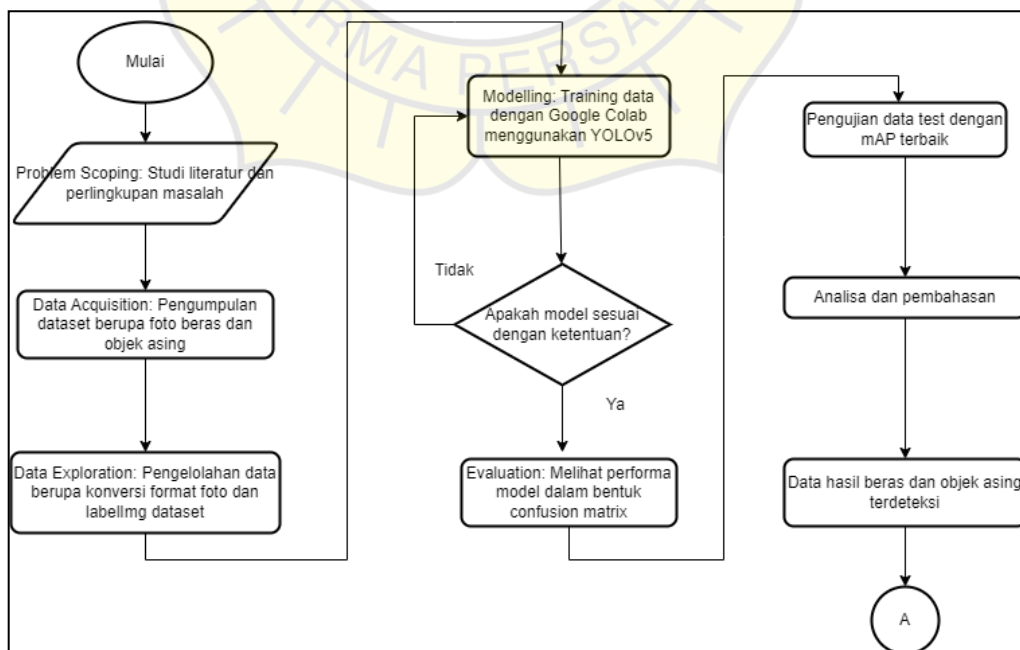
Tabel 2.5 Performa Tipe Model YOLO v5

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	45.2	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

Pada **Gambar 2.15** diperlihatkan bahwa semakin tinggi pada sumbu y grafik model, maka performa mAP model akan semakin baik, namun semakin ke kiri, deteksi akan semakin cepat. Tabel 2.5 menampilkan beberapa indikator yang dapat dijadikan pertimbangan saat memilih model. Model YOLO v5 tipe ini dilatih dengan menggunakan data COCO berukuran 640 piksel. Dalam konteks kecepatan, semakin besar nilai params, maka kecepatan juga akan meningkat. Params merupakan parameter pelatihan, sehingga komputasi akan memakan waktu lebih lama jika nilai parameter semakin besar, dan deteksi pun akan menjadi lebih lambat.

2.6.4 Alur Kerja Metode YOLO-v5

Tahapan dan metode dalam pengembangan program YOLOv5 dijelaskan melalui *AI Project Cycle*, sebuah rangkaian langkah atau metode kerja dalam pembuatan proyek kecerdasan buatan. *AI Project Cycle* yang diterapkan dalam penelitian ini dipresentasikan melalui diagram alur pada gambar 2.17.



Gambar 2.16 Diagram Alur Proses AI Project Cycle Untuk Metode YOLOv5

Berdasarkan ilustrasi pada Gambar 2.16, langkah-langkah dalam perancangan program menggunakan metode YOLOv5 mengikuti tahapan-tahapan berikut dalam AI

Project Cycle:

1. *Problem Scoping*

Proses ini bertujuan untuk mengidentifikasi permasalahan yang ada dan mencari solusi menggunakan kecerdasan buatan (AI). Dalam tahap penentuan ruang lingkup masalah (problem scoping), dilakukan pembagian rinci menggunakan konsep 4W, yaitu What, Who, Why, dan Where, sebagaimana tergambar pada Tabel 3.3. Pada langkah ini, peneliti melakukan pencarian, pembelajaran, dan pemahaman terhadap materi dari berbagai sumber literatur yang relevan dengan masalah yang dihadapi. Referensi tersebut dapat berasal dari berbagai sumber, termasuk penelitian-penelitian sebelumnya, jurnal, artikel, buku, dan sumber ilmiah lainnya.

Tabel 2.6 *Problem Scoping*

4W	Rincian
<i>What</i>	Berdasarkan padi yang sudah melalui proses tahapan penggilingan pascapanen, masih terdapat objek asing seperti butiran merah/hitam, butiran kerikil (batu), hewan (kutu), dan butiran gabah.
<i>Who</i>	Pertanian dan Penggilingan Padi
<i>Where</i>	Penggilingan Padi
<i>Why</i>	Selama ini Penggilingan Padi Bapak Hj Burhan masih menggunakan alat tradisional untuk melakukan proses penggilingan padi pascapanen. Alat tersebut tidak dapat menyaring objek asing secara benar, maka dari itu masih banyak sekali campuran yang terdapat di dalam karung penyimpanan beras.

2. *Data Acquisition*

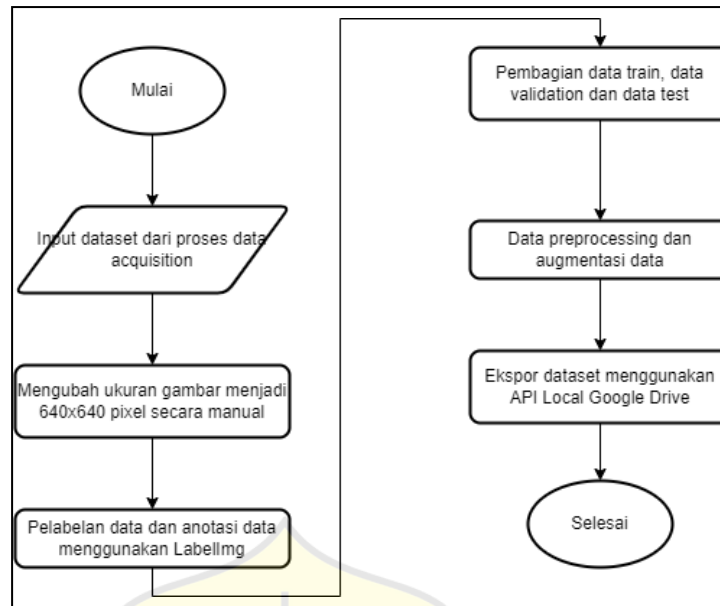
Proses ini dilakukan untuk mengumpulkan dataset yang terdiri dari gambar beras dan objek asing pada pertanian padi, yang nantinya akan digunakan untuk melatih model YOLOv5. Dalam penelitian ini, peneliti menggunakan sampel dataset yang telah diambil oleh penulis setelah proses penggilingan padi selesai dan objek keluar melalui konveyor. Sampel foto diambil dengan izin dari Bapak Hj Burhan. Data yang diambil mencakup lima kelas objek, yaitu butiran beras, butiran kerikil (batu), butiran merah/hitam, hewan (kutu), dan butiran gabah. Jumlah total sampel foto beras dan objek asing yang akan diproses mencapai 1000, seperti yang terlihat pada gambar 2.18.



Gambar 2.17 Dataset Sampel Beras dan Objek Asing

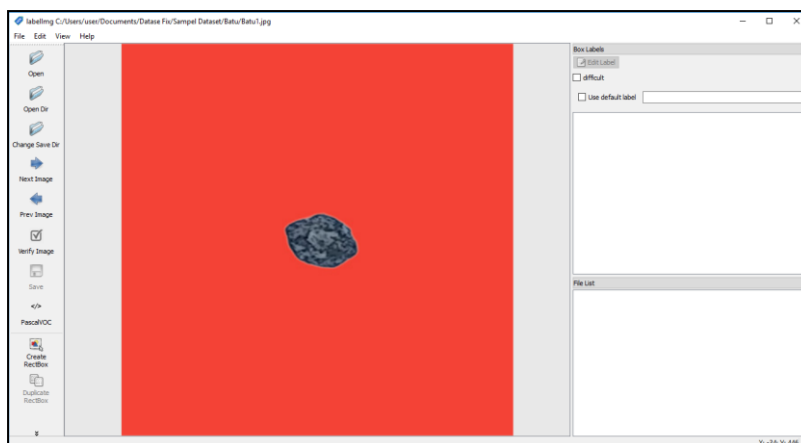
3. *Data Exploration*

Langkah ini bertujuan untuk memproses dataset yang telah dikumpulkan. Pengolahan foto dilakukan agar format setiap gambar sesuai dengan persyaratan algoritma deteksi yang telah dibuat. Rincian langkah-langkah pada eksplorasi data dapat ditemukan dalam gambar 2.18..



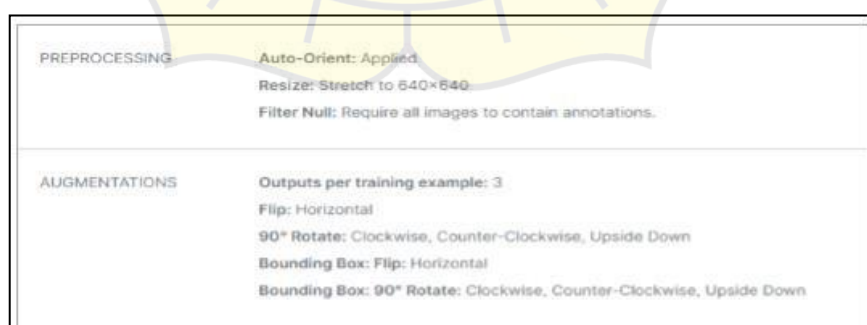
Gambar 2.18 Diagram Alir *Data Exploration*

Langkah pertama dalam eksplorasi data adalah meresize citra menjadi 640x640 piksel menggunakan aplikasi. Ini dilakukan karena adanya keterbatasan pada GPU yang digunakan; ukuran yang terlalu besar dapat menyulitkan proses pelatihan dan memberikan beban yang berat pada GPU. Setelah mengubah ukuran gambar dataset, langkah selanjutnya adalah melakukan pelabelan dan anotasi menggunakan LabelImg. Pelabelan data dilakukan untuk memberikan nama atau label pada objek yang diannotasikan, sedangkan anotasi data dilakukan untuk menandai kotak batas pada objek di gambar, yang biasa dikenal sebagai bounding box, seperti yang ditunjukkan pada gambar 2.19.



Gambar 2.19 LabelImg Sampel Dataset Objek Asing

Selain menyiapkan dataset sebanyak 750 gambar sebelumnya untuk membuat model, juga disiapkan data uji berupa 250 citra asli. Setelah proses pelabelan selesai, dataset dibagi menjadi dua bagian dengan rasio 70% data latih dan 30% data validasi. Dengan demikian, total data pada masing-masing set adalah 500 untuk set latih dan 250 untuk set validasi. Langkah berikutnya dalam eksplorasi data adalah tahap preprocessing dan augmentasi data. Detail dari preprocessing data dan augmentasi data yang digunakan dapat ditemukan pada gambar 2.20.



Gambar 2.20 Rincian *Preprocessing Data* dan *Augmentasi Data*

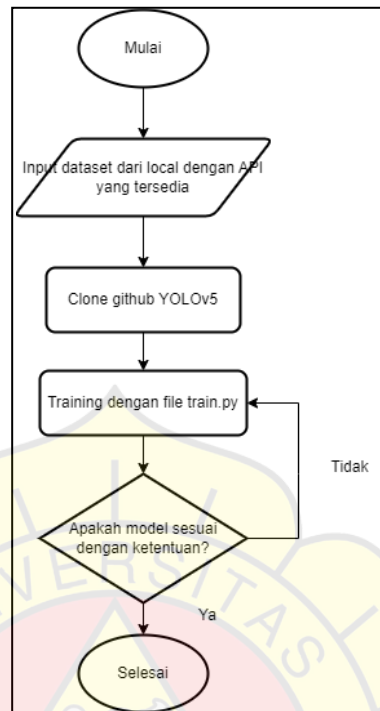
Langkah preprocessing data bertujuan untuk memastikan konsistensi properti gambar pada saat pembelajaran dan inferensi. Penggunaan Auto-Orient dilibatkan untuk menghilangkan informasi EXIF dari gambar, sehingga tampilannya sama seperti saat disimpan di disk. Selanjutnya, ukuran data diatur menjadi 640x640 piksel, mempertimbangkan resolusi gambar yang terkumpul, waktu latihan model, dan pertimbangan pada pretrained model YOLOv5, yang sebelumnya dilatih dengan gambar berukuran 640 piksel. Penggunaan filter null digunakan untuk menghapus data yang telah ditandai sebagai null atau tidak memiliki objek pada suatu gambar, sehingga tidak ikut dalam proses pelatihan model.

Tahap augmentasi data dilakukan untuk mengatasi overfitting pada model dan meningkatkan akurasi inferensi dengan menambah variasi dalam dataset. Augmentasi data terdiri dari dua variasi, yaitu Flip dan Rotasi 90°, dengan detail masing-masing dijelaskan pada Gambar 2.20. Total dataset yang dihasilkan berjumlah 1000 foto, dengan pembagian setiap set sebanyak 500 untuk set latih, 250 untuk set validasi, dan 250 untuk set pengujian.

4. *Modelling*

Langkah pemodelan merupakan fase di mana dataset yang telah dipersiapkan akan diolah untuk membuat sebuah model. Proses pelatihan model dilakukan dengan menggunakan model YOLOv5 yang sudah dilatih sebelumnya. Penggunaan pretrained model ini bertujuan untuk mempercepat waktu pelatihan dan menghasilkan kinerja model yang memadai untuk deteksi objek. Pelatihan dataset dilakukan dengan

menggunakan *Google Colab*. Detail proses pemodelan dapat dilihat pada Gambar 2.21.



Gambar 2.21 Proses Tahap Modelling

Setelah dataset yang telah diolah sebelumnya, proses pengunduhan dilakukan melalui alat Google Colab menggunakan API yang disediakan oleh Google Drive. Model dan kode sumber diambil dari situs resmi YOLOv5 melalui Github. Setelah dataset dan model YOLOv5 dimuat ke Google Colab, langkah berikutnya adalah melatih model dengan menggunakan kode `train.py` yang ditulis dalam bahasa pemrograman Python. Pada saat pelatihan model, beberapa konfigurasi seperti ukuran gambar, batch, dan epoch diatur sesuai kebutuhan. Dalam proses ini, metode "trial and error" digunakan untuk mengoptimalkan konfigurasi skema pelatihan hingga ditemukan konfigurasi dengan tingkat akurasi yang tinggi. Beberapa metrik evaluasi yang digunakan untuk menganalisis kinerja model melibatkan Confusion Matrix, Precision, Recall, mAP, loss,

dan akurasi.

5. Evaluasi

Dalam menganalisis kinerja model yang telah dilatih, diperlukan sejumlah metrik evaluasi. Beberapa metrik evaluasi yang diperhatikan untuk mengevaluasi kinerja model meliputi matriks kebingungan (*confusion matrix*), presisi (*precision*), *recall*, mAP, dan akurasi.

6. Uji Dataset

Pada tahap ini, dilakukan pengujian pada data pengujian menggunakan model deteksi objek yang telah di-training sebelumnya, menghasilkan mAP terbaik. Pengujian ini melibatkan parameter evaluasi seperti presisi, *recall*, dan akurasi.

7. Analisis dan Diskusi

Pada proses ini, dilakukan analisis terhadap hasil uji yang telah diperoleh. Setelah uji deteksi dilakukan, data hasil prediksi akan dibandingkan dengan data hasil label. Data yang diperoleh akan diolah untuk dihitung dan diproses sebagai jumlah populasi dari penelitian ini.

2.7 *Hyperparameter*

Menurut Haris Nasrullah M. (2022), Hyperparameter adalah variabel yang mengendalikan proses pelatihan dan menetapkan nilai parameter model. Variabel ini merupakan faktor kunci yang memengaruhi kinerja model untuk mencapai hasil yang optimal. Hyperparameter tidak diperoleh secara otomatis selama pelatihan, melainkan diinisiasi secara manual sebelum memulai proses pelatihan. Dalam YOLO, terdapat 31 *hyperparameter*, tetapi yang umumnya digunakan meliputi *optimizer*, *learning rate*, *epoch*, *batch*, dan *weight*.

2.7.1 Optimizer dan Learning Rate

Menurut Haris Nasrullah M. (2022), Optimizer merujuk pada algoritma yang digunakan untuk mengubah atribut jaringan saraf, seperti weight dan learning rate, dengan tujuan meminimalkan cost function. Dalam YOLOv7, Stochastic Gradient Descent (SGD) digunakan sebagai algoritma optimizer. SGD bekerja dengan memperkirakan gradien kesalahan untuk status model saat ini menggunakan contoh dari kumpulan data pelatihan. Selanjutnya, algoritma ini memperbarui bobot model melalui backpropagation dari kesalahan, yang dikenal sebagai propagasi balik sederhana. Jumlah bobot yang diperbarui selama pelatihan disebut sebagai ukuran langkah atau "learning rate". Learning rate adalah hyperparameter yang dapat dikonfigurasi dan digunakan dalam pelatihan jaringan saraf, dengan nilai positif kecil, biasanya berada dalam rentang antara 0,0 dan 1,0 (Doshi, 2019).

Learning rate berfungsi sebagai pengendali untuk menyesuaikan seberapa cepat model beradaptasi dengan masalah yang dihadapi. Nilai yang lebih kecil membutuhkan waktu pelatihan yang lebih lama karena perubahan yang lebih kecil pada bobot setiap kali diperbarui. Di sisi lain, nilai yang lebih besar dapat menghasilkan perubahan yang cepat, mengurangi waktu pelatihan. Namun, jika learning rate terlalu besar, model dapat konvergen ke solusi yang kurang optimal, sementara jika terlalu kecil, proses pelatihan dapat menjadi lambat dan tidak stabil (Doshi, 2019).

2.7.2 Epoch

Menurut Haris Nasrullah M. (2022), Epoch merujuk pada jumlah iterasi pelatihan yang digunakan untuk memproses semua batch data yang masuk ke dalam

optimizer dan keluar dari optimizer. Dalam satu epoch, seluruh dataset melewati proses pelatihan dan diulang kembali dari awal. Nilai epoch biasanya ditentukan secara eksperimental dan dapat dimulai dari angka kecil seperti 10, kemudian ditingkatkan secara bertahap hingga tidak ada penurunan yang signifikan pada loss function. Peningkatan jumlah epoch dapat meningkatkan akurasi model, namun perlu diperhatikan bahwa semakin banyak epoch yang digunakan, waktu pelatihan juga akan meningkat (Emanuella dkk., 2022).

2.7.3 Batch Size

Haris Nasrullah M. (2022) menyatakan bahwa batch size merupakan salah satu hyperparameter utama yang harus disesuaikan sebelum memulai proses pelatihan. Batch size mengacu pada jumlah gambar yang digunakan dalam setiap epoch untuk melatih jaringan. Pengaturan hyperparameter yang terlalu tinggi dapat mengakibatkan jaringan memerlukan waktu yang terlalu lama untuk mencapai konvergensi, di mana tidak ada peningkatan akurasi yang signifikan. Sebaliknya, jika batch size terlalu rendah, jaringan dapat berputar-putar tanpa mencapai kinerja yang dapat diterima. Penting untuk mencatat bahwa sifat dari dataset juga dapat memengaruhi ukuran batch (Kandel & Castelli, 2020).

2.7.4 Weight

Haris Nasrullah M. (2022) menjelaskan bahwa weight atau bobot dalam konteks neural network merujuk pada koefisien dari fungsi yang akan dipecahkan. Bobot merupakan nilai riil yang terkait dengan setiap fitur, mencerminkan signifikansi fitur tersebut dalam memprediksi nilai akhir. Saat neural network dilatih pada dataset, bobot diinisialisasi, dan selama proses pelatihan, bobot ini terus dioptimalkan untuk menghasilkan bobot yang paling optimal (Kandel &

Castelli, 2020).

2.8 Evaluasi Kinerja Model

Dalam pengembangan model deteksi objek, penting untuk melakukan evaluasi kinerja model guna mengukur sejauh mana model tersebut berhasil dalam mendeteksi objek yang ditargetkan. Beberapa metode evaluasi umum yang digunakan meliputi *Intersection over Union (IoU)*, *presisi & recall*, *Average Precision*, dan *mean Average Precision (mAP)*..

2.8.1 *Intersection over Union (IoU)*

Intersection over Union (IoU) adalah metrik yang digunakan untuk menilai tingkat kecocokan antara kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar. IoU diukur dengan menghitung luas tumpang tindih antara dua kotak pembatas dan membandingkannya dengan total luas kedua kotak pembatas. Nilai IoU memberikan indikasi seberapa baik kedua kotak pembatas berpotongan dan berfungsi sebagai ukuran untuk mengevaluasi performa model deteksi objek.

Dengan menerapkan IoU, kita dapat menentukan apakah suatu deteksi valid (True Positive) atau tidak (False Positive). IoU dihasilkan oleh area yang tumpang tindih antara kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar dibagi dengan area gabungan di antara keduanya (Padilla et al., 2021). IoU dapat dinyatakan dalam persamaan.

$$IoU = \frac{B_{prediksi} \cap B_{groundTruth}}{B_{prediksi} \cup B_{groundTruth}} \quad (2.1)$$

Dan dapat diilustrasikan pada gambar berikut :

$$\text{IoU} = \frac{\text{area irisan}}{\text{area gabungan}} = \frac{\text{img1} \cap \text{img2}}{\text{img1} \cup \text{img2}}$$

Gambar 2.22 Ilustrasi Persamaan IoU

Sumber : Padilla dkk, 2021 – telah diolah kembali

Gambar 2.22 menunjukkan bahwa formula untuk menghasilkan nilai IoU adalah hanya perbandingan antara area tumpang tindih dan area gabungan. Kriteria untuk mengevaluasi apakah skor IoU tersebut baik atau buruk adalah semakin besar tumpang tindih atau semakin dekat jarak antara bounding box prediksi dan bounding box ground truth (area yang tumpang tindih antara keduanya semakin besar), maka skor IoU akan semakin tinggi.

2.8.2 *Precision-Recall*

Menurut Haris Nasrullah M. pada tahun 2022, Precision merupakan kemampuan model dalam mengenali hanya objek yang relevan sebagai persentase prediksi positif yang benar. Sementara itu, recall adalah kemampuan model untuk menemukan semua kasus yang relevan, yaitu semua bounding box dari ground truth, sebagai persentase prediksi positif yang benar di antara semua ground truth yang tersedia. Penilaian metode deteksi objek sebagian besar didasarkan pada *precision* dan *recall* yang dinyatakan pada persamaan (2.2) dan (2.3).

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{TP}{\text{semua deteksi}} \quad (2.2)$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{TP}{\text{semua ground truth}} \quad (2.3)$$

True Positive (TP) terjadi ketika deteksi bounding box sesuai dengan ground truth, *False Positive* (FP) terjadi ketika deteksi *bounding box* salah, baik karena objek tidak ada atau terjadi kesalahan deteksi terhadap objek yang ada. Sementara

itu, *False Negative* (FN) terjadi ketika *bounding box* dari ground truth tidak terdeteksi (Padilla dkk., 2020). Jika jumlah False Positive rendah, maka nilai precision akan tinggi, tetapi objek yang terlewatkan akan tinggi, menyebabkan nilai False Negative tinggi dan recall rendah. Namun, jika objek diterima dengan mengurangi ambang batas IoU, *recall* akan meningkat, tetapi jumlah False Positive juga akan meningkat, yang pada gilirannya dapat menurunkan nilai precision. Sebagai indikator model yang baik, baik nilai precision maupun recall seharusnya tetap tinggi meskipun ambang batas IoU mengalami variasi.

2.8.3 *mean Average Precision (mAP)*

Mean Average Precision atau mAP adalah metrik yang digunakan untuk mengevaluasi ketepatan detektor objek di seluruh kelas dalam suatu dataset. Sebelum mendapatkan nilai mAP, perlu diketahui nilai Average Precision (AP) dengan menghitung Area Under Curve (AUC) dari kurva precision-recall. Namun, kurva precision-recall sering memiliki pola naik-turun zig-zag, sehingga menentukan nilai AUC bisa sulit. Untuk mengatasi ini, digunakan dua pendekatan, yaitu interpolasi 11 titik dan interpolasi semua titik. Pada interpolasi 11 titik, nilai precision diinterpolasi pada 11 titik recall terdekat, dengan recall yang membentang dari [0.0, 0.1, 0.2, ..., 0.9, 1.0]. Persamaan untuk interpolasi 11 titik dinyatakan dalam persamaan (2.4).

$$AP_{11} = \frac{1}{11} \sum_{R \in \{0,0.1, \dots, 0.9, 1.0\}} P_{interp} R, \quad (2.4)$$

Dimana

$$P_{interp} R = \max_{R:R \geq R} P(\tilde{R})$$

Ini berarti bahwa daripada menggunakan nilai presisi sebenarnya pada titik \tilde{R} , kita mengambil nilai presisi tertinggi dari titik yang paling dekat dengan \tilde{R} , namun hanya mempertimbangkan titik setelah \tilde{R} . Pada interpolasi semua titik, alih-alih menginterpolasi hanya 11 titik dengan jarak yang sama, interpolasi dapat dilakukan melalui semua titik sedemikian rupa sehingga AP untuk semua titik dapat dinyatakan pada persamaan (2.6).

$$AP_{\text{semua}} = \sum_n (R_n + 1 - R_n) P_{\text{interp}}(R_{n+1}), \quad (2.5)$$

Dimana

$$P_{\text{interp}}(R_{n+1}) = \max_{\tilde{R} \leq R \leq R_{n+1}} P(\tilde{R}),$$

Dalam konteks ini, bukannya menggunakan presisi yang diamati hanya pada beberapa titik, AP sekarang diperoleh dengan menginterpolasi presisi pada setiap tingkat, mengambil presisi maksimum yang memiliki nilai recall lebih besar atau sama dengan R_{n+1} . Secara dasar, mean *Average Precision* (mAP) hanyalah AP rata-rata dari semua kelas yang dijelaskan dalam persamaan berikut:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.6)$$

Dengan AP_i mewakili kelas ke i dan N menunjukkan total kelas yang dievaluasi (Padilla dkk., 2020).

2.9 *Pytorch & Torchvision*

PyTorch adalah sebuah perpustakaan sumber terbuka yang pertama kali dikembangkan oleh *Facebook Research*. Berfungsi sebagai pengganti *numpy*, *PyTorch* memungkinkan pengguna memanfaatkan kekuatan GPU dalam proses pembelajaran mesin. Sebagai platform penelitian pembelajaran mesin, *PyTorch* memberikan fleksibilitas dan kecepatan maksimum. *PyTorch* menyediakan Tensor yang dapat beroperasi baik di CPU maupun GPU, mempercepat komputasi dalam skala besar. Salah satu komponen kunci dalam *PyTorch* adalah *Torch*, yang mengandung struktur data untuk tensor multidimensi dan mendefinisikan operasi matematika pada tensor, mirip dengan *numpy*. Dalam pengembangan sistem deteksi objek dengan *PyTorch*, *Torchvision* digunakan sebagai pustaka computer vision yang terintegrasi dengan *PyTorch*. *Torchvision* menyediakan utilitas untuk transformasi gambar dan video yang efisien, model terlatih yang umum digunakan, serta beberapa dataset umum.

2.10 **Flask**

Flask adalah kerangka kerja web yang ditulis dalam bahasa Python dan termasuk dalam kategori microframework. Fungsinya sebagai kerangka kerja aplikasi dan tampilan untuk situs web memungkinkan pengembang untuk membuat situs web yang terstruktur dan mengelola perilaku situs web dengan lebih mudah melalui penggunaan Flask dan bahasa Python.

Dikategorikan sebagai microframework, Flask tidak memerlukan alat atau pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponen umum, seperti validasi formulir dan database, tidak terpasang secara default di Flask. Ini disebabkan oleh ketersediaan fungsi dan komponen-komponen tersebut

dari pihak ketiga, yang dapat diintegrasikan ke dalam Flask melalui ekstensi sehingga fitur dan komponen tersebut seolah diimplementasikan oleh Flask itu sendiri. Meskipun disebut sebagai microframework, Flask tidak mengalami kekurangan dalam hal fungsionalitas. Istilah "microframework" pada Flask merujuk pada tujuan untuk membuat inti aplikasi sesederhana mungkin namun tetap mudah untuk ditambahkan fungsionalitasnya. Dengan demikian, fleksibilitas dan skalabilitas Flask dapat dianggap tinggi dibandingkan dengan beberapa framework lainnya.

2.10.1 Fitur – Fitur Flask

Seperti yang dijelaskan oleh Rahadian Irsyad pada tahun 2018, Flask adalah sebuah microframework yang memiliki inti yang sangat sederhana dan kecil, namun tetap dapat diperluas dan berkembang. Sebagai akibat dari sifat tersebut, fitur bawaan dari Flask cukup sedikit, termasuk:

1. Server pengembangan bawaan.
2. Debugger yang responsif.
3. Dukungan terintegrasi untuk pengujian unit.
4. Kompatibilitas dengan mesin aplikasi Google.
5. Penyusunan permintaan RESTful.
6. Templating menggunakan Jinja2.
7. Dukungan untuk cookies yang aman.
8. Berbasis Unicode.
9. Mengikuti standar WSGI 1.0.

Selain itu, Flask juga didukung oleh dokumentasi yang sangat baik dan berbagai forum di internet yang dapat digunakan untuk mendiskusikan berbagai masalah terkait Flask.

2.10.2 Keunggulan dan Kekurangan Flask

Menurut Rahadian Irsyad pada tahun 2018, Flask memiliki kelebihan dan kekurangan jika dibandingkan dengan kerangka kerja web berbasis Python lainnya seperti Django, CherryPy, dan sebagainya. Beberapa kelebihan Flask meliputi:

1. Kemudahan dalam penggunaan karena memiliki inti yang sederhana dan desain modular.
2. Kemampuan untuk mengelola fungsi permintaan HTTP dengan mudah.
3. API yang baik dan konsisten.
4. Dokumentasi yang lengkap dan terstruktur dengan baik, disertai contoh yang dapat digunakan secara langsung.
5. Kemudahan pemasangan dan penyebaran untuk produksi.
6. Kemudahan dalam pengujian unit (Unit testability).
7. Tingkat fleksibilitas yang tinggi dengan konfigurasi yang mudah diubah.

Meskipun memiliki kelebihan-kelebihan tersebut, Flask juga memiliki kekurangan, di antaranya:

1. Tidak memiliki ORM dan lapisan basis data bawaan, sehingga memerlukan aplikasi pihak ketiga.
2. Tidak bersahabat dengan async.

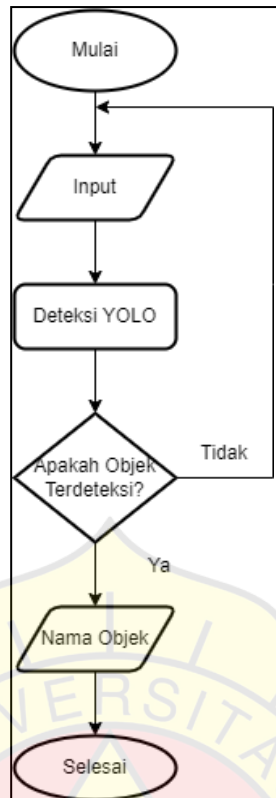
Karakteristik Flask yang mengharuskan keterlibatan aplikasi pihak ketiga untuk menjalankan fiturnya dapat menjadi kelebihan atau kekurangan tergantung pada kebutuhan dan preferensi pengguna.

2.11 Penyusunan Aplikasi Android

Dalam proses identifikasi jenis beras dan objek asing menggunakan algoritma YOLO, langkah-langkah berikut dapat diikuti:

1. Input berupa gambar diperlukan sebagai langkah awal untuk memulai proses deteksi beras dan objek asing.
2. Pemanfaatan algoritma YOLO dilakukan untuk melakukan prediksi dan deteksi objek pada gambar yang telah diinputkan.
3. Jika terdeteksi keberadaan objek, sistem akan menampilkan label untuk beras dan objek asing dengan kotak pembatas. Jika tidak terdeteksi, pengguna dapat menginputkan gambar baru melalui kamera.
4. Output yang dihasilkan berupa kotak pembatas yang menunjukkan lokasi beras dan objek asing dalam gambar.

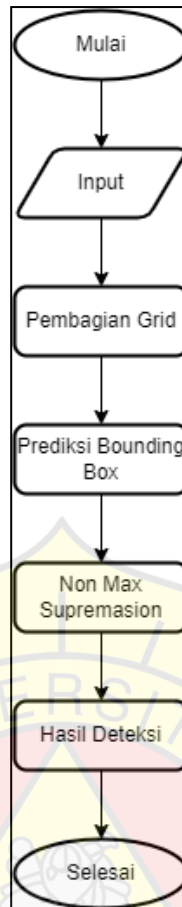
Selain merancang alur kerja aplikasi, pada tahap ini juga dilakukan desain antarmuka untuk tampilan deteksi beras dan objek asing.



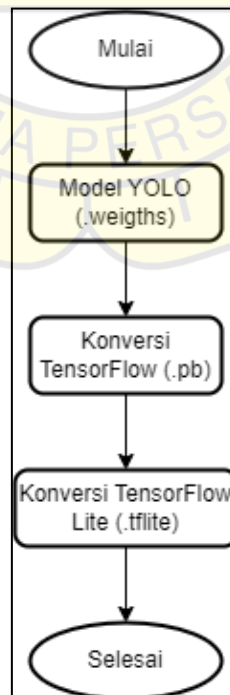
Gambar 2.23 Alur Sistem Aplikasi Android

Berikut adalah beberapa langkah dalam mendeteksi objek menggunakan algoritma YOLO:

1. Tahap pertama dalam proses deteksi adalah menggunakan gambar sebagai input.
2. Gambar tersebut akan dipecah menjadi beberapa grid dengan ukuran $S \times S$.
3. Prediksi bounding box dilakukan untuk menghasilkan nilai probabilitas dan koordinat objek, yang terdiri dari b_h , b_w , b_y , dan b_x .
4. Untuk menemukan bounding box terbaik dari prediksi yang telah dilakukan, dapat digunakan teknik *Non-Max Suppression* (NMS). Nilai NMS dipengaruhi oleh skor objektivitas dan *Intersection over Union* (IoU).



Gambar 2.24 Implementasi Algoritma YOLO



Gambar 2.25 Konversi Model YOLO ke TensorFlow Lite

2.12 Pemodelan UML


UML, yang singkatannya dari Unified Modeling Language, merupakan suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membuat, dan mendokumentasikan berbagai artefak dalam sistem perangkat lunak. Artefak tersebut mencakup berbagai informasi yang dihasilkan selama proses pengembangan perangkat lunak, seperti model, deskripsi, atau perangkat lunak yang dihasilkan.



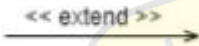
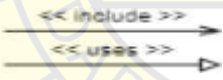
2.12.1 Use case Diagram

Diagram *use case* adalah suatu model yang mengilustrasikan perilaku yang diharapkan dari sistem yang sedang dikembangkan (Sugiarti, 2018). Diagram ini membayangkan interaksi antara satu atau lebih aktor dengan sistem yang sedang dibangun. Penggunaan diagram use case mempermudah pemahaman terhadap fitur-fitur yang tersedia dalam sistem dan menunjukkan siapa yang diizinkan untuk mengakses fitur-fitur tersebut.

Diagram *use case* menggunakan beberapa simbol yang memiliki makna tertentu. Simbol-simbol tersebut pada dasarnya adalah sebagai berikut:

Tabel 2.7 Simbol Use Case Diagram






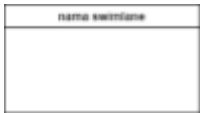
No	Simbol	Deskripsi
1	Use case 	Fungsi yang terpapar oleh sistem sebagai entitas yang melakukan pertukaran pesan antara entitas atau aktor, umumnya diungkapkan dengan kata kerja pada awal frasa kata benda <i>use case</i> .

2	<p>Aktor/actor</p> 	<p>Seseorang, proses, atau sistem lain yang terlibat dalam interaksi dengan sistem informasi yang akan dikembangkan di luar sistem tersebut biasanya dilambangkan dengan aktor. Meskipun simbol aktor berupa gambar seseorang, namun aktor tidak selalu merupakan kata benda pribadi di awal frasa kata benda aktor.</p>
3		<p>Interaksi antara aktor dan partisipan <i>Use Case</i> melibatkan komunikasi di mana partisipan <i>Use Case</i> terlibat dalam interaksi dengan aktor.</p>
4	<p>Ektensi/<i>extend</i></p> 	<p>Keterkaitan tambahan <i>use case</i> dengan <i>use case</i> lain terjadi ketika <i>use case</i> yang ditambahkan memiliki kemampuan untuk berdiri sendiri tanpa ketergantungan pada <i>use case</i> tambahan. Analoginya mirip dengan prinsip pewarisan dalam pemrograman berorientasi objek, di mana <i>use case</i> tambahan cenderung memiliki nama depan yang serupa dengan <i>use case</i> yang ditambakkannya.</p>
5	<p><i>Include</i></p> 	<p>Hubungan <i>use case</i> tambahan terjadi ketika <i>use case</i> yang ditambahkan membutuhkan <i>use case</i> lain sebagai prasyarat atau untuk menyelesaikan tugasnya. Terdapat dua aspek utama dalam memasukkan kasus penggunaan. Penambahan tersebut menyiratkan bahwa <i>use case</i> yang ditambahkan akan diakses setiap kali <i>use case</i> inti dieksekusi.</p>

2.12.2 Activity Diagram

Rosa & Shalahuddin (2018) menyatakan bahwa diagram aktivitas atau diagram tindakan mencerminkan urutan langkah atau prosedur yang menggambarkan alur kerja atau operasi dari suatu sistem, proses bisnis, atau menu perangkat lunak. Perlu ditekankan bahwa diagram aktivitas menggambarkan fungsi yang dapat dijalankan oleh sistem.




Tabel 2.8 Simbol Activity Diagram




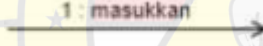
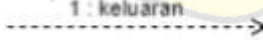
No	Simbol	Deskripsi
1	Status Awal 	Diagram aktivitas memiliki kondisi awal yang menunjukkan keadaan awal operasi sistem.
2	Aktivitas 	Operasi yang dilakukan oleh sistem, biasanya dimulai dengan kata kerja.
3	Percabangan/ <i>decision</i> 	Percabangan asosiasi terjadi ketika terdapat lebih dari satu pilihan aktivitas.
4	Penggabungan/ <i>join</i> 	Penggabungan asosiasi terjadi ketika lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Diagram fungsional memiliki keadaan akhir yang menunjukkan kondisi akhir yang dijalankan oleh sistem.
6	<i>Swimlane</i> 	Menggunakan swimlane untuk memisahkan aktivitas yang memiliki tanggung jawab terhadap aktivitas yang terjadi.

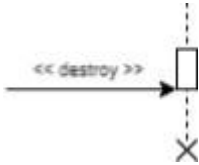
2.12.3 Sequence Diagram

Diagram urutan mengilustrasikan perilaku objek dalam suatu skenario penggunaan, mencakup masa hidup objek, serta pesan yang dikirim dan diterima antar objek. Oleh karena itu, dalam mendeskripsikan diagram urutan, informasi tentang objek yang terlibat dalam kasus penggunaan, bersama dengan metode-metode kelas yang digunakan oleh objek tersebut, perlu diketahui. Pembuatan diagram urutan juga penting untuk memahami skenario yang terkait dengan kasus penggunaan (Rosa dan Shalahuddin, 2018).

Tabel 2.9 Simbol Sequence Diagram

No	Simbol	Deskripsi
1	<p>Aktor</p> 	<p>Individu, proses, atau sistem lain yang terlibat dalam interaksi dengan sistem informasi yang akan dikembangkan berada di luar lingkup sistem tersebut. Oleh karena itu, walaupun simbol aktor direpresentasikan dengan gambar manusia, tidak selalu berarti bahwa aktor tersebut adalah individu yang menggunakan suatu objek atau sistem.</p>
2	<p>Garis Hidup/ <i>Lifeline</i></p> 	<p>Menggambarkan eksistensi atau masa hidup suatu objek.</p>
3	<p>Objek</p> 	<p>Mengindikasikan objek yang terlibat dalam pertukaran pesan.</p>

4	<p style="text-align: center;">Waktu Aktif</p> 	<p>Mendefinisikan objek yang aktif dan dapat berinteraksi, termasuk segala sesuatu yang berkaitan dengan kegiatan dalam rentang waktu tersebut.</p>
5	<p style="text-align: center;">Pesan Tipe <i>Create</i></p> 	<p>Ini mengindikasikan bahwa sebuah objek menghasilkan atau menciptakan objek lain, ditunjukkan oleh arah panah yang mengarah ke objek yang baru diciptakan.</p>
6	<p style="text-align: center;">Pesan Tipe <i>Call</i></p> 	<p>Menyatakan bahwa suatu objek memanggil fungsi/metode pada objek lain atau pada dirinya sendiri.</p>
	<p style="text-align: center;">Pesan Tipe <i>Send</i></p> 	<p>Ini menunjukkan bahwa objek asing mengirimkan data, input, atau informasi ke objek lain, ditandai dengan panah yang mengarah ke objek penerima.</p>
8	<p style="text-align: center;">Pesan Tipe <i>Return</i></p> 	<p>Untuk menunjukkan bahwa suatu objek yang melakukan operasi atau metode menghasilkan pengembalian ke objek yang ditentukan, panah menunjuk ke objek yang menerima pengembalian.</p>

9	<p>Pesan Tipe Destory</p> 	<p>Mengindikasikan bahwa suatu objek mengakhiri eksekusi atau kehidupan objek lain, ditunjukkan oleh panah yang mengarah ke objek yang akan diakhiri. Jika ingin membuat objek baru, disarankan untuk menghancurkan objek yang sudah ada.</p>
---	---	---

