

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Penelitian Terkait

Dalam konteks penelitian sedang dikerjakan, tinjauan kepada penelitian terdahulu menjadi krusial karena memungkinkan dalam memahami hubungan antara penelitian sebelumnya dan penelitian terkini. Berikut merupakan ringkasan penelitian pendahulunya.

**Tabel 2.1 Tinjauan Penelitian Terdahulu**

No	Penulis	Tahun	Metode	Judul	Kesimpulan
1	N. Prakash, R. Rajakumar, N. Leela Madhuri, M. Jyothi, A. Pavitra Bai, M. Manjunath, K. Gowthami	2022	Algoritma <i>Convolutional</i> <i>Neural</i> <i>Network</i> (CNN)	<i>Image</i> <i>Classification</i> <i>for Rice</i> <i>Varieties</i> <i>Using Deep</i> <i>Learning</i> <i>Models</i>	Padi adalah tanaman dengan pertumbuhan tercepat di India dan seiring bertambahnya populasi, permintaan beras juga meningkat. Sebagian besar negara Asia menanam beras dan mengekspornya ke seluruh dunia. Bergantung pada budaya makanan masyarakat, berbagai

					<p>jenis beras dibudidayakan. Pada saat yang sama, kualitas makanan menjadi prioritas, jadi kami menggunakan teknik <i>computer vision</i> untuk mengekstraksi karakteristik kualitas beras. Masing-masing <i>varietas</i> ini memiliki 15.000 gambar, dimana total 75.000 gambar digunakan dalam pelatihan dan pengujian.</p> <p>Pengklasifikasi citra terbaik dipilih berdasarkan akurasi terbaik. Hasil model yang diusulkan mempertimbangkan kinerja yang lebih</p>
--	--	--	--	--	---

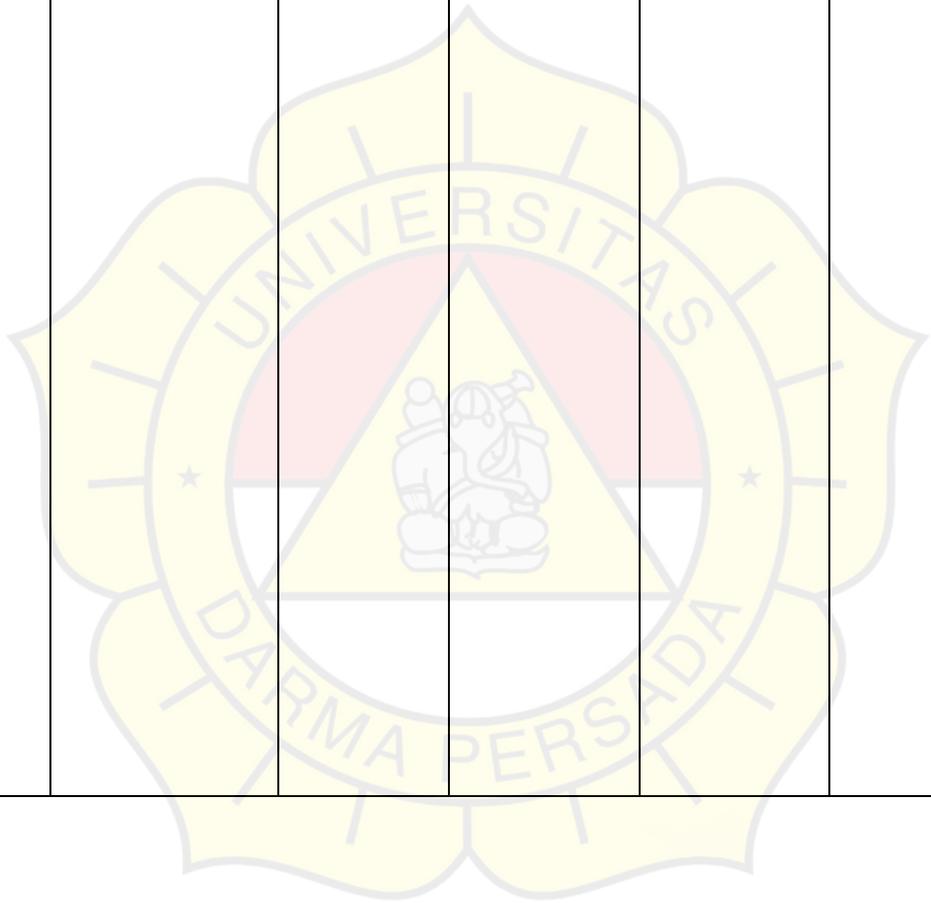
					baik dalam menentukan <i>varietas</i> padi.
2	Alfath Daryl Alhajir (Universitas Pembangunan Nasional “Veteran”)	2018	YOLO ( <i>You Only Look Once</i> )	Implementasi Metode <i>Object Detection</i> dengan Algoritma YOLO ( <i>You Only Look Once</i> ) Untuk Deteksi Beras dan Objek Asing	Algoritma yang digunakan untuk deteksi objek adalah algoritma YOLO, dipilih karena kesederhanaan dan kemudahan mengadaptasi model ke berbagai kasus penggunaan. Algoritma YOLO adalah algoritma berbasis CNN, yaitu algoritma berbasis jaringan saraf yang digunakan dalam bidang studi pembelajaran mesin yang ditujukan untuk pemrosesan gambar atau visi komputer.

					<p>Dalam penelitian ini, algoritma yang berhasil diimplementasikan sebagai sistem deteksi diskriminasi beras berhasil mencapai akurasi rata-rata 0,9015 mAP pada dataset pelatihan dan 0,5944 pada dataset, menggunakan metode seperti <i>augmentasi</i> data, pencarian <i>grid</i>, yang jelas sangat membantu kinerja teladan.</p>
--	--	--	--	--	---

3	Nova Eka Budiyanta, dkk. (Universitas Katolik Indonesia Atma Jaya)	2021	YOLO ( <i>You Only Look Once</i> )	Sistem Deteksi Kemurnian Beras Berbasis <i>Computer Vision</i> dengan Pendekatan Algoritma YOLO	Tujuan penelitian ini adalah menerapkan sistem deteksi kemurnian beras pada campuran kontaminan, yang akan digunakan sebagai parameter nilai dalam pemilahan kontaminan yang terdeteksi pada proses kontrol kualitas beras. Prosesi pelatihan model telah berhasil meminimalisasi <i>loss</i> dengan <i>signifikans</i> , dimana nilai <i>loss</i> nya sejumlah 1,89 saat <i>iterasi</i> yang ke 1000 dan 0,16 saat <i>iterasi</i> yang ke 15000. Kesuksesan prosesi melatih model ini menandakan ujian model dalam
---	---	------	--	---	---

					menerapkan prosesi observasi berlangsung maksimal dan bagus, serta dilaporkan skor akurasi rata-ratanya 86,11 %.
4	Arum TiaraSari, Emy Haryatmi.	2021	<i>Faster R-CNN</i>	Penerapan <i>Convolutional Neural Network Deep Learning</i> dalam Pendeteksian Citra Biji Jagung Kering.	Tujuan penelitian ini adalah menerapkan metode <i>Convolutional Neural Network</i> atau (CNN) <i>deep learning</i> adalah cara deteksi citra sebuah biji jagung kering. Penelitian terdiri dari tiga tahap utama. Dimana pertama, dilakukan preprocessing untuk normalisasi data input citra biji jagung kering dengan cara <i>cropping &amp; wrapping</i> .

					Lalu kedua,
--	--	--	--	--	-------------



					<p>dibentuk model &amp; melatih sistem menggunakan CNN.</p> <p>Tahap akhir atau ketiga ialah menguji sistemnya, CNN digunakan untuk kenali citra biji jagung kering serta untuk penentuan nilai akurasi deteksinya.</p> <p>Penelitian ini memakai 20 citra biji jagung untuk data <i>testing</i>, sementara 80 buah citra biji jagung dipakai sebagai dataset <i>training</i>. Hasil dari penelitian menunjukkan bahwa penilaiam akurasi deteksi biji jagung kering ini dipengaruhi Oleh ukuran serta posisi pengambilan</p>
--	--	--	--	--	--

					<p>citra dari kamera hp. Pemakaian tujuh <i>convolutional layer</i> menghasilkan nilai akurasi kisaran 80% hingga 100%. Nilai rerata akurasi data testing sejumlah 0,90296. Pemakaian <i>convolutional layer</i> memungkinkan sistem untuk mendeteksi kekuatan bentuk dari suatu citra.</p>
5	Muhammad Pailus	2022	<i>Faster R-CNN</i>	Pengembangan Model Identifikasi Penyakit Pada	<p>Penelitian ini menggunakan metode <i>Faster R-CNN</i> untuk mengidentifikasi</p>

				<p>Tanaman Padi</p> <p>Berbasiskan</p> <p><i>Faster R-</i></p> <p>CNN</p>	<p>penyakit pada</p> <p>tanaman padi dengan</p> <p>menggunakan 1500</p> <p>data yang diambil</p> <p>secara mandiri. Data</p> <p>ini terdiri dari tiga</p> <p>kelas penyakit dengan</p> <p>masing-masing kelas</p> <p>memiliki 500 gambar</p> <p>sebelum diannotasi.</p> <p>Setelah itu, data</p> <p>tersebut diannotasi</p> <p>dan diresize menjadi</p> <p>ukuran 512x512</p> <p>sebelum dilakukan</p> <p>proses <i>training</i>.</p> <p>Model ini</p> <p>menggunakan</p> <p><i>backbone</i> dan</p> <p><i>pretrained</i> model</p> <p>resnet50_fpn, serta</p> <p>Fast R-CNN sebagai</p> <p><i>predictor</i> pada layer</p> <p>terakhirnya. Dengan</p>
--	--	--	--	---	--

					<p>konfigurasi ini, model mampu mendeteksi penyakit spesifik di area yang terinfeksi pada tanaman padi. Selain itu, model juga mampu mendeteksi lebih dari satu jenis penyakit pada tanaman padi dalam satu gambar. Hasil pengujian menunjukkan bahwa model mencapai nilai <i>mean Average Precision</i> (mAP) sebesar 87.6% dan <i>recall</i> sebesar 85.7%, yang menandakan performa yang baik dalam mendeteksi dan mengidentifikasi penyakit pada tanaman padi.</p>
--	--	--	--	--	--

					<p>Selanjutnya, model ini juga diuji pada dataset publik dengan resolusi gambar yang berbeda-beda, dan hasilnya cukup memuaskan karena model mampu mendeteksi penyakit yang ada dengan baik.</p> <p>Secara keseluruhan, penelitian ini membuktikan bahwa menggunakan Faster R-CNN dengan menggunakan resnet50_fpn sebagai <i>pretrained</i> model dan Fast R-CNN sebagai <i>predictor</i> dapat menghasilkan model yang efektif dalam mendeteksi penyakit pada tanaman padi</p>
--	--	--	--	--	---

					dengan akurasi yang tinggi.
6	Madhusri Maity; Sriparna Banerjee; Sheli Sinha Chaudhuri	2021	<i>Faster R-CNN</i> dan YOLO	Deteksi kendaraan berbasis <i>Faster R-CNN</i> dan YOLO	Deteksi kendaraan bergerak otomatis memainkan peran penting dan menantang dalam melakukan pengawasan lalu lintas yang cerdas. Sejumlah proyek penelitian yang bertujuan untuk melakukan deteksi dan pelacakan kendaraan yang tepat telah dilakukan dan metode yang dirancang di bawah proyek-proyek ini telah menemukan kegunaannya dalam berbagai aplikasi penting, misalnya

					<p>untuk meminimalkan kecelakaan fatal yang sebagian besar terjadi karena kelalaian pengemudi atau karena jarak pandang yang buruk selama kondisi cuaca buruk atau karena pencahayaan yang tidak tepat, dll. Saat ini, beberapa jaringan syaraf tiruan telah diusulkan untuk melakukan deteksi objek. Makalah ini menyajikan tinjauan komprehensif tentang metode deteksi dan pelacakan kendaraan berbasis <i>Convolutional Neural Network Region-based /Faster R CNN</i></p>
--	--	--	--	--	---

					<p>metode deteksi serta pelacak kendaraan berbasis <i>You Only Look Once</i> (YOLO). Dalam survei ini, kami telah membagi metode pendeteksian kendaraan yang ada ke dalam beberapa kelompok yang berbeda tergantung pada arsitekturnya (<i>Faster R-CNN</i>/YOLO)</p>
7	A. FUAD JAUHARI	2023	<i>Convolutional Neural Network</i> (CNN)	<p>KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE CONVOLUTIONAL</p>	<p>Metode <i>Convolutional Neural Network</i> (CNN) mempunyai bermacam model arsitektur yang dipakai saat pengklasifikasi</p>

				<p><i>NEURAL NETWORK</i> PADA ARSITEKTUR MOBILENET</p>	<p>jenis beras didasarkan citra. Pengklasifikasi lebih mudah diterapkan apabila arsitektur CNN bisa diimplementasikan di perangkat android. Permodelan arsitektur yang dipakai di proses klasifikasi merupakan arsitektur MobileNet bermetode ekstraksi fitur (<i>Feature Extraction</i>). Arsitektur itu dipakai saat tahap pelatihan serta menguji di <i>Google Colaboratory</i>. Maka, hasil dari pelatihan ini disimpan kedalam format file <i>.tflite</i> kemudia di imporkan ke bentuk <i>assets android</i> supaya</p>
--	--	--	--	--	---

					<p>bisa diterapkan langsung ke dalam android programisasi. Sehingga hasil pengujian dan pelatihan di <i>Google Colaboratori</i> mempunyai data set berukuran 224x224 pixel. <i>Output</i> dari pelatihan arsitektur MobileNet di data set baik yang nilainya 1.0 dan akurasi dari validasi antara 0.9333. Kemudian di data set buruk akurasi pelatihannya nilainya 1.0 &amp; nilai dari akurasi validasi turun jadi 0,6889. Hasilnya pelatihan ini di android tool, nanti setiap nasi kemudian diuji lima kali dalam</p>
--	--	--	--	--	--

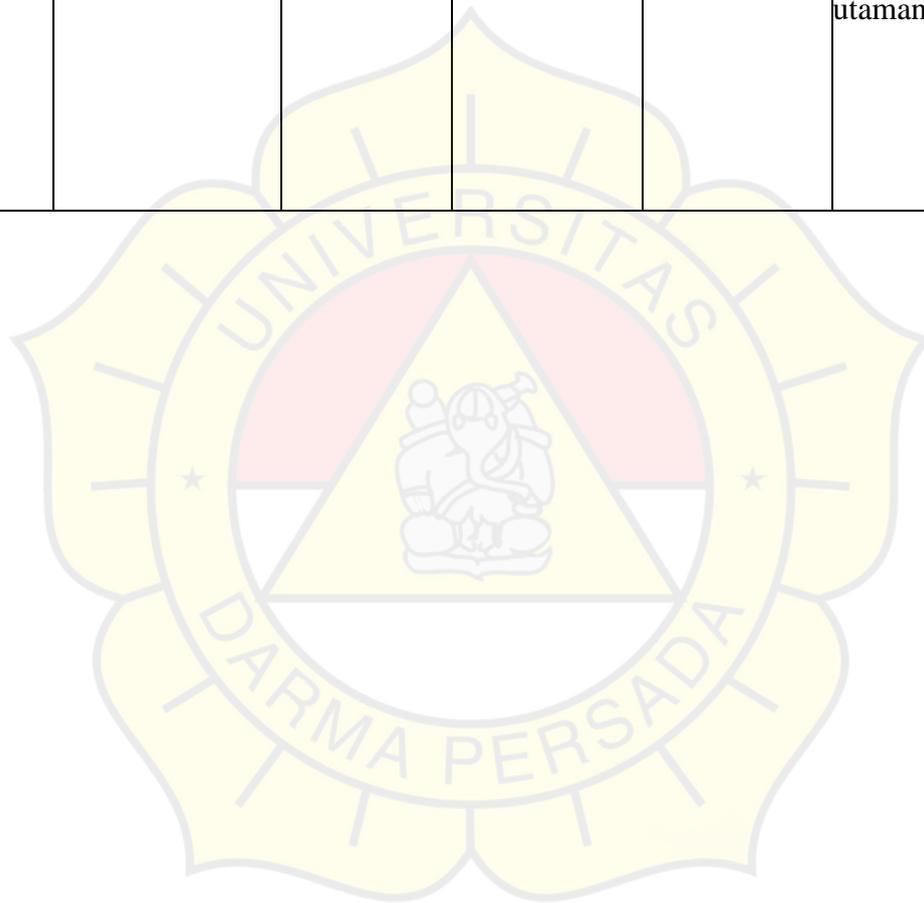
					<p>tiga kondisi pencahayaan berbeda: nasi basmati, nasi hitam, dan nasi merah, sehingga mencapai akurasinya 100% di semua model pencahayaan. Beras IR 64 memberikan hasil akurasi yaitu 60% untuk cahaya kuning, 80% untuk cahaya putih, dan 100% untuk cahaya campuran kuning dan putih. Akurasi dari Sticky Rice menyentuh 100% dengan cahaya kuning, 80% dengan cahaya putih, lalu 80% dengan campuran cahaya kuning dan putih. Faktor berpengaruh sehingga</p>
--	--	--	--	--	--

					hasil pengujian berbeda di android tools antara lain kualitas gambar, jarak objek, dan kondisi pencahayaan.
--	--	--	--	--	--

8	Lakapu, Yuliana	2019	<i>Near Infrared Spectroscopy</i>	Klasifikasi Jenis Beras Menggunakan Metode <i>Near Infrared Spectroscopy</i>	Indonesia negara produsen beras yang mempunyai bermacam jenis beras yang kualitasnya berbeda-beda. Kualitas beras itu berdasarkan susunan kimia & fisik. Lalu, NIR <i>spectroscopy</i> ialah metode penyerapan gelombang <i>infrared</i> yang dekat dengan molekul penyusun bahan. Maka, Tujuan penelitian ini ialah untuk pengklasifikasian beras berdasar jenis memakai metode <i>Near Infrared spectroscopy</i> yang kelebihanannya non destruktif dan cepat
---	-----------------	------	-----------------------------------	--	---

					<p>menganalisis.</p> <p>Pengakuisisian spektra NIR dilakukan dengan 65 sampel (ada 4 jenis beras) lewat metode difusi refleksi. Model klasifikasi dibuat memakai <i>Principal Component Analysis</i> atau (PCA) di spektra reflektan yang telah diolah lewat 3 pra perlakuan data (Normalisasi 0-1, <i>smoothing savitzky golay</i> di derivatif pertama &amp; kedua).</p> <p>Hasil analisa menunjukkan jika kombinasi metode NIR <i>Spectroscopy</i> dengan PCA pra-perlakuan data bisa mengklasifikasikan</p>
--	--	--	--	--	---

					sampel dari beras dalam empat kelompok berbeda yang total variasi Analisis PCA dari ketiga komponen utamanya ialah 99.7%
--	--	--	--	--	--



9	AR Lestiawan	2022	<i>You Only Look Once.</i> (YOLO)	DETEKSI DAN KLASIFIKAS I OBJEK KENDARAA N MENGGUNA KAN METODE YOLO	Kendaraan ialah alat yang digunakan oleh mayoritas masyarakat untuk mobilisasi atau perpindahan dari lokasi yang dekat ataupun jauh. Terdapat jenis kendaraan roda dua (sepeda motor) & kendaraan roda 4 (mobil, truk dan bus). Majunya teknologi transportasi memiliki dampak di perkembangan lalu lintas serta angkutan jalan, akan terbentuk perbedaan model perangkat lalu lintas,
---	--------------	------	--------------------------------------	---	--

					<p>sarana angkutan, dan prasarana jalan.</p> <p>Penelitian soal kecerdasan buatan atau AI (<i>artificial intelligent</i>) seperti mendeteksi dan mengklasifikasi objek kendaraan membantu kita dalam mengidentifikasi obyek dan menghitung jumlah kendaraan lewat dalam rekaman video.</p> <p>Penelitian penulis menerapkan algoritma YOLO dalam mendeteksi dan klasifikasi objek kendaraan. Sehingga penelitian memakai kumpulan data yang terdiri dari 4 tipe kelas yakni sepeda</p>
--	--	--	--	--	--

					<p>motor, mobil, bus dan truk. Hasil uji program dengan metode deteksi objek YOLO membantu membedakan sepeda motor, kendaraan beroda empat atau lebih yang bertanda dengan mendeteksi kotak hijau diantara kumpulan kendaraan pada gambar video. Berdasarkan metode deteksi objek YOLO berhasil menghitung total jumlah kendaraan yang lewat dibawah sensor</p>
--	--	--	--	--	---

					deteksi dan akurasi sebesar 79%.
10	Atriyana, Trisnawan, Wahyudi Harianto, Syahminan	2019	K-Means Clustering	Klasifikasi Beras Menggunakan Metode K-Means Clustering Berbasis Pengolahan Citra Digital	Di Indonesia padi merupakan komoditas utama penyokong makanan masyarakat. Proses pengolahan padi menjadi beras dilakukan dengan dua metode yaitu proses penumbukan dan pengolahan modern memakai gilingan. Nasi merupakan pokok makanan sehari-hari. Ada beberapa tipe yang beredar di pasaran yaitu beras IR 42, IR 64, Rojo

					<p>lele dan pandan wangi. Beragamnya bentuk dan varietas padi yang beredar di pasaran mempunyai banyak kelemahan yang dimiliki manusia ketika mengklasifikasikan varietas padi dengan menggunakan penglihatan. Makanya, perlunya teknik olah citra digital untuk mendukung analisis varietas padi. Penelitian bermaksud menganalisa jenis beras memakai metode <i>K-Means Clustering</i> didasarkan warna RGB. Sebelum perhitungan K Means</p>
--	--	--	--	--	--

					<p>wajib diproses ekstraksi fitur warna RGB agar dapat nilai <i>green</i>, nilai <i>blue</i>, nilai <i>red</i> di setiap citra. Hasil penelitian menyimpulkan ternyata <i>image processing</i> untuk menentukan jenis beras memakai metode k-means clustering bisa membantu pengguna untuk mengetahui jenis beras.</p>
--	--	--	--	--	--

## 2.2 Dokumentasi Penelitian Tempat Penjualan Beras (BULOG)

Penulis melakukan observasi penelitian ke tempat penjualan beras (bulog) pada tanggal 14 Juni 2023 yang dimana pada lokasi tersebut menjadi tempat penyimpanan beras yang dapat dijual kepada masyarakat untuk dikonsumsi memenuhi kebutuhan pangan. Disini penulis mendatangi kantor tempat penjualan beras untuk menyerahkan surat permohonan izin melakukan penelitian pada tempat tersebut, serta penulis ingin meminta sampel jenis beras untuk dijadikan objek deteksi.



**Gambar 2.1** Survei Tempat Penjualan Beras (BULOG)

Pada gambar dibawah penulis sedang melakukan wawancara terkait apa saja jenis beras yang dijual, serta jenis apa yang seringkali dibeli oleh masyarakat. Lalu menanyakan seputar tempat penjualan beras tersebut bisa berkembang dari 10 tahun yang lalu hingga sekarang.



**Gambar 2.2** Melakukan Wawancara Kepada Pemilik BULOG

Kemudian penulis melihat pegawai melakukan penjemuran beras untuk memastikan bahwa beras yang siap dijual bagus tidak ada yang rusak/bubuk.



**Gambar 2.3** Proses Penjemuran Beras

Selanjutnya beras dimasukkan kedalam karung untuk siap dipasarkan ke warung – warung atau pasar swalayan disekitar wilayah JABODETABEK, beras tersebut diantar menggunakan truk pengangkut untuk diserahkan kepada *customer*.



**Gambar 2.4** Pengangkutan Beras

### **2.3 Beras**

Menurut penelitian yang dilakukan oleh (Alhajir, 2018) dan rekan- rekannya pada tahun 2018, Beras ialah hasil utama proses menggiling padi, dimana terdiri dari bagian luar sekam yang terkelupas dan bagian dalam berupabekatul, dedak dan lembaga yang sudah dipisahkan. Peranan beras sangat penting sebagai sumber makanan utama masyarakat Indonesia, dimana total produksi mencapai 314,10 ribu ton di tahun 2020. Pemrosesan beras dimulai dari saat berbentuk gabah, selanjutnya mengalami serangkaian tahap seperti pembersihan, pengeringan, penggilingan, dan pemolesan untuk menghasilkan beras yang siap dikonsumsi. Setelah melalui tahap-tahap tersebut, beras siap untuk menjadi makanan pokok yang menyediakan nutrisi penting bagi masyarakat Indonesia.

Meskipun telah melalui proses pemrosesan, beras tetap berisiko mengandung benda asing atau impuritas contohnya beras patah, beras hitam atau bahkan kerikil.

atau beras patah yang tidak terproses sebelumnya. Untuk mengatasi hal ini, beras perlu menjalani proses pemrosesan tambahan dengan menggunakan berbagai metode seperti grading untuk memisahkan berdasarkan ukuran beras, color sortir untuk memisahkan berdasarkan warna atau destoning untuk memisahkan dari batu. Dengan melakukan proses pemrosesan tambahan ini, kualitas beras dapat ditingkatkan dan impuritas yang tidak diinginkan dapat dihilangkan, sehingga beras yang dihasilkan menjadi lebih bersih, berkualitas, dan siap untuk dikonsumsi dengan aman. Proses-proses ini menjadi langkah penting dalam memastikan bahwa beras yang dikonsumsi oleh masyarakat bebas dari benda asing dan memberikan nilai tambah dalam pangan yang aman dan bermutu.

Standarisasi SNI 6128:2020 membagi kualitas beras kedalam tiga kelas, yakni beras premium, beras medium 1, dan beras medium 2. Ketiga kelas ini sudah mencakup kualitas beras mulai dari yang terbaik hingga yang rendah, sesuai dengan parameter dan spesifikasi yang telah ditetapkan dalam standar tersebut. Standar ini menetapkan komposisi dan komponen mutu beras, dengan beras kepala menjadi komponen mutu yang paling diinginkan. Oleh karena itu, Sangat penting untuk mengembangkan algoritma rekognisi yang memungkinkan memfasilitasi proses pemisahan beras. Sehingga, algoritma itu berfungsi untuk sistem pengenalan yang memberikan instruksi pada aktuator pemisah pada mesin, sehingga mesin dapat memisahkan beras sesuai dengan standar kualitas yang diinginkan. Implementasi sistem rekognisi ini akan memberikan dukungan yang berarti bagi produsen beras dalam menciptakan mesin pemisah beras yang efektif dan sesuai dengan standar. Sehingga, hasilnya dapat dinikmati oleh konsumen dengan kualitas terjamin bagus serta sesuai dengan standarisasi yang berlaku.

## 2.4 Citra

Menurut (Alhajir, 2018) Citra (gambar) adalah sekumpulan titik yang memiliki identitas tertentu yang membentuk kesatuan gambar dengan artian bagus secara “intrinsik” ataupun "artistik". Citra yang bagus ialah citra dengan gambaran keindahan secara estetika serta jelas untuk analisis dan tujuan lainnya.

*Pixel* adalah satuan dasar dari sebuah citra. Istilah "*pixel*" ialah akronim dari "*picture element*", yang artinya elemen atau representasi titik terkecil suatu gambar grafis dan biasanya berbentuk kotak yang disebut sebagai sel. Ada 2 jenis citra, yakni citra digital dengan citra analog. Citra digital adalah citra yang dikemas dan disimpan di media penyimpanan dan bisa diolah komputer. Sedangkan, Citra analog sendiri ialah citra yang sifatnya kontinyu, bak hasil pengambilan gambar menggunakan kamera analog atau tampilan pada monitor (sinyal video) atau layar TV atau monitor.

Selain itu, citra ini dibedakan menjadi 2 bagian, yakni citra yang bergerak atau (*moving image*) dengan citra yang diam atau (*still image*). Citra diam adalah gambar statis, tetapi jika ditampilkan secara berurutan dalam urutan tertentu, akan memiliki kesan gambar kontinyu bergerak. Tiap gambar di urutannya disebut dengan frame. Sebagai contoh, tayangan televisi dan pilem layar lebar yang pada dasarnya terdiri dari jutaan frame yang ditampilkan secara berurutan untuk menciptakan efek gerakan.

### 2.4.1 Citra Digital

Menurut (Alhajir, 2018) Citra digital adalah representasi nilai citra yang awalnya bersifat kontinu, namun kemudian diubah menjadi nilai- nilai diskrit. Secara formal, citra digital kerap dipandang dengan fungsi 2D (2 Dimensi) yaitu  $f(x, y)$ , dengan ukuran berbaris & berkolumna. Di dalam citra digital, setiap titik atau piksel memiliki koordinat  $(x, y)$ , lalu nilai  $f(x, y)$  yang mewakili intensitas atau tingkat keabuan dalam posisi itu.

Citra digital itu disusun dari kumpulan titik yang umumnya berbentuk persegi panjang dan diatur teratur membentuk kolom & baris. Di tiap titiknya ini mempunyai koordinat yang umumnya dituliskan di bilangan bulat yang positif, seperti 0 dan 1, tergantung di sistem yang dipakai. Nilai piksel ini merupakan format dasar dari keseluruhan citra digital. Pada sebagian besar sistem pencitraan, nilai piksel ini umumnya berupa bilangan bulat yang positif sebagai representasi dari intensitas cahaya atau warna pada titik tersebut.

Citra digital kerap merepresentasikan berupa bentuk matriks dua dimensi yang tiap elemen matriksnya adalah piksel (*picture element*) yang mewakili irisan antara baris dan kolom dalam citra. Persamaan (1) memperlihatkan representasi citra digital kedalam matrik 2 dimensi.

Sebuah citra  $f(x,y)$  di fungsi matematik bisa dituliskan seperti ini :  $0 \leq x \leq$

$$M - 1$$

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

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

Keterangannya itu:

M = banyak baris dalam array citra N = banyak kolom dalam array citra

G = banyak skala dalam keabuan (*gray level*)

Dalam computer, citra digital dikemas dan disimpan menjadi sebuah file dengan berformat tertentu seperti bmp, jpeg, gif dan sebagainya, yang menunjukkan cara sebuah citra digital disimpan dengan kompresi atau tidak.

#### **2.4.2 Klasifikasi Citra Digital**

Citra digital dapat dibagi menjadi dua jenis berdasarkan cara penyimpanan atau pembentukannya, yakni :

1. Gambar Bitmap (*raster*) adalah gambaran nan terdiri sejumlah titik susunan gambar, juga dikenal sebagai piksel. Kualitas gambar bitmap dipengaruhi akan banyak sedikit pixel yang dipakai, sehingga semakin banyak piksel yang digunakan, sehingga kualitas gambar akan semakin bagus dan halus. Sebaliknya, jika jumlah piksel sedikit, kualitas gambar akan menurun. Gambar bitmap umumnya dihasilkan dari kamera handphone, kamera digital, scanner, & perangkat sejenis yang merekam atau mengabadikan gambar dalam bentuk titik-titik piksel.
2. Gambar vektor adalah gambar yang terdiri dari bidang, kurva dan garis. Setiap elemen ini diwakili oleh formulasi matematik. Keunggulan dari gambar vektor terletak pada kemampuannya untuk diperbesar tanpa mengorbankan kualitas gambar. Hal ini berarti gambar vektor bisa diperbesar hingga ukuran yang lebih besar tanpa mengalami kehilangan ketajaman atau distorsi. Gambar vektor umumnya dibuat memakai aplikasi berdesain vektor seperti Macromedia Freehand, Adobes Illustrator, Corel Draw,

atau perangkat lunak sejenisnya. Kelebihan dari aplikasi desain vektor ini adalah memungkinkan pengguna untuk membuat gambar dengan presisi matematis, sehingga menghasilkan grafik yang sangat halus dan dapat diubah ukurannya tanpa masalah. Hal ini menjadikan gambar vektor pilihan yang populer untuk berbagai keperluan desain, seperti logo, ilustrasi, poster, dan sebagainya.

### 2.4.3 Elemen Citra Digital

Menurut (Alhajir, 2018) Citra digital mengandung beberapa dasar elemen yang dapat dimanipulasi saat mengolah citra. Setiap elemen dasar itu ialah:

1. Kecerahan (*brightness*) adalah istilah untuk menggambarkan intensitas cahaya dalam sebuah titik (*pixel*) di sebuah citra. Sehingga intensitas kecerahan pada suatu titik sebenarnya merupakan rata-rata dari intensitas cahaya di area sekitarnya.
2. Kontras (*contrast*) Kontras dalam sebuah gambar menggambarkan perbedaan antara tingkat terang (*lightness*) dan gelap (*darkness*). Citra dengan kontras rendah memiliki komposisi dominan dari tingkat terang atau gelap, sehingga perbedaan antara area terang dan gelap kurang jelas. Sebagai hasilnya, gambar tersebut mungkin terlihat lebih datar atau kurang menonjolkan detailnya. Di sisi lain, citra dengan kontras yang baik memiliki distribusi yang merata antara tingkat terang dan gelap, sehingga perbedaan antara area terang dan gelap lebih tajam dan jelas. Hal ini menyebabkan gambar memiliki lebih banyak detail dan dimensi visual yang lebih kaya.

Gambar dengan kontras yang baik cenderung lebih menarik dan lebih mudah untuk dilihat, karena setiap elemen di dalamnya lebih terlihat dan menonjol.

3. Kontur (*contour*) adalah Perubahan intensitas cahaya pada *pixel-pixel* yang berdekatan dalam citra disebut sebagai tepi (*edge*) objek. Fenomena perubahan intensitas ini memungkinkan kita untuk mendeteksi batas antara objek atau elemen yang berbeda dalam citra. Ketika ada perbedaan yang signifikan dalam intensitas cahaya di sekitar suatu area dalam citra, kita dapat mengidentifikasi tepi objek tersebut sebagai batas visual antara dua objek atau bagian citra yang berbeda. Teknik deteksi tepi ini berguna dalam berbagai aplikasi pengolahan citra, seperti pengenalan pola, segmentasi objek, dan analisis visual.
4. Warna (*colour*) adalah *Persepsi visualisasi* mata manusia terhadap warna dipengaruhi oleh panjangnya gelombang cahaya nan memantulkan suatu benda. Semua warna memiliki kepanjangan gelombang berbeda-beda, dan cara warna dilihat dan diproses dapat menyebabkan distorsi warna pada penglihatan manusia. Distorsi warna dapat terjadi baik dalam ruang dan waktu. Misalnya, kadang-kadang kita dapat melihat bercak abu-abu di sekitar warna hijau, yang sebenarnya tampak ungu. Ini terjadi karena kesejajaran yang tidak sempurna dalam sistem penglihatan kita. Selain itu, perubahan cepat dari satu warna ke warna lainnya dapat menyebabkan efek distorsi warna, seperti ketika melihat warna hijau dan seketika melihat warna abu-abu yang tampaknya berwarna ungu. Fenomena ini merupakan hasil dari kompleksitas cara otak kita memproses informasi warna, dan mengingat

bahwa persepsi warna dapat bervariasi di antara individu. Distorsi warna ini menjadi hal yang penting untuk dipahami dalam konteks desain grafis, fotografi, dan bidang-bidang lain yang memanipulasi atau mereproduksi warna untuk menciptakan tampilan visual yang diinginkan.

5. Bentuk (*shape*) citra adalah tampilan objek dalam citra dua dimensi, meskipun objek yang diamati biasanya tiga dimensi yang diproyeksikan ke bidang dua dimensi. Pengenalan bentuk dalam citra dapat membantu untuk membedakan apakah suatu objek memiliki bentuk tertentu atau memang bentuknya asli.
6. Tekstur (*texture*) dalam citra adalah cara sistem visual manusia mempersepsikan kesatuan citra daripada informasi pada setiap titik secara terpisah. Pengenalan tekstur melibatkan kesamaan parameter citra seperti kecerahan, ukuran, dan warna, sehingga dua citra tidak dapat dianggap sama hanya berdasarkan satu parameter saja.

## **2.5 Pengolahan Citra (*Image Processing*)**

Menurut (Alhajir, 2018) Pengolahan citra merupakan proses analisis dan manipulasi citra yang melibatkan interpretasi visual. Citra digital diperoleh secara otomatis dari sistem penangkapan citra dan membentuk matriks dengan elemen yang merepresentasikan nilai intensitas cahaya atau tingkatkeabuan dari setiap *piksel*.

Meskipun citra menyimpan banyak informasi, seringkali citra mengalami penurunan mutu seperti cacat atau derau (*noise*), kontras yang terlalu tinggi, kurangnya ketajaman, atau kabur (*blurring*), dan sejenisnya. Oleh karena itu, citra yang mengalami gangguan perlu dimanipulasi agar memiliki kualitas yang lebih

baik dan dapat diinterpretasi dengan baik oleh manusia atau mesin, seperti komputer. Pengolahan citra dapat bertujuan meningkatkan kualitas citra, menonjolkan informasi tertentu dalam citra, mengelompokkan atau mengukur elemen dalam citra, atau menggabungkan bagian-bagian citra yang berbeda.

Teknik pengolahan citra menggunakan komputer untuk memproses pola bayangan dan warna pada gambar yang ada. Hasil pengolahan citra ditampilkan pada layar monitor. Tujuan utama dari pengolahan citra adalah menciptakan citra dengan kualitas yang lebih baik daripada citra asalnya. Pengolahan citra digunakan dalam berbagai bidang seperti fotografi untuk mengubah intensitas cahaya dalam foto, perfilman untuk animasi, dunia kedokteran untuk analisis medis, dan industri game video.

## **2.6 Computer Vision**

Menurut (Alhajir, 2018) *Computer Vision* adalah cabang ilmu *Artificial Intelligence* yang bertujuan untuk mengenali objek fisik dan keadaan berdasarkan gambar atau citra. Dalam *Computer Vision*, komputer berusaha untuk meniru kemampuan penglihatan manusia dan dapat melakukan beberapa hal, antarlain:

1. *Object Detection*: Mengenali objek yang ada dalam sebuah gambar dan mengetahui batas-batasnya.
2. *Recognition*: Memberikan label pada objek yang teridentifikasi.
3. *Description*: Memberikan atribut atau karakteristik tertentu pada objek.
4. *3D Inference*: Menafsirkan informasi mengenai adegan tiga dimensi dari gambar dua dimensi yang dilihat.

5. *Interpreting Motion*: Menafsirkan gerakan yang terjadi dalam sebuah gambar.

Proses dalam *Computer Vision* dapat dibagi menjadi tiga aktivitas utama, yaitu:

1. Memperoleh, mengambil, atau mengakuisisi citra.
2. Melakukan operasi pengolahan citra untuk memodifikasi data citra.
3. Menganalisis dan menginterpretasi citra dengan tujuan tertentu, seperti mengarahkan robot, mengendalikan peralatan, memantau proses manufaktur, dan lain-lain.

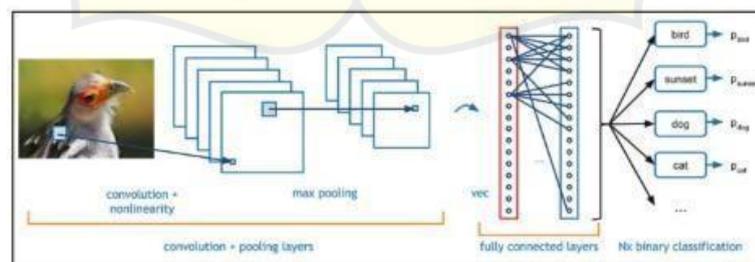
*Computer Vision* merupakan gabungan dari pengolahan citra dan pengenalan pola. Pengolahan citra dilakukan sebagai tahap awal untuk meningkatkan kualitas citra, sedangkan pengenalan pola digunakan untuk mengidentifikasi objek dalam citra. Hirarki dalam *Computer Vision* terdiri dari tiga tahap:

1. Pengolahan tingkat rendah, yang melibatkan penghilangan noise dan peningkatan kualitas gambar.
2. Pengolahan tingkat menengah, yang melibatkan deteksi batas-batas objek dalam citra.
3. Pengolahan tingkat tinggi, yang melibatkan analisis citra untuk menghasilkan informasi yang dapat dipahami oleh manusia.

Semua ini memungkinkan komputer untuk "melihat" dan memahami dunia visual seperti halnya manusia.

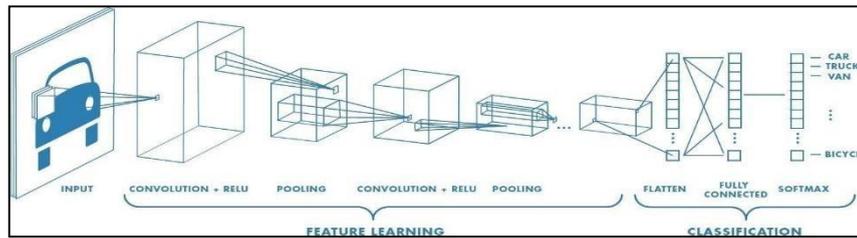
## 2.7 Convolutional Neural Network (CNN)

Menurut (Alhajir, 2018) *Convolutional Neural Network* (CNN) merupakan jenis jaringan saraf tiruan (JST) di mana *neuron-neuron* dalam layer-nya tersusun dalam bentuk tiga dimensi (O'Shea, 2015). CNN memiliki perbedaan dengan JST *konvensional*, karena setiap neuron dalam layer tertentu hanya terhubung ke wilayah kecil pada layer sebelumnya. Perbedaan utama antara CNN dan JST *konvensional* adalah bahwa CNN dirancang khusus untuk mengenali pola dan fitur dalam data yang memiliki struktur spasial, seperti gambar. Dengan terhubungnya setiap neuron hanya ke wilayah kecil pada layer sebelumnya, CNN dapat mengekstraksi fitur-fitur lokal dalam data secara efisien, sehingga memungkinkan untuk pengenalan pola yang lebih baik. Oleh karena itu, CNN telah menjadi pilihan populer dalam berbagai tugas pengenalan gambar, pengenalan objek, dan berbagai aplikasi pengolahan citra lainnya. Fokus penggunaan CNN adalah untuk memproses input berupa citra atau gambar (image). Perbedaan utama dengan *Artificial Neural Network* (ANN) adalah pada baris akhir CNN, yang secara penuh terhubung (*fully connected*), sementara dalam ANN setiap neuron terhubung dengan semua neuron lainnya. Ilustrasi arsitektur CNN dapat dilihat pada gambar.



**Gambar 2.5** Ilustrasi Arsitektur CNN (*sumber* : Alfath Daryl Alhajir, dkk., 2021)

Secara garis besar proses operasi CNN dapat dilihat pada gambar dibawah ini.

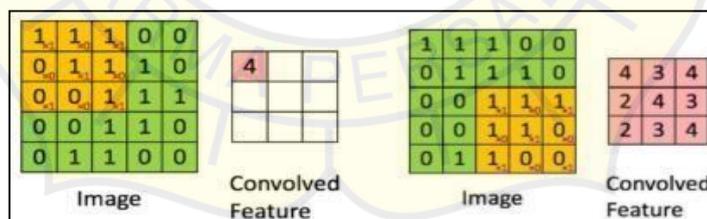


**Gambar 2.6** Proses CNN (sumber : Sena, 2017 )

Berdasarkan gambaran di atas bisa dilihat terdapat 4 proses operasi utama yang dikerjakan pakai metode CNN. Maka, 4 proses operasi dijelaskan seperti berikut :

### 2.7.1 Convolutional

Tujuan utama dari *Convolutional Neural Network* (CNN) adalah untuk mengekstraksi fitur dari gambar yang masuk. CNN bekerja dengan belajar soal fitur gambar memakai kotak kecil yang disebut kernel atau filter untuk mengenali pola atau pixel di data yang masuk (Keijzers, 2010). Cara kerja *convolutional* bisa dilihat dalam gambar dibawah ini :



**Gambar 2.7** Operasi Konvolusi (sumber : I Wayan Suartike E.P, 2016)

Pada matriks 3x3, misalnya, elemen perkalian dihitung, dan hasil perkaliannyaditambahkan agar menghasilkan bilangan yang bulat dalam bentuk 1 elemen (Suartika E. P, I Wayan, Wijaya Arya Yudhi, 2016). Matriks 3x3 yang digunakan dalam teknik CNN jugadisebut sebagai "filter", "kernel", atau "*detektor fitur*", dan matriks yang dibuat dengan menggesekkan filter pada gambar dan

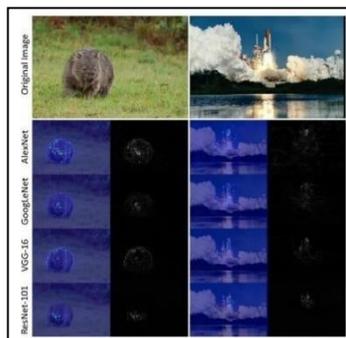
mampu menghitung titik-titik filter disebut sebagai "fitur konvolusi", "peta aktivasi", atau "*peta fitur*". Sebuah fungsi  $F$  secara matematis direpresentasikan dengan menggunakan teknik sebagai jumlah lingkaran pada sudut-sudutnya, yang kemudian digerakkan melintasi fungsi  $g$  untuk menghasilkan fungsi  $h$ . Berikut adalah persamaan (3) yang digunakan dalam operasi *convolution*:

$$h(x, y) = F(x, y) \otimes g(x, y) \quad (1)$$

Filter berfungsi sebagai pendeteksi fitur dari gambar yang dimasukkan. Dalam metode Convolutional Neural Network (CNN), operasi konvolusi dilakukan pada seluruh gambar menggunakan filter. Hasil akhir dari operasi konvolusi ini disajikan dalam bentuk fitur map. Filter dalam CNN hanya berkaitan dengan dimensi (panjang kali lebar) dan jumlah filter dalam setiap layer.

Isi nilai pada filter sebuah CNN disebut "*weighting value*." Nilai bobot ini awalnya diambil secara *randomisasi* dan berubah selama prosesi pelatihan atau *training*. Proses *training* ini bertujuan untuk mengoptimalkan nilai bobot sehingga CNN dapat belajar dan mengekstraksi fitur-fitur yang relevan dari gambar dengan akurat.

Hasil fitur map merupakan representasi visual dari fitur-fitur yang berhasil dideteksi oleh CNN pada gambar. Gambaran ini memperlihatkan bagaimana CNN mengidentifikasi berbagai fitur dan pola dalam gambar melalui filter dan proses konvolusinya.



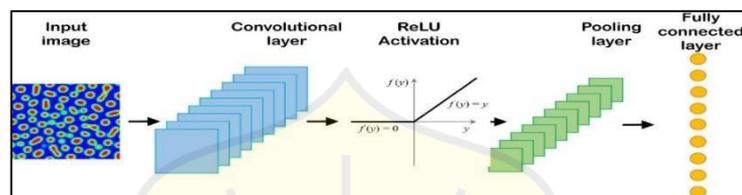
**Gambar 2.8** *Feature Map* (sumber : Alfi Salim)

### 2.7.2 *Rectified Linear Unit (ReLU)*

ReLU merupakan sebuah operasi yang dipakai setelah setiap operasi konvolusi dalam metode CNN. ReLU adalah operasi non-linear dalam fungsi aktivasi yang memungkinkan peningkatan kemampuan pemodelan jaringan. Persamaan ReLU bisa dilihat kedalam persamaan (3) berikut (Nair & Hinton, 2010). *jika  $x \leq 0$  maka  $x = 0$  dan apabila  $x > 255$  maka  $x = 255$*  (2)

Operasi ReLU (Rectified Linear Unit) ialah operasi pengubah semua nilai pixel negatif di dalam fitur map menjadi bernilai nol. Tujuannya dari penggunaan ReLU ialah untuk mengenalkan fungsi non linear pada metode *Convolutional Neural Network (CNN)*. Fungsi non linear ini sangat penting karena sistem non-linear memiliki sifat yang tidak linier, dinamis, sulit dikendalikan, sulit diprediksi. Sistemisasi non-linear memiliki tingkat sensitivitas yang lebih tinggi dan cenderung lebih mampu menangkap kompleksitas data. Dalam implementasi CNN, ReLU digunakan sebagai fungsi aktivasi untuk menambahkan elemen non-linearitas pada aliran data. Hal ini memungkinkan CNN untuk mempelajari dan mengekstraksi fitur-fitur yang lebih kompleks dan memiliki kemampuan representasi yang lebih kuat. Penggunaan ReLU secara luas digunakan dalam

bidang kecerdasan buatan dan pengolahan citra karena efisiensinya dan kemampuannya untuk mengatasi masalah non-linear. Gambar di bawah ini menjelaskan proses ReLU yang dapat dilihat sebagai representasi visual dari bagaimana operasi tersebut mengubah nilai piksel negatif menjadi nol. (Nair & Hinton, 2010).



**Gambar 2.9** Proses Operasi ReLU (*sumber : mukhlisatun nada, 2019*)

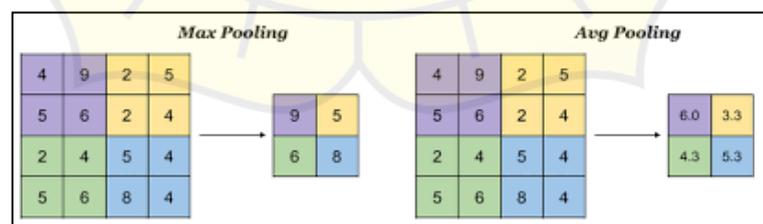
Dari gambar di atas, terlihat sebuah proses di mana nilai yang mewakili warna hitam akan diubah menjadi warna abu-abu. Di sisi lain, semua nilai yang mewakili warna putih dianggap bernilai positif dan tidak mengalami perubahan. Proses ini mungkin menggambarkan suatu teknik pengolahan citra yang disebut sebagai thresholding atau ambang nilai. Dalam thresholding, piksel-piksel dengan nilai di bawah ambang tertentu diubah menjadi warna abu-abu atau hitam, sementara piksel-piksel dengan nilai di atas ambang tetap tidak berubah atau tetap putih. Teknik ini sering digunakan untuk mengkonversi citra menjadi citra biner (hitam putih), atau untuk menonjolkan objek atau fitur tertentu dalam citra dengan cara mengubah latar belakang menjadi abu-abu atau hitam. Penggunaan teknik thresholding bergantung pada tujuan dan karakteristik citra yang akan diproses. Itu dapat menjadi langkah awal yang penting dalam analisis citra dan pengolahan selanjutnya.

### 2.7.3 Pooling

*Pooling* juga biasa disebut dengan *subsampling* atau *downsampling*, ada beberapa jenis proses pooling yang dapat digunakan, seperti:

1. *Max*, cara kerjanya yaitu mencari nilai maksimal dari nilai piksel yang didapatkan dari proses ReLU.
2. *Average*, cara kerja dari *Average* adalah dengan mencari nilai rata – rata dari nilai piksel yang dihasilkan dari proses ReLU.
3. *Sum*, cara kerjanya yaitu dengan menjumlahkan nilai yang didapat dari nilai piksel yang telah diproses ReLU.

Dari ketiga jenis tersebut proses *max pooling* dikatakan lebih bagus dan populer dibandingkan yang lain dalam pengimplementasiannya (Redmon, 2016). *Max pooling* juga mendapatkan nilai yang baik dalam perhitungan range yang besar dari pada *average* atau *min pooling* (Boureau et al., 2010). Dibawah ini adalah contoh operasi *max pooling* dengan menggunakan matrik 2x2 dalam fitur map (diperoleh setelah proses operasi *convolutional* dan ReLU) dapat dilihat pada gambar berikut (Suartika E. P, I Wayan, Wijaya Arya Yudhi, 2016):



**Gambar 2.10** Operasi *Max Pooling* (sumber : lery sakti ramba,2020)

Pada gambar diatas dapat dilihat bahwa proses *max pooling* bekerja dengan menggeser matrik 2x2, hal ini disebut dengan *stride*. Proses ini dilakukan dengan

Cara memilih nilai tertinggi dari setiap matrik  $2 \times 2$  Menjadi satu kolom matrik.

Untuk proses *max pooling* dapat dijelaskandengan persamaan (4) berikut :

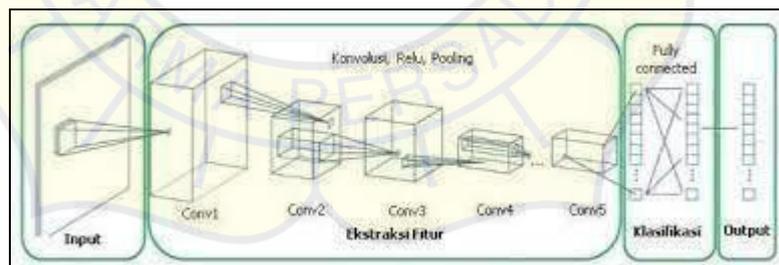
Input :  $N \times N$

Matrik konvolusi :  $k \times k$  Output :  $N/k \times N/k$

Dimana hasil nilai matrik  $k \times k = \text{nilai max}$

#### 2.7.4 Fully Connected Layer

*Fully connected layer* adalah proses traditional *Multi Layer Perceptron* (MLP) yang merupakan proses layer terakhir pada metode CNN. *Fully connected layer* dapat diproses setelah proses *convolutional*, ReLU, pooling, selesai dikerjakan. Tujuan dari *fully connected layer* ini adalah mengklasifikasikan gambar masukan ke dalam berbagai kelas berdasarkan kelas – kelas dataset yang mana memanfaatkan proses – proses yang telah diselesaikan dari proses *convolutional*, ReLU, dan *pooling*.



**Gambar 2.11** Proses Klasifikasi *Fully Connected Layer*

(sumber : Nur Ibrahim, 2022 )

Menurut (Alhajir, 2018) *Fully connected layer* ini adalah proses terakhir yang dilakukan dalam metode CNN. Cara kerja *fully connected layer* ini adalah mencocokkan nilai – nilai matrik yang berasal dari proses *pooling* dengan nilai matrik yang ada pada *dataset*. Dengan dilakukannya proses tersebut maka *fully connected layer* dapat mengklasifikasikan nilai *dataset* mana yang sesuai dengan nilai yang telah di masukkan. Persamaan *fully connected layer* dapat dilihat pada persamaan.

$$object = xa - xb$$

(Semua nilai negative (-) diubah menjadi positif (+))

Keterangan :

$xa$  = matrik uji

$xb$  = matrik latih

*Fully connected layer* ini juga dapat dilakukan beberapa kali dalam sebuah CNN, yang mana pada kasus penelitian ini dilakukan 2 kali. Nilai matrik mendapatkan pertimbangan untuk diseleksi dan dicocokkan pada dataset yang telah diberikan. Hasil akhir bergantung pada hasil prediksi, seberapa dekat nilai input dengan nilai yang ada pada dataset.

Secara umum tipe lapisan pada CNN dibagi menjadi dua yaitu:

1. Layer ekstraksi fitur (*feature extraction layer*) Lapisan pertama dari gambar arsitektur terdiri dari banyak lapisan, yang masing-masing terdiri dari neuron yang terhubung ke area lokal lapisan sebelumnya. Lapisan konvolusi adalah jenis lapisan awal, diikuti oleh lapisan penggabungan. Fungsi

aktivasi diterapkan pada setiap lapisan, dengan peralihan posisi antara jenis pertama dan jenis kedua. Gambar input diterima secara langsung oleh lapisan ini, yang memprosesnya untuk membuat output vektor yang akan ditangani di lapisan berikutnya.

2. Layer klasifikasi (*classification layer*), terdiri dari banyak lapisan, dan setiap lapisan terdiri dari neuron yang benar-benar terhubung dengan lapisan lainnya. Lapisan ini mendapatkan input vektor dari lapisan ekstraksi fitur gambar dan mengubahnya menjadi Jaringan Syaraf Tiruan dengan memasukkan banyak lapisan tersembunyi. Akurasi kelas untuk klasifikasi adalah hasil keluarannya.

## 2.8 *Object Detection*

*Object detection* menetapkan keberadaan item dalam gambar, serta cakupan dan penempatannya. Ini adalah sistem pengenalan objek dua kelas, dengan satu kelas mewakili kelas objek dan kelas lainnya mewakili kelas non-objek. Deteksi lunak dan deteksi keras adalah dua jenis pendeteksian objek. Deteksi lunak hanya mengidentifikasi keberadaan objek, sedangkan deteksi keras mendeteksi keberadaan dan penempatan objek (Jalled & Voronkov, 2016). Secara umum, kerangka kerja pendeteksian objek terdiri dari tiga proses utama, yaitu:

1. Pertama, Untuk menghasilkan wilayah yang diminati atau saran wilayah, model atau algoritma digunakan. Saran wilayah ini terdiri dari sejumlah besar kotak pembatas yang menjangkau seluruh gambar (yaitu, komponen pelokalan objek).
2. Pada langkah kedua, fitur visual diekstraksi untuk setiap kotak pembatas, mereka dievaluasi dan ditentukan apakah dan objek mana yang ada dalam

proposal berdasarkan fitur visual (yaitu komponen klasifikasi objek).

3. Pada langkah akhir Selama pasca-pemrosesan, kotak yang tumpang-tindih digabungkan ke dalam satu kotak pembatas tunggal dengan menggunakan penekanan non-maksimum.

## **2.9 Algoritma Faster R-CNN (*Faster Region-Convolutional Neural Network*)**

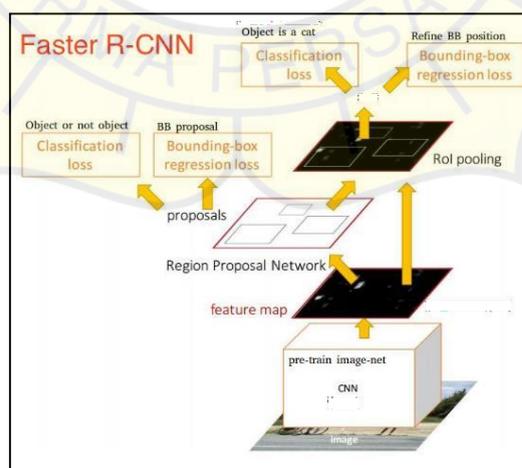
Menurut (Alhajir, 2018) *Faster R-CNN* adalah sistem deteksi objek ini terdiri dari dua modul yang saling berinteraksi. Modul pertama adalah *Deep Fully Convolutional Network* yang bertugas untuk mengusulkan wilayah-wilayah potensial yang mengandung objek. Modul kedua adalah *Fast R- CNN detector* yang menggunakan wilayah-wilayah usulan dari modul pertama untuk melakukan deteksi objek secara lebih akurat. Kedua modul ini beroperasi sebagai satu jaringan terpadu yang bekerja sama untuk mendeteksi objek. Modul RPN (*Region Proposal Network*) memberikan informasi kepada modul *Fast R- CNN* tentang wilayah-wilayah mana yang perlu diperhatikan atau dicari untuk proses deteksi lebih lanjut. Dalam keluarga algoritma R-CNN, *evolusi* antar versi biasanya berkaitan dengan efisiensi komputasi (mengintegrasikan tahap pelatihan yang berbeda), pengurangan waktu tes, dan peningkatan kinerja (MAP). Jaringan ini biasanya terdiri dari (Ananth, 2019):

1. Algoritma proposal wilayah untuk menghasilkan “*bounding box*” atau lokasi objek yang mungkin dalam gambar.
2. Tahap pembuatan *feature* untuk mendapatkan *feature* dari objek ini, biasanya menggunakan CNN.
3. Lapisan klasifikasi untuk memprediksi kelas mana dari objek ini.

4. Lapisan regresi untuk membuat koordinat kotak pembatas objek lebih tepat.

Di Fast R-CNN, satu-satunya bagian jaringan yang berdiri sendiri adalah algoritma proposal wilayah. R-CNN dan Fast R-CNN menggunakan algoritma proposal wilayah berbasis CPU seperti *Algoritma Selective Search*, yang memerlukan waktu sekitar 2 detik per gambar dan berjalan pada komputasi CPU. Namun, Faster R-CNN memperbaiki masalah ini dengan menggunakan jaringan konvolusi lain yang disebut RPN (*Region Proposal Network*) untuk menghasilkan proposal wilayah. Perubahan ini tidak hanya mengurangi waktu proposal wilayah dari 2 detik menjadi hanya 10 ms per gambar, tetapi juga memungkinkan tahap proposal wilayah untuk berbagi lapisan dengan tahap deteksi berikutnya, yang menyebabkan peningkatan keseluruhan dalam representasi fitur. Dalam sisa artikel, istilah "Faster R-CNN" biasanya merujuk pada pipeline deteksi yang menggunakan RPN sebagai algoritma proposal wilayah, sementara Faster R-CNN berfungsi sebagai jaringan detektor.

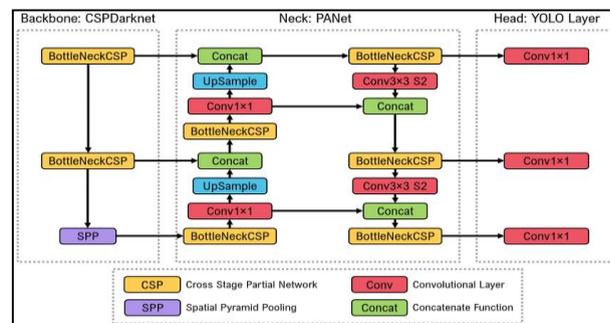
### 2.9.1 Arsitektur *Faster R-CNN*



**Gambar 2.12** Arsitektur Faster R-CNN ( sumber : Richard Dharmadi, 2018)

## 2.10 Arsitektur YOLO v5

YOLO v5 berbeda dari edisi sebelumnya, menurut (Anora, 2021). Alih-alih Darknet, YOLO v5 menggunakan PyTorch. CSP Darknet 53 berfungsi sebagai tulang punggung untuk YOLO v5. *Backbone* ini menghilangkan duplikasi informasi pada gradien di *backbone* dan mengintegrasikan perubahan gradien ke dalam peta fitur, mengurangi parameter untuk meningkatkan kecepatan inferensi, akurasi, dan ukuran model. Untuk meningkatkan aliran informasi, YOLO v5 menggunakan *Path Aggregation Network* (PANet) sebagai penghambat. PANet menggunakan *Jaringan Piramida Fitur* (FPN) yang baru, yang memiliki banyak lapisan dari *bawah ke atas* dan *dari atas ke bawah*. Hal ini dapat membantu penyebaran fitur tingkat rendah model. PANet meningkatkan lokalisasi lapisan bawah, yang meningkatkan akurasi lokalisasi objek. Selain itu, *head* pada YOLO v5 sama dengan *head* YOLO v4 dan YOLO v3, yang membuat tiga output peta fitur terpisah untuk mencapai *skala multi-prediksi*. Hal ini juga membantu dalam prediksi yang efisien dari benda-benda kecil hingga besar dalam model. Gambar dikirim ke CSP Darknet 53 untuk ekstraksi fitur sebelum dikirim kembali ke PANet untuk penggabungan sifat tersebut. Akibatnya, hasilnya dihasilkan oleh lapisan YOLO. Grafik berikut menggambarkan arsitektur YOLO v5.

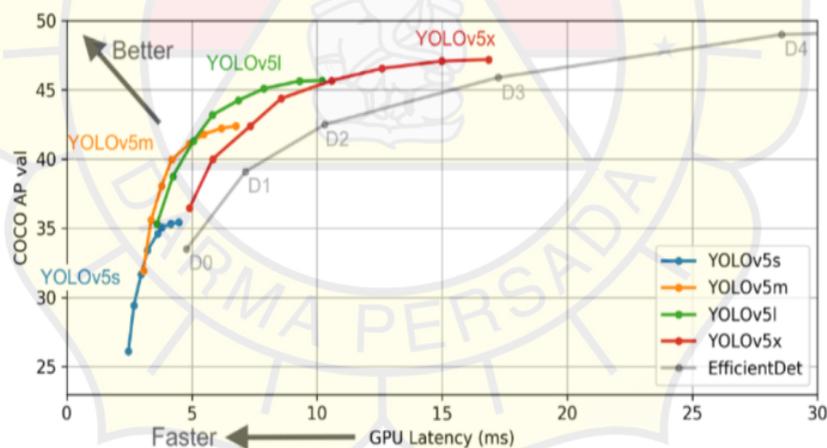


**Gambar 2.13** Arsitektur YOLO v5 ( sumber : C.Y, Wang et al., )

Conv adalah singkatan dari lapisan konvolusi. C3 terdiri dari tiga lapisan konvolusi dan modul yang mengalir di atas penghalang yang berbeda. *Spatial Pyramid Pooling* (SPP) adalah lapisan penyatuan (*pooling*) yang digunakan untuk mengurangi batas ukuran tetap jaringan. Upsample digunakan untuk meningkatkan ukuran fungsi lapisan sebelumnya pada node yang dekat. Concat adalah lapisan pengiris yang mengiris lapisan sebelumnya. Tiga Conv2d terakhir adalah modul pendeteksi objek yang digunakan pada inti jaringan.

### 2.10.1 Tipe YOLO v5

YOLO v5 memiliki banyak varian, masing-masing dengan kecepatan deteksi dan kinerja mAP yang berbeda, menurut (Anora, 2021). Grafik di bawah ini menggambarkan perbedaan antara setiap versi YOLO v5.



**Gambar 2.14** Performa Tipe Model YOLO v5 ( Sumber : Jurnal A Levina, 2021)

**Tabel 2.4 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 2.21 Semakin tinggi posisi di grafik, semakin tinggi kualitas model yang diukur dengan Model Average Precision (MAP). Namun, semakin ke kiri posisi, pendeteksian objek akan semakin cepat. Tabel 2.4 mencantumkan beberapa indikasi yang perlu dipertimbangkan ketika memilih model. Jenis model YOLO v5 dilatih menggunakan data COCO dengan ukuran 640 piksel. Jika Anda melihat *Speed* atau kecepatannya, maka akan lebih besar jika *parameternya* juga lebih besar. Parameter adalah variabel pelatihan. Akibatnya, jika parameter semakin besar maka komputasi akan semakin lama dan pendeteksian akan semakin lambat.

## 2.11 Ultralytics YOLOv8

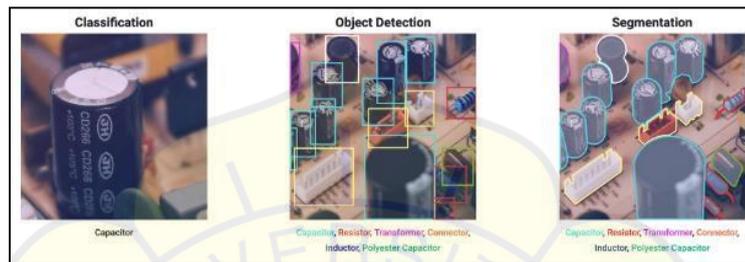
Menurut Glenn-Jocher, Sergiuwaxmann, 2023 Ultralytics YOLOv8 adalah versi terbaru dari model deteksi objek YOLO dan segmentasi gambar yang dikembangkan oleh Ultralytics. YOLOv8 merupakan model state-of-the-art (SOTA) yang dibangun di atas kesuksesan versi-versi sebelumnya dari YOLO, serta memperkenalkan fitur-fitur dan peningkatan baru untuk meningkatkan kinerja dan fleksibilitasnya.

Salah satu fitur utama YOLOv8 adalah kemampuannya yang sangat ekstensibel. Model ini didesain sebagai kerangka kerja yang mendukung semua versi sebelumnya dari YOLO, sehingga pengguna dapat dengan mudah beralih antara versi yang berbeda dan membandingkan performanya. Hal ini membuat YOLOv8 menjadi pilihan ideal bagi mereka yang ingin menggunakan teknologi YOLO terbaru, tetapi tetap dapat memanfaatkan model YOLO yang sudah mereka miliki.

Selain kemampuan ekstensibilitas, YOLOv8 juga menawarkan sejumlah inovasi lain yang membuatnya menarik untuk berbagai tugas deteksi objek dan segmentasi gambar. Inovasi-inovasi ini meliputi jaringan tulang punggung baru, kepala deteksi bebas jangkar baru, dan fungsi kerugian baru. YOLOv8 juga sangat efisien dan dapat dijalankan di berbagai platform perangkat keras, mulai dari CPU hingga GPU. Secara keseluruhan, YOLOv8 merupakan alat yang kuat dan fleksibel untuk deteksi objek dan segmentasi gambar, dengan menyajikan teknologi terbaru (SOTA) dan kemampuan untuk memanfaatkan serta membandingkan semua versi YOLO sebelumnya.

### 2.11.1 Klasifikasi

Menurut Glenn-Jocher, Sergiuwaxmann, 2023 Klasifikasi gambar merupakan tugas yang paling dasar dari ketiga tugas tersebut, dan melibatkan pengelompokan seluruh gambar ke dalam beberapa kelas yang telah ditentukan sebelumnya.



**Gambar 2.15** Klasifikasi YOLOv8 ( sumber : *Augmented Startups, 2023* )

Output dari pengklasifikasi gambar adalah label kelas tunggal dan skor kepercayaan yang menunjukkan sejauh mana gambar tersebut cocok dengan setiap kelas. Klasifikasi gambar berguna ketika Anda hanya ingin mengetahui kelas apa yang dimiliki gambar tanpa perlu mengetahui lokasi atau bentuk objek dalam gambar tersebut.

#### 1. Tip

Model klasifikasi YOLOv8 menggunakan akhiran "-cls", yang dikenal sebagai "yolov8n-cls.pt", dan telah dilatih sebelumnya pada dataset ImageNet.

#### 2. Train

Model YOLOv8n-cls dilatih pada dataset MNIST160 selama 100 epoch dengan ukuran gambar 64. Untuk melihat daftar lengkap argument yang tersedia, Anda dapat merujuk ke halaman konfigurasi.

Contoh Python :

```
from ultralytics import YOLO

# Load a model

model = YOLO("yolov8n-cls.yaml") # build a new model from scratch
model = YOLO("yolov8n-cls.pt") # load a pretrained model (recommended
for training)

# Train the model

model.train(data="mnist160", epochs=100, imgsz=64)
```

Contoh CLI :

```
yolo classify train data=mnist160 model=yolov8n-cls.pt epochs=100
imgsz=64
```

### 3. Val

Untuk memvalidasi akurasi model YOLOv8n-cls yang telah dilatih pada set data MNIST160, tidak diperlukan argumen tambahan karena model menyimpan data pelatihan dan argumen sebagai atribut model itu sendiri.

Contoh Python :

```
from ultralytics import YOLO

# Load a model

model = YOLO("yolov8n-cls.pt") # load an official model
model = YOLO("path/to/best.pt") # load a custom model

# Validate the model

metrics = model.val() # no arguments needed, dataset and settings
remembered

metrics.top1 # top1 accuracy
```

```
metrics.top5 # top5 accuracy
```

Contoh CLI :

```
yolo classify val model=yolov8n-cls.pt # val official model
```

```
yolo classify val model=path/to/best.pt # val custom model
```

#### 4. Predict

Menggunakan model YOLOv8n-cls yang telah dilatih, kita dapat melakukan prediksi pada gambar.

Contoh Python :

```
from ultralytics import YOLO

# Load a model
model = YOLO("yolov8n-cls.pt") # load an official model
model = YOLO("path/to/best.pt") # load a custom model

# Predict with the model
results = model("https://ultralytics.com/images/bus.jpg") # predict on an
image
```

Contoh CLI :

```
yolo classify predict model=yolov8n-cls.pt

source="https://ultralytics.com/images/bus.jpg" # predict with official
model
```

```
yolo classify predict model=path/to/best.pt

source="https://ultralytics.com/images/bus.jpg" # predict with custom
model.
```

## 5. Export

Model YOLOv8n-cls dapat diekspor ke format lain seperti ONNX, CoreML, dan sebagainya.

Contoh Python :

```
from ultralytics import YOLO

# Load a model

model = YOLO("yolov8n-cls.pt") # load an official model

model = YOLO("path/to/best.pt") # load a custom trained

# Export the model

model.export(format="onnx")
```

Contoh CLI :

```
yolo export model=yolov8n-cls.pt format=onnx # export official model

yolo export model=path/to/best.pt format=onnx # export custom trained

model
```

Format ekspor YOLOv8-cls yang tersedia meliputi :

Format	Format	Model
PyTorch	-	yolov8n-cls.pt
TorchScript	torchscript	yolov8n-cls.torchscript
ONNX	onnx	yolov8n-cls.onnx
OpenVINO	openvino	yolov8n-cls_openvino_model/
TensorRT	engine	yolov8n-cls.engine
CoreML	coreml	yolov8n-cls.mlmodel

TensorFlow SavedModel	save_model	yolov8n-cls_save_model/
TensorFlow GraphDef	pb	yolov8n-cls.pb
TensorFlow Lite	tflite	yolov8n-cls.tflite
TensorFlow Edge TPU	edgetpu	yolov8n- cls_edgetpu.tflite
TensorFlow.js	tfjs	yolov8n-cls_web_model/
PaddlePaddle	paddle	yolov8n- cls_paddle_model/

### **2.11.2 Image Annotation**

Ketepatan performa kecerdasan buatan sangat bergantung pada akurasi data pelatihan yang digunakan. Salah satu teknik utama yang digunakan untuk menciptakan data pelatihan dalam bidang visi komputer adalah anotasi gambar. Dengan menggunakan gambar yang telah diberi anotasi, mesin dapat diajarkan untuk melihat dunia sebagaimana manusia melakukannya. Proses anotasi gambar melibatkan penentuan wilayah secara manual dalam gambar dan pembuatan deskripsi berbasis teks dari wilayah tersebut. Ini merupakan langkah penting dalam membangun dasar yang akurat untuk melatih model visi komputer. Anotasi gambar digunakan dalam berbagai kasus penggunaan, termasuk dalam penelitian ini untuk deteksi dan pengenalan rambu lalu lintas otomatis (Morikawa,2019).

Anotasi adalah proses pelabelan data dalam berbagai media seperti gambar, teks, atau video dalam machine learning. Label-label ini biasanya telah ditentukan sebelumnya dan digunakan untuk memberikan informasi pada model visi komputer tentang objek yang terdapat dalam gambar. Algoritma kemudian menggunakan data beranotasi ini untuk belajar dan mengenali pola-pola serupa ketika diberikan data baru. Tergantung pada sifat proyek dan kebutuhan industri, berbagai bentuk anotasi dibutuhkan untuk mendukung pengembangan model visi komputer yang akurat dan efektif.

### **2.11.3 Bounding Box**

Salah satu jenis anotasi gambar yang paling umum dan sederhana adalah kotak pembatas (bounding box). Dalam bentuk anotasi ini, labeller harus menggambar kotak yang melingkupi objek utama dalam gambar sebaik mungkin. Penggunaan kotak pembatas 2D sering digunakan dalam berbagai industri untuk tujuan klasifikasi objek, pelokalisasi, dan deteksi objek.

Kotak pembatas digunakan untuk memberikan informasi lokasi dan ukuran objek dalam gambar. Dengan memberi anotasi kotak pembatas pada gambar, mesin dapat belajar untuk mengenali dan memahami posisi relatif objek terhadap gambar secara lebih akurat. Teknik anotasi ini banyak digunakan dalam aplikasi pengenalan objek, segmentasi gambar, dan berbagai tugas computer vision lainnya.

Dalam beberapa kasus, anotasi kotak pembatas bisa menjadi langkah awal untuk mendukung pengembangan model visi komputer yang lebih kompleks. Anotasi ini memberikan fondasi bagi algoritma machine learning untuk belajar dan mengenali pola serta fitur dari objek-objek dalam gambar.



**Gambar 2.16** Contoh Penggunaan Bounding Box ( *Sumber: Low, 2020* )

#### **2.11.4 Polygon Annotation**

Anotasi poligon menjadi penting karena tidak setiap objek dapat diakomodasi secara tepat oleh kotak pembatas. Metode anotasi poligon digunakan untuk mendapatkan representasi yang lebih akurat untuk tempat-tempat yang memiliki bentuk tidak beraturan, seperti objek non-simetris pada gambar udara seperti buah, pohon, bangunan terkenal, atau rumah.

Dalam kasus-kasus seperti ini, anotasi poligon memungkinkan labeller untuk menggambar garis yang mengikuti kontur objek dengan presisi yang tinggi, sehingga lebih sesuai dengan bentuk objek yang sebenarnya. Metode ini memerlukan tingkat kepresisian yang lebih tinggi dan seringkali melibatkan usaha lebih besar daripada anotasi kotak pembatas. Namun, hasilnya lebih akurat dan relevan, sehingga memberikan informasi yang lebih kaya kepada model visi komputer.

Anotasi poligon sering digunakan dalam berbagai aplikasi seperti deteksi objek, segmentasi gambar, dan pemetaan objek pada data citra. Penggunaannya membantu meningkatkan kualitas dan efektivitas model kecerdasan buatan dalam

mengenali dan memahami objek-objek yang kompleks dan beraneka ragam dalam gambar.



**Gambar 2.17** Contoh Penggunaan Polygon Annotation (Sumber: Low, 2020)

### **2.11.5 Point Annotation**

Anotasi titik melibatkan menempatkan titik-titik utama secara akurat pada lokasi tertentu dalam gambar. Bentuk anotasi ini paling umum digunakan untuk pengenalan wajah dan analisis sentimen. Dengan mengidentifikasi dan mengikuti pergerakan titik-titik kunci pada ekspresi wajah, algoritma pembelajaran mesin dapat mendeteksi emosi melalui pembacaan prediktif.

Penggunaan anotasi titik pada gambar memungkinkan model visi komputer untuk lebih fokus pada fitur-fitur utama dalam gambar, seperti posisi mata, hidung, mulut, dan titik-titik lain yang dapat mengindikasikan ekspresi wajah dan emosi. Proses anotasi ini memerlukan kepresisian yang tinggi untuk memastikan titik-titik tersebut ditempatkan dengan tepat, sehingga memberikan dasar yang akurat bagi model untuk belajar dan mengenali pola-pola emosi.

Anotasi titik sangat bermanfaat dalam berbagai aplikasi yang memerlukan analisis ekspresi wajah dan emosi, seperti dalam pengenalan wajah untuk autentikasi, deteksi kebohongan, dan analisis perasaan dalam pengalaman pengguna. Dengan mengumpulkan data beranotasi titik yang tepat, model kecerdasan buatan dapat lebih akurat dalam mengidentifikasi dan memahami emosi manusia dari data gambar.



**Gambar 2.18** Contoh Penggunaan Poin Annotation ( *Sumber: Low, 2020* )

#### **2.11.6 Semantic Annotation**

Segmentasi semantik adalah teknik yang biasanya digunakan untuk deteksi dan lokalisasi objek-objek tertentu dalam gambar. Metode ini bertujuan untuk memahami dan mengidentifikasi area yang berbeda dalam gambar serta memberikan label pada setiap piksel berdasarkan kategori yang tepat.

Penerapan pemahaman gambar yang terperinci seperti segmentasi semantik sering ditemukan di berbagai industri, dan ini sangat populer di industri kendaraan otonom. Kendaraan otonom memerlukan pemahaman yang mendalam tentang lingkungannya untuk beroperasi secara aman dan efisien. Dengan menggunakan segmentasi semantik, mobil otonom dapat mengenali dan memahami objek-objek

seperti kendaraan lain, pejalan kaki, rambu lalu lintas, dan lainnya dalam sekitarnya.

Di industri Agritech, segmentasi semantik juga digunakan untuk analisis ladang tanaman. Teknik ini dapat membantu mendeteksi penyakit pada tanaman, mengidentifikasi pertumbuhan yang tidak normal, atau bahkan memantau perkembangan tanaman secara keseluruhan. Dengan menggunakan segmentasi semantik, analisis ladang tanaman dapat dilakukan secara otomatis dan akurat, membantu petani dalam pengelolaan dan pemantauan pertanian.

Secara keseluruhan, segmentasi semantik merupakan alat yang kuat dalam pengolahan citra dan pengenalan objek, dan telah mendukung kemajuan dalam berbagai bidang termasuk transportasi otonom dan pertanian pintar.



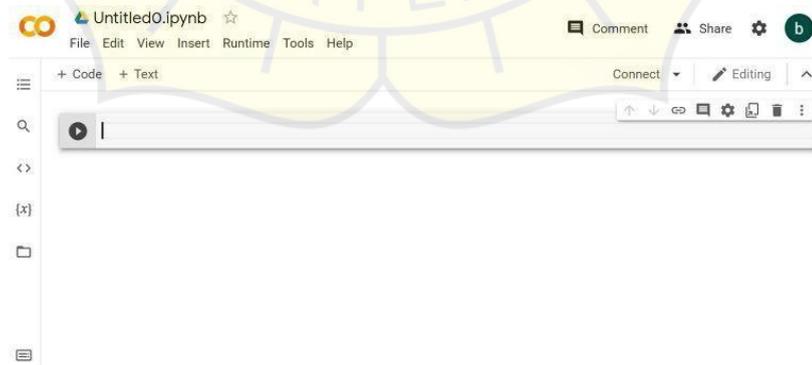
**Gambar 2.19** Contoh Penggunaan Semantic Annotation (*Sumber: Low, 2020*)

## **2.12 Google Colaboratory**

*Colaboratory*, atau "Colab" untuk lebih singkatnya, adalah produk dari Google Research. Colab memungkinkan siapa saja menulis dan mengeksekusi kode Python melalui browser, serta sangat cocok untuk keperluan machine learning, analisis data, serta pendidikan (Google, 2020).

Secara lebih teknis, Colab merupakan layanan notebook Jupyter yang dihosting dan dapat digunakan tanpa perlu melakukan instalasi atau konfigurasi apa pun. Layanan ini menyediakan akses gratis ke resource komputasi, termasuk penggunaan GPU. Colab dapat digunakan tanpa biaya. Selain itu, notebook Colab dapat digunakan dan dibagikan dengan pengguna lain tanpa perlu mengunduh, menginstal, atau menjalankan sesuatu. Notebook Colab disimpan di Google Drive, dan juga dapat dimuat dari repositori GitHub. Dengan fitur ini, notebook Colab dapat dengan mudah dibagikan dan bekerja bersama dengan cara yang serupa dengan Google Dokumen atau Spreadsheet. Pengguna dapat mengelola dan menelusuri notebook Colab menggunakan Google Drive.

Notebook Colab dijalankan dengan menghubungkan ke mesin virtual yang memiliki batas waktu penggunaan maksimum selama 12 jam. Jika terjadi ketidakaktifan dalam jangka waktu tertentu, notebook Colab akan terputus dari Virtual Machine (VM). Batas waktu penggunaan maksimum dan waktu tunggu nonaktif dapat berubah dari waktu ke waktu, atau berdasarkan tingkat penggunaan. Tampilan awal notebook Colab dapat dilihat pada Gambar 2.27 berikut ini..



**Gambar 2.20** Tampilan *Notebook Google Colaboratory*

(Sumber : Nuril Hidayah , 2021)