

**BAB II**  
**LANDASAN TEORI**

**2.1 Tinjauan Pustaka**

**2.1.1 Analisis Penelitian Terdahulu**

Referensi penelitian terdahulu dengan menggunakan metode YOLO (You Only Look Once) untuk menghitung jumlah objek dan objek penelitiannya dirangkum dalam tabel berikut.

**Tabel 2. 1 Tinjauan Referensi Penelitian Terdahulu**

<b>NO</b>	<b>Penulis</b>	<b>Tahun</b>	<b>Metode</b>	<b>Judul</b>	<b>Hasil</b>
1	Agung Wibowo Ardiyantha Surbakti dan Rahmi Eka Putri ( Universitas Andalas)	2023	YOLOv 3 (You Only Look Once)	Penghitug Pengunjung dan Deteksi Masker Menggun kan OpenCV dan YOLO	Penelitian ini bertujuan untuk mengontrol jumlah penumpang dari kendaraan umum di pintu bus. Penelitian ini menggunakan metode YOLO

					<p>guna mendeteksi jumlah orang secara realtime. Data yang didapatkan di ambil dari pecahan video dan gambar. Selanjutnya, data akan dibagi menjadi tiga bagian yaitu data pelatihan, data pengujian, dan data validasi. Hasil penelitian menunjukkan tingkat akurasi sebesar 82%</p>
--	--	--	--	--	---

					<p>untuk pengujian pada video pintu masuk dan 72% untuk pengujian pada video pintu keluar, dengan menggunakan total 50 data video untuk pintu masuk dan pintu keluar.</p>
2	Adikrisna Pradiptya( Universitas Dinamika )	2023	YOLO (You Only Look Once)	Sistem Penghitung Jumlah Kerumunan Orang dengan Metode YOLO	<p>Penelitian ini merancang sistem yang dapat menghitung jumlah orang dalam kerumunan menggunakan</p>

					<p>metode YOLO.</p> <p>Dengan metode YOLO sistem dapat mendeteksi dan menghitung jumlah orang dengan akurat. Hasil dari penelitian ini ialah 80.96% tingkat akurasi dan tingkat error-nya 19.04%.</p>
3	Tsabita Al Asshifa Hadi Kusuma, Koredianto Usman, dkk(	2021	YOLO ( <i>You Only Look Once</i> )	People Counting Untuk Transportasi Publik Menggunakan	<p>Penelitian ini bertujuan untuk menghitung jumlah penumpang angkutan</p>

	Universitas Telkom)			<p>Metode You Only Look Once</p> <p>umum berbasis <i>computer</i> <i>vision</i> dengan menggunakan metode YOLO.. Arsitektur yang digunakan ialah YOLOv4. Selain itu, penelitian ini melibatkan tiga skenario pelatihan data dan dua skenario pengujian. Evaluasi dilakukan menggunakan beberapa metrik</p>
--	------------------------	--	--	--

					<p>termasuk akurasi, presisi, recall, f1-score, IoU, dan mAP.</p> <p>Hasil penelitian menunjukkan akurasi sebesar 69% untuk deteksi orang dengan mAP sebesar 72.68%, presisi sebesar 77%, dan rata-rata IoU sebesar 62.88%.</p>
4	Kelvin ( Universitas Hasanuddin )	2020	YOLO (You Only	Aplikasi Penghitun g Jumlah Perenang Pada	Penelitian ini membangun sebuah aplikasi yang bertujuan untuk

			<i>Look Once)</i>	Kolam Renang	mengatasi masalah bahaya kesehatan yang dapat timbul akibat tingginya kandungan urine dalam air kolam renang umum. Penelitian ini menggunakan 343 citra sebagai data yang terbagi menjadi 263 data latih dan 80 data uji. Algoritma YOLO generasi ke-3 digunakan
--	--	--	-----------------------	-----------------	--

					dalam penelitian ini untuk mendeteksi objek yang ada. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mencapai tingkat akurasi sebesar 90.8%.
5	Sulfan Bagus Setyawan, dkk (Politeknik Negeri Madiun)	2021	YOLO (You Only Look Once)	Sistem Deteksi Pengendara Sepeda Motor Tanpa Helm Dan Kelebihan Penumpaan	Penelitian ini bertujuan untuk mengurangi jumlah pelanggaran lalu lintas yang dilakukan oleh pengendara



				g Pada Dengan Mengguna kan Yolo V3	sepeda motor dengan tidak mematuhi peraturan penggunaan helm dan membawa penumpang melebihi kapasitas yang ditentukan. Sistem deteksi yang dibangun menggunakan algoritma YOLOv3 untuk mendeteksi pengendara sepeda motor tanpa helm dan penumpang yang melebihi
--	--	--	--	--	--

					<p>kapasitas.</p> <p>Hasil penelitian menunjukkan bahwa sistem ini mampu mendeteksi pelanggaran tersebut dengan akurasi sebesar 84.6%.</p> <p>Penelitian ini memiliki tujuan untuk membantu mengurangi pelanggaran lalu lintas yang sering dilakukan oleh pengendara sepeda motor, sehingga dapat</p>
--	--	--	--	--	---

					mencegah terjadinya korban jiwa dalam kecelakaan lalu lintas.
6	Ahlul halliwal aqdi (Universitas Lampung)	2023	YOLO (You Only Look Once)	Sistem Pendeteksi Dan Penghitungan Polen Hidup Dan Mati Pada Tanaman Kelapa Sawit Menggunakan Algoritma Yolo V5 Berbasis Artificial	Penelitian ini menunjukkan bahwa implementasi AI, khususnya menggunakan algoritma YOLO V5, dapat digunakan untuk mendukung pengembangan industri kelapa sawit dengan memastikan kualitas bahan

				Intelligenc e	<p>tanam melalui deteksi dan penghitungan viabilitas polen. Hasil yang diperoleh memiliki presisi tertinggi dibandingkan dengan model lainnya, yaitu sebesar 0,9204. Oleh karena itu, model tersebut digunakan untuk mendeteksi dan menghitung viabilitas polen. Hasil</p>
--	--	--	--	------------------	--

					<p>pengujian menunjukkan presisi sebesar 99,59%, recall sebesar 97,88%, dan akurasi sebesar 97,80%.</p>
7	<p>Bharathi S,dkk. (Dr. Mahalingam College of Engineering and Technology (MCET))</p>	2022	<p>YOLO (You Only Look Once) dan Shallow CNN (Convolutional Neural Network).</p>	<p>Expression Recognition using YOLO and Shallow CNN Model</p>	<p>The results of this research discuss the latest developments in Human Facial Expression Recognition techniques in the digital era, utilizing cascaded Convolutional Neural</p>

					<p>Network models for real-time facial expression recognition accuracy. The cascaded CNN approach, leveraging YOLO V5 for face detection and custom-trained CNN for expression recognition, achieves an impressive accuracy of 95.57%, surpassing existing HFER models.</p> <p>Overall, this</p>
--	--	--	--	--	--

					<p>research emphasizes the growing need for accurate facial expression recognition in the digital era and underscores the importance of ethical and privacy considerations in the deployment of this technology.</p>
--	--	--	--	--	--

### 2.1.2 Perbandingan Analisis Penelitian Terdahulu dengan 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 tercantum pada tabel di bawah ini.

**Tabel 2. 2 Tinjauan Penelitian Saat Ini.**

No	Penulis	Tahun	Metode	Judul	Hasil
1	Daniel Morantha	2023	YOLO( <i>You Only Look Once</i> )	Sistem Penghit ung Jumlah Ekspre si Orang Untuk Mengu kur Total Emosi Audien ce Di Vendin g Machin e Mengg unakan Metode	Pada penelitian ini dilakukan dengan 100 epoch dengan jumlah batch size sebanyak 10, image size 512 pixel menggunakan total dataset 1283 gambar yang terbagi menjadi 3 yaitu data latih sebanyak 716, data uji sebanyak 204 dan data validasi sebanyak 102. Penelitian ini memiliki <i>class</i>



				You	<i>object detection</i>
				Only	sebanyak 3 yaitu
				Look	<i>happy, sad</i> dan
				Once	<i>anger</i> . Performa
				(Yolo)	yang dihasilkan
				Pada	oleh arsitektur
				Studi	YOLOv5
				Peneliti	sebesar 85%
				an Pt	<i>accuracy</i> pada
				Parago	data uji dan
				n	sebesar 82%
				Pratam	<i>accuracy</i> pada
				a	data validasi.
				Teknol	
				ogi	

## 2.2 *Vending Machine*

*Vending machine* adalah sebuah mesin penjual barang secara otomatis yang penggunaannya saat ini dapat digunakan untuk menjual berbagai produk sekaligus untuk mengiklankan produk. Mesin ini biasanya ditempatkan di tempat-tempat umum seperti halte Transjakarta, stasiun kereta api, dan bandara agar dapat digunakan oleh banyak orang. Cara kerja mesin adalah dengan membayar tunai atau

menggunakan pembayaran elektronik, seperti kartu kredit atau aplikasi pembayaran mesin, lalu memilih produk, setelah itu produk yang dipilih segera diantarkan.

### **2.3 *Artificial Intelligence (AI)***

Menurut (F Chollet, 2019) Artificial Intelligence (AI) atau kecerdasan buatan adalah upaya untuk mengotomatisasi tugas-tugas intelektual yang biasanya dilakukan oleh manusia. Bidang ini lahir pada tahun 1950-an ketika para pionir dari ilmu komputer mulai mempertanyakan apakah komputer dapat "berpikir". Definisi yang singkat mengenai kecerdasan buatan adalah upaya untuk mengotomatisasi tugas-tugas intelektual manusia. Bidang ini mencakup pembelajaran mesin (machine learning) dan pembelajaran mendalam (deep learning), serta berbagai pendekatan lain yang tidak melibatkan pembelajaran. Pada awalnya, terdapat pendekatan simbolik AI yang menggunakan aturan baku yang telah ditentukan oleh para programmer. Namun, pendekatan ini sulit diterapkan pada masalah kompleks seperti pengenalan gambar, pengenalan ucapan, dan terjemahan bahasa. Oleh karena itu, muncul pendekatan baru yaitu pembelajaran mesin yang menggantikan kecerdasan buatan simbolik.

### **2.4 *Deep Learning***

Menurut (F Chollet, 2019) Deep learning (DL) adalah subbidang khusus dalam machine learning yang mengadopsi pendekatan baru dalam mempelajari representasi data. Fokus utama deep learning adalah pembelajaran secara bertahap melalui lapisan-lapisan representasi yang semakin kompleks dan bermakna. Kata "deep" dalam deep learning merujuk pada konsep lapisan-lapisan representasi yang

berturut-turut, bukan pada pemahaman yang lebih dalam yang dicapai melalui pendekatan ini.

Dalam deep learning modern, model-model yang terlibat seringkali terdiri dari puluhan bahkan ratusan lapisan representasi yang dipelajari secara otomatis melalui data pelatihan. Sebagai perbandingan, pendekatan lain dalam machine learning cenderung fokus pada pembelajaran hanya satu atau dua lapisan representasi, yang kadang-kadang disebut shallow learning.

Representasi-representasi berlapis ini dalam deep learning (biasanya) dipelajari melalui jaringan saraf (neural networks) yang terdiri dari lapisan-lapisan yang saling bertumpuk. Meskipun istilah "jaringan saraf" terinspirasi oleh neurobiologi, model-model deep learning tidak menggambarkan secara akurat bagaimana otak bekerja. Tidak ada bukti bahwa otak mengimplementasikan mekanisme pembelajaran yang serupa dengan mekanisme pembelajaran dalam model deep learning modern.

Penting untuk tidak membingungkan deep learning dengan neurobiologi. Deep learning adalah kerangka matematis untuk mempelajari representasi data, dan keterkaitan hipotetis antara deep learning dan biologi masih merupakan spekulasi. Fokus pada deep learning sebaiknya berdasarkan pada pemahaman dan aplikasi konsep matematis dalam mempelajari representasi data

## **2.5 *Computer Vison***

Menurut (M Elgendy, 2020) *Computer vision* adalah ilmu yang berkaitan dengan persepsi visual dalam sistem kecerdasan buatan (AI). Hal ini melibatkan

kemampuan sistem untuk mempersepsi dan memahami dunia melalui gambar dan video dengan membangun model fisik dunia. Dengan menggunakan teknik-teknik komputer, *computer vision* memungkinkan sistem AI untuk mengenali, menganalisis, dan menginterpretasikan informasi visual guna mengambil tindakan yang sesuai. Sebagaimana manusia memahami dunia melalui penglihatan, *computer vision* menjadi salah satu cara bagi sistem AI untuk memahami lingkungan di sekitarnya

## **2.6 *Object Detection***

Menurut (R Szeliski, 2022) Deteksi objek (*Object Detection*) adalah sebuah teknik dalam bidang pengolahan citra komputer yang bertujuan untuk mengidentifikasi dan memperoleh informasi tentang keberadaan serta lokasi objek-objek yang ada dalam suatu gambar atau video. Tujuan utama dari deteksi objek adalah untuk secara otomatis menemukan dan melabeli objek-objek tertentu yang menarik perhatian dalam data visual, sehingga memungkinkan komputer untuk memahami dan berinteraksi dengan dunia visual dengan cara yang serupa dengan manusia.

Dalam pengembangan deteksi objek, beberapa kompetisi dan dataset seperti PASCAL Visual Object Classes (VOC) Challenge dan ImageNet Large Scale Visual Recognition Challenge (ILSVRC) telah memberikan landasan penting dalam pengembangan algoritma deteksi objek. Awalnya, deteksi objek terfokus pada deteksi wajah dan pejalan kaki, namun bidang pengolahan citra komputer selalu tertarik untuk menyelesaikan masalah deteksi dan pelabelan objek secara umum, tidak hanya klasifikasi citra keseluruhan.

Beberapa pendekatan yang telah berhasil dalam deteksi objek melibatkan penggunaan metode seperti detektor berbasis fitur, klasifikasi SVM dengan metode spatial pyramid matching, serta model bagian deformable dengan topologi bintang. Kemajuan signifikan dalam deteksi objek juga terjadi dengan penggunaan deep learning, di mana jaringan saraf dalam (*deep neural networks*) digunakan untuk mempelajari representasi yang semakin kompleks dan berlapis dari data visual. Pendekatan ini telah membantu meningkatkan presisi dan keakuratan deteksi objek dalam berbagai tugas dan dataset, seperti yang ditunjukkan dalam perolehan nilai presisi rata-rata (AP) pada tugas deteksi objek dalam dataset Microsoft COCO.

Deteksi objek memiliki berbagai aplikasi yang luas, termasuk pengawasan keamanan, deteksi kendaraan otonom, analisis medis, dan pengenalan objek dalam industri teknologi. Dalam pengembangan deteksi objek, fokus utama adalah pada pengembangan algoritma dan teknik yang dapat secara efisien dan akurat mengenali objek dalam berbagai situasi dan kondisi, sehingga memberikan kontribusi yang signifikan dalam pengembangan kecerdasan buatan dan komputisasi visual.

## **2.7 Parameter *Evaluasi* dalam *Object Detection***

Dalam deteksi objek, *Average Precision* (AP) merupakan metrik yang sering digunakan untuk mengukur akurasi. Sebelum mempelajari berbagai variasi AP, penting untuk memahami beberapa konsep dasar yang didefinisikan di bawah ini.

1. True Positive (TP): Kebenaran deteksi dari ground-truth bounding box.

2. False Positive (FP): Kesalahan deteksi dari objek yang tidak ada atau kesalahan tempat deteksi dari objek yang ada.

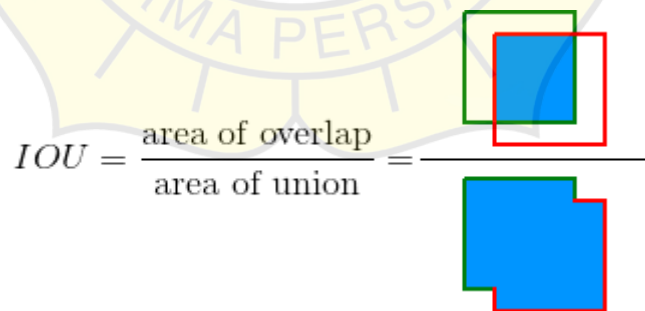
3. False Negative (FN): Ground-truth bounding box yang tidak terdeteksi.

Penting

untuk dicatat bahwa hasil True Negative (TN) tidak berlaku dalam konteks deteksi objek, karena ada banyak kotak pembatas yang tidak boleh dideteksi pada gambar tertentu.

Untuk deteksi objek, IOU mengukur area tumpang tindih antara kotak pembatas yang diprediksi  $B_p$  dan kotak pembatas ground-truth  $B_{gt}$  dibagi dengan luas unit di antara keduanya.

$$IOU = \frac{\text{area}(BB_{prediksi} \cap BB_{groundTruth})}{\text{area}(BB_{prediksi} \cup BB_{groundTruth})} \quad (2.1)$$



**Gambar 2. 1 Ilustrasi Intersection Over Union (IOU)**

Untuk mengklasifikasi deteksi sebagai benar atau salah sapat dengan membandingkan IOU dengan ambang batas tertentu  $t$ . Untuk  $IOU > t$ , deteksi

dianggap benar. Untuk  $IOU < t$  maka deteksi dianggap salah. Sebaliknya evaluasi metode deteksi objek paling sering didasarkan pada istilah precision P dan recall R. Precision didefinisikan sebagai berikut.

$$P = \frac{TP}{TP+FN} \quad (2.2)$$

Recall didefinisikan sebagai berikut.

$$R = \frac{TP}{TP+FN} \quad (2.3)$$

Presisi dihitung dengan membagi jumlah deteksi yang benar positif (true positive) dengan jumlah total deteksi positif (true positive dan false positive). Recall mengukur sejauh mana model mampu menemukan semua objek yang sebenarnya ada dalam dataset. Recall dihitung dengan membagi jumlah deteksi yang benar positif (true positive) dengan jumlah total objek yang sebenarnya ada (true positive dan false negative). Mean Average Precision (mAP) adalah ukuran yang menggabungkan presisi dan recall pada berbagai tingkat ambang batas (threshold) untuk deteksi objek. mAP dihitung dengan menghitung rata-rata presisi di setiap ambang batas dan kemudian mengambil rata-rata dari nilai-nilai tersebut.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.4)$$

Di mana  $AP_i$  adalah AP untuk kelas ke – i dan N jumlah total kelas yang dievaluasi. Konsep umum untuk menghitung nilai presisi maupun recall biasanya diterapkan dalam bentuk confusion matrix. Confusion matrix digambarkan sebagai berikut.

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

Gambar 2. 2 Confusion Matrix

*Confusion Matrix* adalah sebuah tabel yang dipergunakan dalam pembelajaran mesin untuk mengevaluasi kinerja klasifikasi, dengan memberikan gambaran tentang jumlah prediksi yang benar dan yang salah sebagaimana telah dijelaskan sebelumnya. Dari sini dapat pula dihitung akurasi dari hasil prediksi sebuah mode dengan formula berikut.

$$\text{Akurasi} = \frac{TP}{TP+FP+FN} \quad (2.5)$$

## 2.8 Pengolahan Citra

Menurut (M Elgendy, 2020) pengolahan citra adalah tahap pemrosesan data gambar sebelum dimasukkan ke dalam model ML. Tujuannya adalah untuk menyederhanakan dan meningkatkan akurasi algoritma yang diterapkan pada gambar tersebut. Pengolahan citra melibatkan berbagai tugas seperti perubahan ukuran, transformasi geometri dan warna, konversi warna ke skala abu-abu, serta langkah-langkah lainnya untuk mengubah gambar menjadi bentuk yang dapat



diproses oleh algoritma secara umum. Melalui teknik-teknik pengolahan citra ini, data gambar yang diperoleh dari berbagai sumber dapat dinormalisasi dan disesuaikan sehingga siap digunakan dalam model ML atau jaringan saraf. Pengolahan citra memiliki tujuan untuk meningkatkan kualitas gambar agar dapat dengan mudah dipahami oleh manusia atau mesin, terutama komputer. Dalam pengolahan citra, berbagai teknik digunakan untuk mentransformasikan citra menjadi citra yang memiliki kualitas yang lebih baik daripada citra aslinya. Dengan demikian, input dari pengolahan citra adalah citra, dan outputnya juga merupakan citra yang telah mengalami perbaikan. Salah satu aspek yang termasuk dalam bidang ini adalah pemampatan citra (image compression).

## **2.9 Akuisisi Citra**

Menurut (Gansar Suwanto, et al., 2021) menyatakan pada tahap akuisisi data, dilakukan pengambilan citra. Tujuan dari tahap ini adalah untuk menentukan jenis data yang diperlukan dan memilih metode perekaman citra digital. Proses ini melibatkan beberapa langkah, dimulai dari persiapan objek yang akan diambil citranya, persiapan alat yang digunakan, hingga proses pengambilan citra itu sendiri (Kusumadewa & Supatman, 2018).

## **2.10 Algoritma *You Only Look Once* (YOLO)**

Menurut (M Elgendy, 2020) YOLO adalah serangkaian model DL end-to-end yang dirancang untuk deteksi objek cepat, dan merupakan salah satu upaya awal dalam membangun detektor objek real-time yang cepat. Ini merupakan salah satu algoritma deteksi objek yang lebih cepat yang ada. Meskipun akurasi modelnya

hampir sebanding, tetapi tidak sebaik R-CNN, mereka populer untuk deteksi objek karena kecepatan deteksinya, sering ditunjukkan dalam video real-time atau input feed kamera. YOLO merupakan sebuah algoritma yang menggunakan jaringan saraf satu tingkat untuk mendeteksi dan mengklasifikasikan objek secara real-time. YOLO menganggap pengenalan gambar sebagai masalah regresi yang cepat dan sederhana. Ini memungkinkan YOLO untuk memproses video streaming secara real-time dengan waktu tunda kurang dari 25 detik. Selama proses pelatihan, YOLO menganalisis keseluruhan gambar dan menggambarkan informasi global dalam pengenalan objek. Konsep inti YOLO adalah menggunakan gambar secara keseluruhan sebagai input ke jaringan dan langsung menghasilkan prediksi lokal dan kelas kotak pembatas sebagai output. Dalam YOLO, setiap kotak pembatas diprediksi berdasarkan fitur keseluruhan gambar, dan setiap kotak pembatas mengandung lima prediksi dan tingkat kepercayaan yang berkaitan dengan sel kisi di dalamnya. Kerangka dasar YOLO adalah sebagai berikut : w dan h masing

- masing memprediksi lebar dan tinggi seluruh gambar (relatif terhadap seluruh gambar).

YOLO terdiri dari tiga bagian utama, yaitu :

1. Backbone : Jaringan saraf convolutional yang mencampurkan serta membentuk fitur foto pada bermacam tipe perinci foto.

2. Neck : Serangkaian susunan jaringan yang menggabungkan serta mencampurkan fitur foto serta melewati fitur foto ke susunan prediksi.
3. Head : Komponen ini bisa memprediksi fitur foto, menciptakan kotak pembatas, serta memprediksi jenis. Kepercayaan mewakili keakuratan klasifikasi serta keadaan tertentu.

### **2.10.1 Deskripsi YOLOv5**

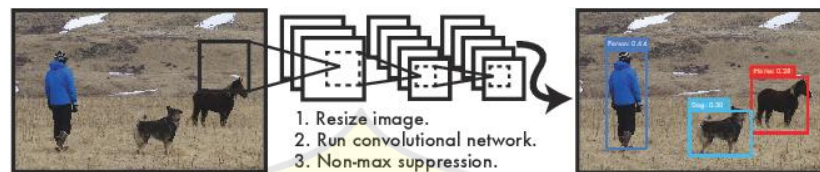
Menurut (Kresna, 2023) YOLOv5 merupakan algoritma terbaru yang merupakan versi lanjutan dari algoritma "You Only Look Once" (YOLO). Algoritma ini memanfaatkan deep learning untuk melakukan deteksi objek secara real-time. YOLOv5 mengadopsi pendekatan jaringan syaraf tiruan (JST) untuk mendeteksi objek dalam citra dengan presisi tinggi.

Pada YOLOv5, sebuah model diterapkan pada citra dalam berbagai lokasi dan skala. Citra dibagi menjadi beberapa wilayah, dan setiap wilayah diprediksi untuk menghasilkan kotak pembatas objek dan probabilitas keberadaannya. Setiap kotak pembatas kemudian dibandingkan dengan probabilitas yang diprediksi untuk menentukan deteksi objek. Wilayah dengan skor tertinggi akan dianggap sebagai hasil deteksi.

Konsep YOLOv5 sangat sederhana. Pertama, gambar input diubah ukurannya menjadi 448 x 448 piksel sesuai dengan kebutuhan implementasi. Kemudian, sistem menjalankan jaringan konvolusi tunggal pada gambar tersebut.

Pada tahap terakhir, sistem memberikan hasil deteksi objek dengan tingkat kepercayaan (confidence) sesuai dengan model yang telah dilatih sebelumnya.

YOLOv5 memiliki kecepatan tinggi dan kinerja yang akurat dalam mendeteksi objek pada citra secara real-time. Algoritma ini telah memberikan kontribusi signifikan dalam pengembangan sistem deteksi objek yang efisien dan efektif.



**Gambar 2. 3 Sistem deteksi pada YOLOv5**

### 2.10.2 Arsitektur YOLOv5

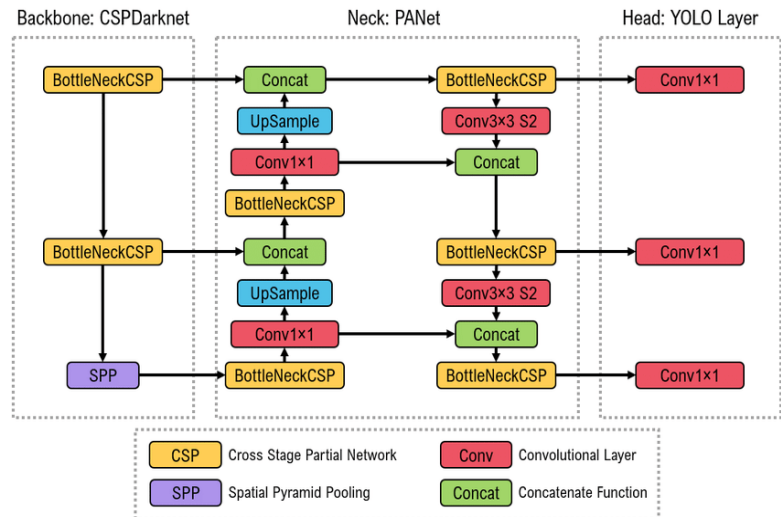
Menurut (A Levina, 2021) YOLOv5 mengalami beberapa perubahan dibandingkan dengan versi sebelumnya. Salah satu perubahan tersebut adalah penggunaan PyTorch sebagai pengganti Darknet. Sebagai bagian inti, YOLOv5 menggunakan CSP Darknet 53 sebagai pendukung. Pendukung ini mengatasi masalah pengulangan informasi gradien pada pendukung yang besar dan mengintegrasikan perubahan gradien ke dalam peta fitur. Hal ini menghasilkan peningkatan kecepatan inferensi, akurasi, dan pengurangan ukuran model dengan mengurangi jumlah parameter.

YOLOv5 juga memanfaatkan Path Aggregation Network (PANet) sebagai penghubung untuk meningkatkan aliran informasi. PANet mengadopsi Feature Pyramid Network (FPN) yang baru dan terdiri dari lapisan bottom up dan top down. Pendekatan ini memperbaiki propagasi fitur pada tingkat rendah dalam model,

meningkatkan kemampuan lokalisasi objek pada lapisan bawah, dan meningkatkan akurasi deteksi secara keseluruhan.

Head pada YOLOv5 tetap sama dengan versi YOLO v4 dan YOLO v3. Hal ini menghasilkan tiga peta fitur keluaran yang berbeda untuk memprediksi objek pada skala multi. Pendekatan ini membantu meningkatkan efisiensi dalam memprediksi objek dengan berbagai ukuran dalam model. Gambar masukan diproses oleh CSP Darknet 53 untuk ekstraksi fitur, kemudian diteruskan ke PANet untuk penggabungan fitur. Hasil akhir dikeluarkan oleh lapisan YOLO.

Pendekatan ini juga membantu meningkatkan efisiensi dalam memprediksi objek dari ukuran kecil hingga besar dalam model. Gambar diolah melalui CSP Darknet 53 untuk ekstraksi fitur dan kemudian melalui PANet untuk menggabungkan fitur. Akhirnya, lapisan YOLO menghasilkan hasil deteksi. Gambar arsitektur YOLOv5 dapat dilihat sebagai berikut.

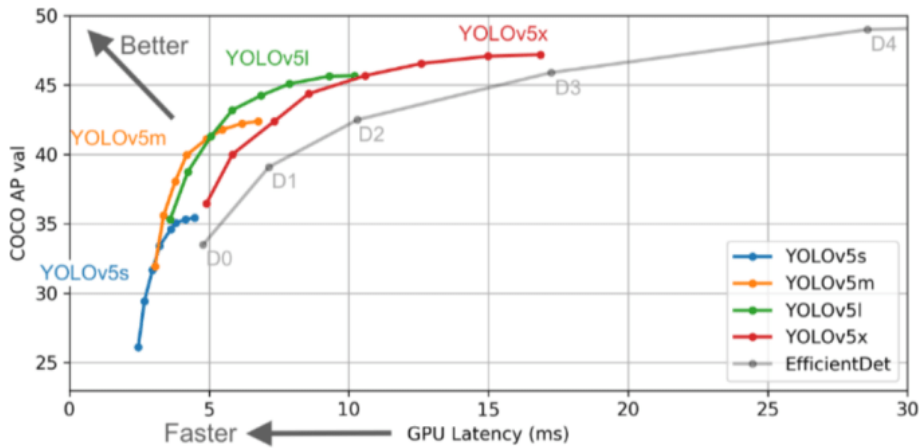


**Gambar 2. 4 Arsitektur YOLOv5**

Lapisan konvolusi ditunjukkan oleh Conv. C3 terdiri dari tiga lapisan konvolusi dan modul yang melewati berbagai hambatan. SPP (Spatial Pyramid Pooling) adalah lapisan pengelompokan spasial yang digunakan untuk mengatasi pembatasan ukuran yang tetap pada jaringan. Upsample digunakan untuk meningkatkan resolusi fusi lapisan sebelumnya pada node terdekat. Concat adalah lapisan penggabungan yang digunakan untuk menggabungkan layer sebelumnya. Tiga Conv2d terakhir adalah modul deteksi objek yang digunakan di bagian kepala jaringan.

### 2.10.3 Tipe YOLOv5

Menurut (A Levina, 2021) YOLO v5 memiliki beberapa tipe dengan masing – masing memiliki perbedaan dari segi kecepatan deteksi dan performa mAPnya. Berikut gambar dari perbedaan dari masing – masing tipe YOLO v5.



**Gambar 2.5** Performa Tipe Model YOLOv5

**Tabel 2.3** Performa Tipe Model YOLOv5

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

Pada **Gambar 2.4** terlihat bahwa semakin tinggi pada sumbu model, maka mAP model akan semakin baik, namun semakin ke kiri, deteksi akan semakin cepat.

Pada **Tabel 2.4**, terdapat beberapa indikator yang dapat dipertimbangkan dalam pemilihan model. Model YOLO v5 dilatih dengan menggunakan data COCO dengan ukuran 640 piksel. Dalam hal kecepatan (Speed), semakin besar nilai

params, maka kecepatan akan semakin tinggi. Params merujuk pada parameter pelatihan. Oleh karena itu, semakin besar nilai params, waktu komputasi akan lebih lama dan deteksi akan menjadi lebih lambat.

## **2.11 Cross-Industry Standard Process for Data Mining (CRISP-DM)**

Menurut (Nurchalifatun, 2017) CRISP-DM adalah singkatan dari *Cross Industry Standard Process Model for Data Mining* dan merupakan pendekatan strategis untuk memecahkan masalah umum dalam bisnis atau penelitian. Metode ini terdiri dari 6 langkah yaitu *Business Understanding, Data Understanding, Data preparation, Modeling, Evaluation, dan Deployment*. Alur proses metode CRISP-DM melewati 6 langkah sebagai berikut:

### **1. Business Understanding**

Tahap *Business Understanding* adalah tahap dalam proses data mining di mana penting untuk memahami masalah yang akan diselesaikan. Pada tahap ini, peneliti merumuskan dan mengidentifikasi masalah bisnis yaitu sistem vending machine masih terbatas. Dalam memahami secara akurat dan realtime respon emosional dari para pengunjung yang berinteraksi maupun yang berada di sekitar vending machine. Setelah memahami masalah, peneliti menentukan solusi bisnis dari jenis datamining yang ada berdasarkan masalah yang akan diselesaikan. Jenis datamining yang cocok untuk mengenali emosional dari para pengunjung adalah classification.

### **2. Data Understanding**



*Data Understanding* adalah tahap dalam proses data mining di mana penting untuk memahami data yang tersedia untuk digunakan dalam membangun solusi untuk memecahkan masalah bisnis. Pada tahap ini, peneliti menelaah data yang telah dikumpulkan untuk melihat kecocokan data terhadap masalah yang akan diselesaikan. Data yang dikumpulkan berupa data gambar dengan 3 kelas ekspresi yaitu sedih, marah dan bahagia dengan total keseluruhan sebesar 1005 gambar.

### **3. *Data Preparation***

*Data Preparation* adalah tahap dalam proses data mining yang melibatkan manipulasi dan konversi data agar sesuai dengan persyaratan teknologi analitik yang digunakan. Pada tahap ini, peneliti menentukan label data dan menghilangkan data yang memiliki resolusi buruk dari data yang telah dikumpulkan agar akurasi model yang dihasilkan lebih optimal.

### **4. *Modeling***

*Modelling* adalah tempat utama di mana teknik data mining diterapkan pada data. Pada tahap ini, peneliti membangun model menggunakan arsitektur YOLOv5 untuk mendeteksi ekspresi sedih, marah dan bahagia.

### **5. *Evaluation***

*Evaluation* adalah tahap yang bertujuan untuk secara ketat menilai hasil data mining dan memperoleh keyakinan bahwa hasil tersebut valid dan dapat diandalkan sebelum melanjutkan. Pada tahap ini, peneliti mengevaluasi hasil pemodelan dengan arsitektur YOLOv5 berdasarkan penilaian kualitatif dan kuantitatif.

## **6. Deployment**

*Deployment* adalah tahap terakhir dalam siklus CRISP-DM, di mana hasil dari proses pemodelan dan evaluasi digunakan untuk menerapkan solusi atau model yang telah dikembangkan ke dalam lingkungan produksi. Pada tahap ini, peneliti membuat rencana deployment model, melakukan deployment model dalam bentuk web menggunakan HTML, CSS, JS, dan Python dengan framework FLASK.

### **2.12 Google Drive**

Menurut (Ainun, 2019) *Google drive* adalah sebuah layanan yang dibuat oleh Google sebagai platform penyimpanan data online berbasis cloud atau internet. Layanan ini pertama kali diperkenalkan oleh Google pada tanggal 24 April 2012. *Google Drive* dapat diakses melalui perangkat komputer maupun ponsel pintar. Selain itu, *Google Drive* juga telah menjadi aplikasi bawaan di beberapa jenis ponsel. Dengan menggunakan *Google Drive*, pengguna dapat menyimpan berbagai jenis file seperti dokumen, gambar, video, dan audio. Kapasitas penyimpanan awal yang disediakan oleh *Google Drive* adalah 15 *Gigabyte*, namun pengguna memiliki opsi untuk melakukan upgrade akun agar mendapatkan kapasitas penyimpanan yang lebih besar. Salah satu kelebihan *Google Drive* adalah kemampuannya untuk diakses secara fleksibel dan praktis, kapan pun dan di mana pun selama perangkat yang digunakan terhubung dengan jaringan internet.



**Gambar 2. 6 Google Drive**

### **2.13 *Google Collaboratory***

Menurut (R Imanuel, 2020) Google Collaboratory adalah sebuah alat atau platform berbasis cloud yang disediakan oleh Google untuk keperluan penelitian. Google Colab memungkinkan pengguna untuk menulis dan menjalankan kode secara online, mirip dengan cara berkolaborasi dalam Google Doc. Google Colab, juga dikenal sebagai Google Collaboratory, merupakan alat yang gratis dan dapat diakses melalui internet. Lingkungan Google Colab didasarkan pada Jupyter Notebook, yang memungkinkan pengguna untuk mengedit dan menjalankan kode Python secara interaktif.

Salah satu keunggulan Google Colab adalah kemampuannya untuk mendukung hampir semua pustaka atau library yang diperlukan dalam pengembangan kecerdasan buatan (Artificial Intelligence) dan pembelajaran mesin (Machine Learning). Pengguna dapat dengan mudah mengimpor dan menggunakan pustaka seperti TensorFlow, PyTorch, dan scikit-learn dalam proyek-proyek mereka.



**Gambar 2. 7 Google Collaboratory**

Secara dasar, Google Colab mirip dengan Jupyter Notebook, namun dengan perbedaan bahwa Google Colab dapat diakses dan dijalankan secara online secara gratis. Pengguna tidak perlu menginstal atau mengkonfigurasi lingkungan pengembangan secara lokal di perangkat mereka, karena semua pemrosesan dan penyimpanan kode dilakukan di cloud.

Google Colab juga menyediakan berbagai fitur yang berguna, seperti dukungan untuk visualisasi data, integrasi dengan Google Drive untuk penyimpanan dan berbagi proyek, serta akses ke mesin pemrosesan grafis (GPU) yang dapat mempercepat pemrosesan pada tugas-tugas yang intensif secara komputasional.

Secara keseluruhan, Google Colab memberikan akses yang mudah, fleksibel, dan gratis untuk mengembangkan proyek-proyek AI dan Machine Learning dalam lingkungan cloud.

#### **2.14 Dataset**

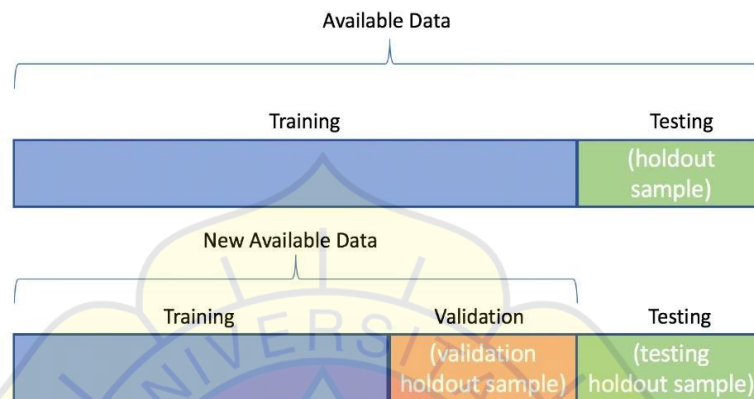
Menurut (A Géron, 2019) Dataset adalah kumpulan data yang digunakan untuk melatih, menguji, atau menguji kinerja model atau algoritma dalam bidang pembelajaran mesin dan kecerdasan buatan. Dataset dapat berupa sekumpulan gambar, teks, video, audio, atau bentuk data lainnya yang direpresentasikan dalam bentuk yang dapat diproses oleh komputer. Dataset berfungsi sebagai input bagi

model atau algoritma untuk mempelajari pola, melakukan prediksi, atau melakukan tugas tertentu sesuai dengan tujuan yang ditentukan.

Dataset yang baik harus mewakili populasi atau domain yang relevan dengan masalah yang ingin diselesaikan atau dipecahkan. Hal ini memastikan bahwa model yang dilatih dengan dataset tersebut dapat menggeneralisasi dan berkinerja baik pada data baru yang belum pernah dilihat sebelumnya. Selain itu, dataset juga harus terdiversifikasi, yaitu mencakup variasi yang cukup dari contoh-contoh yang berbeda untuk memastikan model tidak terlalu khusus atau terlalu umum dalam pemahaman pola. Dataset dapat dibagi menjadi 3, yaitu *train set*, *validation set*, dan *test set*.

1. *Train set* : Train set adalah dataset aktual yang digunakan untuk melatih model, yaitu menyesuaikan bobot dan bias dalam jaringan saraf. Model menggunakan train set ini untuk mempelajari pola dan memperbarui parameter internalnya. sampel data yang digunakan untuk menyesuaikan dengan model.
2. *Validation set* : Sampel data yang digunakan untuk mengukur kinerja model selama proses pelatihan. Validation set tidak digunakan dalam proses pelatihan model, namun data ini digunakan selama pelatihan untuk mengembangkan model. Validation set membantu dalam mengevaluasi kinerja model dan memilih hyperparameter yang optimal, seperti ukuran batch atau jumlah epoch, dengan melihat sejauh mana model mampu menggeneralisasi dari data pelatihan ke data yang belum pernah dilihat sebelumnya.
3. *Test set* : Sampel data yang digunakan untuk menguji kinerja akhir model setelah proses pelatihan dan validasi selesai. Test set digunakan untuk menguji

sejauh mana model yang telah dilatih mampu melakukan prediksi yang akurat dan umumnya terdiri dari data yang belum pernah dilihat oleh model selama proses pelatihan. Dengan menggunakan test set, kita dapat mengukur performa aktual model dan mengevaluasi tingkat keakuratannya dalam melakukan prediksi pada data baru yang belum pernah dilihat sebelumnya.



**Gambar 2. 8 Pembagian Dataset**

### 2.14.1 Roboflow

Roboflow adalah platform yang dirancang untuk memudahkan pengembang dan ilmuwan data dalam mengelola, memproses, dan melabeli data gambar untuk pelatihan model pembelajaran mesin. Dengan menggunakan Roboflow, pengguna dapat mengimpor dataset gambar mereka, membersihkannya, melakukan augmentasi data, melabeli gambar, dan mengekstrak fitur dari dataset tersebut.

### 2.14.2 Augmentasi Data

Menurut (A Levina, 2021) Augmentasi data dapat dilakukan dalam dua cara, yaitu secara online dan offline. Augmentasi data online dilakukan selama pelatihan jaringan, di mana gambar-gambar yang telah diaugmentasi akan langsung

digunakan setelah melewati lapisan pertama jaringan. Dalam augmentasi online, tidak diperlukan langkah pra-pemrosesan data tambahan. Keuntungan dari augmentasi online adalah penghematan waktu karena tidak perlu tahap pra-pemrosesan, dan model dapat secara acak menampilkan gambar yang dihasilkan hanya sekali dan menyimpannya sementara di cache GPU, sehingga mengurangi penggunaan ruang penyimpanan.

Augmentasi data *offline* dilakukan di luar sistem dengan menggunakan algoritma pemrosesan data yang serupa dengan langkah pra-pemrosesan. Metode ini memerlukan waktu dan ruang penyimpanan tambahan karena gambar-gambar harus dibagi menjadi beberapa kelompok dan diproses terlebih dahulu sebelum pelatihan. Namun, metode augmentasi offline juga memiliki keuntungan. Metode ini dapat meningkatkan reproduisibilitas model karena menyimpan salinan setiap gambar yang telah ditambahkan. Misalnya, jika model memiliki performa yang lebih baik pada gambar terang daripada gambar gelap, maka perlu adanya pengumpulan lebih banyak data pelatihan dengan kondisi pencahayaan rendah. Selain itu, augmentasi offline juga dapat mempercepat waktu pelatihan model karena tidak perlu melakukan proses augmentasi saat pelatihan, sehingga dapat menghemat biaya penggunaan GPU.

Dengan demikian, baik augmentasi data secara online maupun offline memiliki kelebihan dan keuntungan masing-masing. Pilihan antara kedua metode ini bergantung pada kebutuhan spesifik dari pelatihan model, ketersediaan sumber daya, dan tujuan yang ingin dicapai.

Beberapa macam augmentasi data yang bisa digunakan untuk ekspansi data adalah sebagai berikut.

1. Flip: Membalik foto secara acak dengan arah vertikal ataupun horizontal.
2. Rotate: Memutar foto secara acak searah jarum jam ataupun bertentangan arah jarum jam sampai jumlah derajat yang dikonfigurasi pas.
3. Crop: Memotong gambar pada bagian tertentu.
4. Shear: Mendistorsi foto melintasi sumbu horizontal ataupun vertikal.
5. Exposure: Menyesuaikan eksposur gamma gambar menjadi lebih terang atau lebih gelap.
6. Blur: Memakai Gaussian Blur pada foto.
7. Noise: Membagikan derau secara acak pada gambar.

### **2.14.3 Underfitting dan Overfitting**

Menurut (A Levina, 2021) Ketidakcocokan dan kelebihan penyesuaian (overfitting) adalah dua masalah utama yang terjadi selama tahap pelatihan. Ketidakcocokan mengindikasikan bahwa jaringan saraf yang dilatih tidak cukup dan memiliki akurasi pembelajaran yang rendah. Di sisi lain, kelebihan penyesuaian berarti bahwa model yang dilatih bekerja dengan baik hanya pada data pelatihan dan tidak dapat beradaptasi dengan baik pada data pengujian. Hal ini mengurangi fleksibilitas dan kemampuan adaptasi model. Ketidakcocokan dan kelebihan penyesuaian merupakan fenomena umum dalam pembelajaran mesin. Hingga saat ini, pandangan umum adalah bahwa ketidakcocokan dan kelebihan penyesuaian terkait dengan kompleksitas arsitektur jaringan saraf, jumlah dan jenis sampel data yang digunakan dalam set pelatihan, serta jumlah iterasi pelatihan yang dilakukan.



#### **2.14.4 *Epoch* dan *Batch Size***

Menurut (Ian Goodfellow et al, 2016) *Epoch* adalah satu iterasi lengkap dari seluruh dataset pelatihan yang digunakan dalam proses pembelajaran mesin. Pada setiap epoch, model melihat dan memproses seluruh dataset pelatihan untuk memperbarui parameter internalnya. *Epoch* menentukan seberapa sering model melihat kembali data pelatihan dan melakukan pembaharuan.

*Batch Size* adalah jumlah sampel data yang diproses dalam satu iterasi dalam algoritma pembelajaran mesin. Data pelatihan sering kali terdiri dari banyak sampel, dan dengan menggunakan batch size, kita dapat membagi data menjadi batch-batch kecil untuk mempercepat proses pembelajaran dan mengelola sumber daya komputasi.

#### **2.15 *WebView***

Menurut (Voutama & Novalia, 2021), *WebView* adalah salah satu fasilitas aplikasi Android Studio. *WebView* merupakan sebuah class di Android yang dapat menampilkan halaman web pada aplikasi Android. Dimana skenario umum yang menggunakan *WebView* bermanfaat adalah ketika kita ingin memberikan informasi dalam aplikasi yang mungkin perlu diperbarui, seperti perjanjian pengguna akhir atau panduan pengguna. Dengan menggunakan Android, dapat dibuat Activity yang berisi *WebView*, lalu menggunakannya untuk menampilkan dokumen yang di hosting secara online.

## 2.16 Flask

Menurut (A Chandra Darmawan, 2023) Flask adalah sebuah framework Python yang dikenal sebagai "micro-framework" karena ukurannya yang kecil dibandingkan dengan kebanyakan framework lainnya. Meskipun tergolong kecil, Flask tetap merupakan sebuah framework yang kuat dan dapat diandalkan. Flask dirancang sebagai micro-framework yang dapat diperluas sesuai kebutuhan, dengan inti yang solid dan kemampuan untuk menambahkan ekstensi sesuai kebutuhan proyek. Salah satu kelebihan Flask adalah fleksibilitas dalam memilih dan menggunakan paket ekstensi yang sesuai dengan kebutuhan proyek, sehingga tidak ada kelebihan ukuran file yang tidak perlu.

Flask memiliki beberapa dependensi utama yang mendukung fungsionalitasnya, seperti sistem routing, debugging, dan Web Server Gateway Interface (WSGI). Subsistem Flask berasal dari Werkzeug, sedangkan dukungan untuk template menggunakan Jinja2, dan integrasi dengan baris perintah menggunakan Click. Semua dependensi ini dikembangkan oleh Armin Ronacher, pencipta Flask.

Flask tidak memiliki dependensi bawaan untuk sistem database, validasi web-forms, otentikasi pengguna, dan tugas tingkat tinggi lainnya. Hal ini memungkinkan pengembang untuk memilih ekstensi atau pustaka yang sesuai dengan kebutuhan proyek. Fleksibilitas ini menjadi keuntungan Flask dibandingkan dengan framework lain yang memiliki opsi yang lebih terbatas dan sulit untuk diubah setelah dipilih.

Dengan Flask, pengembang memiliki kontrol penuh atas ekstensi yang digunakan dalam proyek, sehingga dapat memilih dan menyesuaikan sesuai kebutuhan. Ini membuat Flask menjadi pilihan yang fleksibel dan dapat disesuaikan untuk berbagai jenis proyek, baik yang sederhana maupun kompleks.

### **2.17 Fitur-Fitur Flask**

Menurut (Rahadian Irsyad, 2018) Seperti yang telah dijelaskan sebelumnya, Flask adalah sebuah microframework yang memiliki inti yang sederhana dan kecil. Meskipun demikian, Flask memiliki kemampuan untuk ditambah dan diperluas sesuai kebutuhan. Berdasarkan hal tersebut, fitur – fitur bawaan dari flask sendiri termasuk sedikit jumlahnya, diantaranya adalah :

1. Built-in development server.
2. Debugger cepat.
3. Integrated support untuk pengetesan unit.
4. Kompatibel dengan mesin aplikasi Google.
5. RESTful request dispatching.
6. Jinja2 templating.
7. Mendukung secure cookies.
8. Berbasis unicode.
9. MengikutiWSGI1.0



**TEKNOLOGI INFORMASI**

**UNIVERSITAS DARMA PERSADA**