



UNIVERSITAS
Dinamika

**SISTEM PEMANTAUAN PEMBUANGAN SAMPAH SEMBARANGAN
OLEH MANUSIA BERBASIS *ARTIFICIAL INTELLIGENCE*
MENGUNAKAN METODE YOLOv8**

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

UNIVERSITAS
Dinamika

Oleh:

Andika Arga Pradana

21410200022

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2025

**SISTEM PEMANTAUAN PEMBUANGAN SAMPAH SEMBARANGAN
OLEH MANUSIA BERBASIS *ARTIFICIAL INTELLIGENCE*
MENGUNAKAN METODE YOLOv8**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Terapan Seni**



UNIVERSITAS
Dinamika

Oleh:

Nama : Andika Arga Pradana
NIM : 21410200022
Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2025

TUGAS AKHIR

SISTEM PEMANTAUAN PEMBUANGAN SAMPAH SEMBARANGAN OLEH MANUSIA BERBASIS *ARTIFICIAL INTELLIGENCE* MENGUNAKAN METODE YOLOv8

Dipersiapkan dan disusun oleh

Andika Arga Pradana

NIM: 21410200022

Telah diperiksa, dibahas dan disetujui oleh dewan pembahas

Pada: Jumat, 25 Juli 2025

Susunan Dewan Pembahas

Pembimbing:

I. Heri Pratikno, M.T.

NIDN: 0716117302

II. Weny Indah Kusumawati, S.Kom., M.MT.

NIDN: 0721047201

Pembahas:

III. Pauladie Susanto, S.Kom., M.T.

NIDN: 0729047501



Digitally signed by Heri Pratikno, M.T.
DN: cn=Heri Pratikno, M.T.,
o=Universitas Dinamika, ou=S1 Teknik
Komputer, email=heri@dinamika.ac.id,
c=ID
Date: 2025.08.01 22:32:37 +07'00'
Adobe Acrobat version: 11.0.23



cn=Weny Indah Kusumawati,
o=Undika, ou=Prodi S1 TK - FTI,
email=weny@dinamika.ac.id, c=ID
2025.08.01 12:22:44 +07'00'



cn=Pauladie Susanto, o=Universitas
Dinamika, ou=PS S1 Teknik Komputer,
email=pauladie@dinamika.ac.id, c=ID
2025.08.04 10:15:09 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar Sarjana



Digitally signed by

Julianto

Date: 2025.08.06 14:23:59

+07'00'

Julianto Lemantara, S.Kom., M.Eng.

NIDN. 0722108601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



“Ilmu yang tak disertai keraguan hanya akan menjadi keyakinan buta. Keraguan tanpa ilmu tak lebih dari kebingungan. Namun, ketika ilmu dan keraguan bersatu, lahirlah cahaya yang membimbing umat manusia”

~Andika Arga Pradana~.



Dipersembahkan kepada Ayah dan Ibu tercinta yang doa, cinta, dan dukungannya tak pernah putus dalam setiap langkah hidupku. Laporan Tugas Akhir ini adalah bukti bahwa meskipun pernah gagal, dengan kasih dan keyakinan dari kalian, aku mampu bangkit dan melangkah menuju keberhasilan.

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **Andika Arga Pradana**
NIM : **21410200022**
Program Studi : **S1 Teknik Komputer**
Fakultas : **Fakultas Teknologi dan Informatika**
Jenis Karya : **Laporan Tugas Akhir**
Judul Karya : **SISTEM PEMANTAUAN PEMBUANGAN SAMPAH
SEMBARANGAN OLEH MANUSIA BERBASIS
ARTIFICIAL INTELLIGENCE MENGGUNAKAN METODE YOLOv8**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 10 Juli 2025



Andika Arga Pradana
NIM : 21410200022

ABSTRAK

Permasalahan pembuangan sampah sembarangan merupakan tantangan utama dalam menjaga kebersihan lingkungan, terutama di area publik seperti taman dan jalanan. Upaya pengawasan petugas keamanan terbatas karena tidak bisa mengawasi selama 24 jam, sedangkan CCTV perlu memeriksa hasil rekaman secara manual. Oleh karena itu, diperlukan sistem cerdas berbasis teknologi untuk mendeteksi dan mendokumentasikan aktivitas tersebut secara otomatis. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pemantauan pembuangan sampah sembarangan berbasis *Artificial Intelligence* menggunakan metode YOLOv8 dan *Object Tracking ByteTrack*. Sistem ini memanfaatkan *IP Camera* untuk merekam aktivitas secara *real-time*, mendeteksi manusia dan objek yang dibawa, serta mengidentifikasi tindakan pembuangan sampah sembarangan melalui penghitungan jarak berbasis perbandingan piksel menggunakan *Euclidean Distance*. Pengujian menunjukkan akurasi deteksi manusia dan objek yang dibawa sebesar 100% pada siang hari dan 80% pada malam hari. Akurasi deteksi pembuangan sampah sembarangan sebesar 73,33% pada siang hari dan 66,7% pada malam hari, dengan rata-rata waktu pengiriman data sebesar 6,5 detik. Sistem ini diharapkan dapat membantu pihak terkait dalam memantau dan mendokumentasikan pelanggaran kebersihan lingkungan.

Kata Kunci : YOLOv8, IP Camera, Artificial Intelligence, Object Tracking ByteTrack, Euclidean Distance

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan anugerah-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul "Sistem Pemantauan Pembuangan Sampah Sembarangan Oleh Manusia Berbasis *Artificial Intelligence* Menggunakan Metode YOLOv8".

Dalam proses penyusunan laporan ini, penulis ingin menyampaikan rasa terima kasih yang mendalam kepada semua pihak yang telah memberikan bantuan, arahan, dan dukungan sepanjang perjalanan ini. Ucapan terima kasih secara khusus penulis tujukan kepada:

1. Tuhan Yang Maha Esa, atas kasih dan penyertaan-Nya yang senantiasa membimbing penulis hingga laporan ini terselesaikan.
2. Kedua orang tua dan keluarga tercinta, atas segala dukungan moral, semangat, dan pengorbanan yang tak ternilai selama proses penyusunan tugas akhir ini.
3. Bapak Julianto Lemantara, S.Kom., M.Eng., selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika, atas dukungannya terhadap proses akademik ini.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer dan Dosen Pembahas, atas arahan dan masukan yang sangat berharga untuk penyempurnaan laporan ini.
5. Bapak Heri Pratikno, M.T. selaku Dosen Pembimbing 1, yang telah membimbing secara teknis dan memberikan panduan yang sangat membantu.
6. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku Dosen Pembimbing 2 sekaligus Dosen Wali, atas semangat, bimbingan, dan kesabaran beliau selama proses penulisan tugas akhir ini.
7. Seluruh rekan S1 Teknik Komputer Angkatan 2021, atas kerja sama dan semangat kebersamaan yang telah dibangun selama masa studi.
8. Semua pihak yang tidak dapat disebutkan satu per satu, namun turut memberikan dukungan baik secara langsung maupun tidak langsung selama proses penyusunan laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, segala bentuk masukan dan koreksi yang bersifat membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat serta menjadi kontribusi kecil bagi kemajuan ilmu pengetahuan dan teknologi.

Surabaya, 25 Juli 2025

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
BAB II LANDASAN TEORI	6
2.1 Sampah	6
2.2 IP Camera	7
2.3 Computer Vision	8
2.4 YOLO	8
2.5 Object Tracking	10
2.6 Threading.....	11
2.7 Python.....	11
2.8 Visual Studi Code (VS Code)	12
2.9 Google Colaboratory (Google Colab)	12
2.10 Roboflow	13
2.11 Google Drive.....	13
2.12 Google Sheets	14
BAB III METODOLOGI PENELITIAN.....	15
3.1 Studi Literatur.....	15
3.2 Pendekatan Penelitian Terhadap Perilaku Membuang Sampah	15
3.3 Desain Sistem	15
3.4 Pencarian Dataset	18
3.5 Pengolahan Dataset	20

3.6	Proses Pelatihan Model	22
3.7	Diagram Blok Sistem	24
3.8	Pembuatan Sistem	24
3.9	Pengujian dan Evaluasi Sistem.....	25
3.10	Implementasi dan Uji Coba Sistem.....	25
BAB IV HASIL DAN PEMBAHASAN.....		25
4.1	Validasi Proses Training Data.....	25
4.1.1	Langkah-Langkah Validasi Proses Training Data	25
4.1.2	Hasil Validasi Training Data.....	25
4.2	Pengujian Sistem Dalam Mendeteksi Manusia dan Objek.....	28
4.2.1	Metode Mendeteksi Manusia dan Objek.....	28
4.2.2	Prosedur Pengujian Mendeteksi Manusia dan Objek.....	30
4.2.3	Hasil Pengujian Mendeteksi Manusia dan Objek.....	30
4.3	Pengujian Deteksi Pembuangan Sampah Sembarangan.....	33
4.3.1	Metode Deteksi Pembuangan Sampah Sembarangan	34
4.3.2	Prosedur Pengujian Mendeteksi Pembuangan Sampah.....	34
4.3.3	Hasil Deteksi Pembuangan Sampah Sembarangan	35
4.4	Pengujian Kecepatan Pengiriman Data	38
4.4.1	Metode Pengujian Kecepatan Pengiriman Data	38
4.4.2	Prosedur Pengujian Kecepatan Pengiriman Data	39
4.4.3	Hasil Pengujian Kecepatan Pengiriman Data.....	39
BAB V PENUTUP		40
5.1	Kesimpulan.....	40
5.2	Saran.....	41
DAFTAR PUSATAKA		42
LAMPIRAN.....		43

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Pembuangan sampah sembarangan.....	6
Gambar 2.2 IP Camera	7
Gambar 2.3 Arsitektur YOLOv8	10
Gambar 2.4 Logo Python	12
Gambar 2.5 Logo VS Code	12
Gambar 2.6 Logo Google Colab	13
Gambar 2.7 Logo Roboflow	13
Gambar 2.8 Logo Google Drive.....	14
Gambar 2.9 Logo Google Sheets	14
Gambar 3.1 Flowchart Algoritma sistem	16
Gambar 3.2 Contoh jarak 800 piksel.....	17
Gambar 3.3 Perancangan alat.....	17
Gambar 3.4 Device yang digunakan	18
Gambar 3.5 Dataset manusia.....	19
Gambar 3.6 Dataset objek.....	19
Gambar 3.7 Pelabelan dataset manusia.....	20
Gambar 3.8 Pelabelan dataset objek	20
Gambar 3.9 Proses Train/Test Split.....	21
Gambar 3.10 Proses Preprocessing	21
Gambar 3.11 Tahap Pemilihan Format Model	22
Gambar 3.12 Contoh API Key	22
Gambar 3.13 Download Dataset ke Google Colab	23
Gambar 3.14 Hasil Training di Google Colab	23
Gambar 3.15 Diagram blok sistem	24
Gambar 4.1 Dua Bounding Box manusia	29
Gambar 4.2 Pengujian pada siang hari.....	29
Gambar 4.3 Pengujian pada malam hari	30
Gambar 4.4 Hasil deteksi siang hari	34
Gambar 4.5 Hasil deteksi malam hari	34

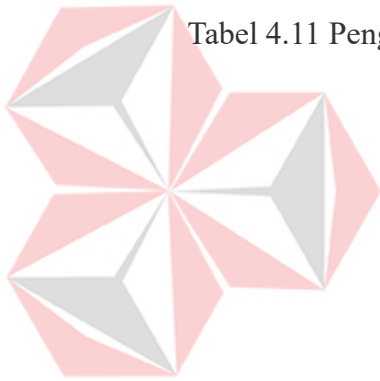
Gambar 4.6 Waktu pelanggaran terdeteksi	38
Gambar 4.7 Waktu berhasil tersimpan ke Google Sheet.....	38



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 4.1 Tabel evaluasi loss pelatihan model YOLOv8 untuk manusia.....	26
Tabel 4.2 Tabel evaluasi loss pelatihan Model YOLOv8 untuk objek.....	26
Tabel 4.3 Tabel performa deteksi model YOLOv8 untuk manusia.....	27
Tabel 4.4 Tabel performa deteksi model YOLOv8 untuk objek	27
Tabel 4.5 Tabel akurasi training deteksi manusia	27
Tabel 4.6 Tabel akurasi training deteksi objek	28
Tabel 4.7 Tabel pengujian deteksi pada siang hari.....	31
Tabel 4.8 Tabel pengujian deteksi pada malam hari	32
Tabel 4.9 Tabel pengujian pada siang hari	35
Tabel 4.10 Tabel pengujian pada malam hari.....	37
Tabel 4.11 Pengujian pengiriman data	39



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Program Deteksi Pembuangan Sampah Sembarangan.....	43
Lampiran 2. Kartu Konsultasi Bimbingan Tugas Akhir	49
Lampiran 3. Bukti Originalitas	50
Lampiran 4. Biodata diri	52



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Permasalahan sampah merupakan salah satu tantangan utama dalam menjaga kebersihan lingkungan, terutama di area publik seperti taman, jalanan, dan fasilitas umum lainnya. Pembuangan sampah sembarangan oleh manusia masih menjadi kebiasaan yang sulit dihilangkan, meskipun telah tersedia tempat sampah di berbagai lokasi. Kebiasaan ini dapat menyebabkan pencemaran lingkungan, penyumbatan saluran air, gangguan kualitas udara, menurunnya kenyamanan dan kualitas hidup masyarakat suatu wilayah.

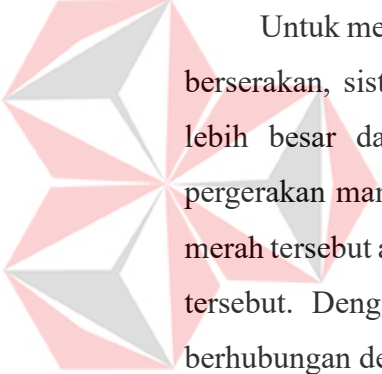
Upaya untuk mengatasi permasalahan ini sering kali memerlukan pengawasan yang intensif, baik melalui petugas keamanan, seperti satpam dan Satpol PP, maupun dengan pemanfaatan teknologi. Salah satu cara yang umum digunakan adalah pemantauan melalui kamera CCTV konvensional, namun untuk memperoleh bukti aktivitas pembuangan sampah sembarangan, hasil rekaman yang biasanya berdurasi berjam-jam harus diperiksa secara manual. Oleh karena itu, diperlukan sistem berbasis teknologi yang cerdas dan otomatis guna meningkatkan efisiensi dalam mendeteksi serta mendokumentasikan aktivitas pembuangan sampah sembarangan.

Dengan berkembangnya teknologi, solusi inovatif dapat diterapkan untuk membantu mendeteksi dan memantau perilaku manusia dalam membuang sampah sembarangan. Pada Penelitian yang dilakukan oleh (Yuskar, 2024), Penggunaan modul ESP32 CAM dan Arduino Mega 2560 berbasis IoT digunakan untuk mendeteksi dan mengawasi pelaku pembuangan sampah sembarangan.

Lalu pada penelitian yang dilakukan oleh (Aisyah, 2023), Sistem deteksi aktivitas manusia dalam membuang sampah menggunakan YOLO. Tingkat akurasi mencapai 90% pada siang dan 77.5% pada malam hari, sistem ini memiliki keterbatasan karena cenderung mengidentifikasi benda yang masih dibawa oleh manusia sebagai sampah.

Di sisi lain, pada penelitian yang dilakukan (Honnainah & Enggar Pawening, 2023), menggunakan YOLOv5 dalam mendeteksi pelanggaran pembuangan sampah sembarangan. Hasil penelitian menunjukkan bahwa metode ini berhasil mendeteksi aktivitas pembuangan sampah sembarangan dalam gambar dengan akurasi 95%. Namun, ketika diuji dengan video, deteksi tidak berhasil.

Dengan mempertimbangkan tiga penelitian tersebut, penelitian ini akan mengembangkan sistem deteksi pembuangan sampah sembarangan dengan menggunakan *IP Camera* (Internet Protocol Camera). Sistem akan menggunakan metode penghitungan jarak berbasis perbandingan piksel dalam citra, sebagaimana diterapkan dalam penelitian (Basyuni & Sugandi, 2022), mengenai pendeteksian jarak antar antrian, serta digunakan rumus Euclidian dari (Kristianto et al., 2025) penelitian tentang pengukuran ruang dimensi objek secara otomatis untuk memastikan apakah seseorang benar-benar membuang sampah.



Untuk membedakan antara objek yang dibawa manusia dan objek yang sudah berserakan, sistem akan memberi *bounding box* baru berwarna merah yang 5% lebih besar dari *bounding box* manusia. *Bounding box* ini akan mengikuti pergerakan manusia, dan setiap objek yang tumpang tindih dengan *bounding box* merah tersebut akan ditandai serta diberi ID unik yang dikaitkan dengan ID manusia tersebut. Dengan pendekatan ini, sistem dapat lebih fokus pada objek yang berhubungan dengan manusia.

Selain itu, sistem ini juga akan menerapkan metode *object tracking* untuk melacak pergerakan manusia dan sampah dari satu frame ke frame berikutnya guna memastikan bahwa ID objek tetap konsisten selama proses *tracking*. Berdasarkan penelitian (Abouelyazid, 2023), yang membandingkan metode *object tracking* seperti *SORT*, *DeepSORT*, dan *ByteTrack*, *ByteTrack* menunjukkan performa terbaik dalam menjaga identitas objek secara konsisten. Dengan keunggulan tersebut, implementasi Algoritma *ByteTrack* dalam penelitian ini diharapkan dapat meningkatkan akurasi dan efisiensi dalam mendeteksi pembuangan sampah sembarangan, serta memastikan bahwa setiap objek yang terdeteksi dapat dilacak secara stabil.

Sistem ini juga akan dilengkapi dengan mekanisme dokumentasi otomatis untuk menyimpan bukti pembuangan sampah sembarangan. Setiap pelanggaran

yang terdeteksi akan diabadikan dalam bentuk gambar, kemudian disimpan di Penyimpanan Lokal, Google Drive, dan Google Sheets untuk keperluan analisis lebih lanjut serta sebagai bukti valid dalam penegakan peraturan terkait kebersihan lingkungan serta. Sistem juga akan menghapus *bounding box* objek yang tidak relevan, seperti sampah yang sudah dibuang manusia guna mengoptimalkan proses analisis video secara *real-time*.

Dengan demikian, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem deteksi perilaku pembuangan sampah sembarangan menggunakan AI yang diintegrasikan dengan sistem pemantauan berbasis *IP Camera*. Dengan adanya sistem ini, diharapkan dapat membantu pihak terkait untuk meningkatkan efektivitas pengawasan, mempercepat respons terhadap pelanggaran, serta menyediakan data yang dapat digunakan sebagai dasar dalam pengambilan kebijakan terkait pengelolaan sampah dan kebersihan lingkungan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana sistem dapat mengetahui manusia yang membuang sampah sembarangan?
2. Bagaimana merancang sistem mendeteksi pembuangan sampah sembarangan berbasis *computer vision* yang berjalan secara *real-time*?
3. Bagaimana cara sistem mempertahankan *ID* agar identitas manusia dan objek tetap konsisten pada setiap *frame*?
4. Bagaimana mekanisme penyimpanan, pengelolaan, dan akses data hasil deteksi agar dapat digunakan untuk analisis?
5. Bagaimana cara agar sistem dapat menandai objek manusia serta objek yang dibawa oleh manusia?

1.3 Batasan Masalah

Agar penelitian ini lebih terarah dan fokus, terdapat beberapa batasan masalah yang ditetapkan sebagai berikut:

1. Sistem hanya mendeteksi pembuangan sampah sembarangan tanpa mempertimbangkan keberadaan tempat sampah.
2. Sistem ini hanya berfungsi sebagai alat pemantauan dan dokumentasi, tanpa adanya mekanisme pemberian sanksi atau tindakan hukum terhadap pelanggar.
3. Akurasi deteksi bergantung pada kualitas rekaman *IP Camera*, pencahayaan, serta kondisi lingkungan sekitar.
4. Sistem tidak dapat mengklasifikasikan jenis sampah yang dibuang, melainkan hanya mendeteksi tindakan pembuangan sampah sembarangan.
5. Sistem dapat bekerja secara akurat jika *bounding box* manusia tidak berdekatan.
6. Dalam penelitian ini, tas plastik berwarna putih digunakan sebagai objek sampah untuk pengujian sistem.
7. Sistem tidak dapat mendeteksi jika sampah dilempar terlalu kencang.

1.4 Tujuan

Berikut adalah tujuan dari penelitian ini berdasarkan latar belakang dan rumusan masalah yang telah disusun:

1. Sistem dapat mengetahui manusia yang membuang sampah sembarangan melalui perbandingan piksel (*Euclidean Distance*) antara manusia dan objek yang dibawa.
2. Mampu merancang sistem deteksi pembuangan sampah sembarangan berbasis *computer vision* yang berjalan secara *real-time*, dengan pengembangan dilakukan menggunakan *Visual Studio Code*.
3. Sistem dapat mempertahankan *ID* agar identitas manusia dan objek tetap konsisten pada setiap *frame* dengan menggunakan algoritma *ByteTrack*.
4. Dapat menyimpan dan mengelola data hasil deteksi proses pembuangan sampah secara sembarangan pada Google Sheets dan penyimpanan secara lokal.
5. Sistem mampu menandai objek manusia dan objek yang dibawanya dalam bentuk *bounding box*.

1.5 Manfaat

Berikut beberapa manfaat yang dapat diperoleh dari penelitian ini:

1. Membantu pihak terkait dalam melakukan tindakan yang diperlukan berdasarkan data hasil pemantauan pembuangan sampah sembarangan.
2. Mempermudah dokumentasi kejadian pelanggaran kebersihan, yang dapat digunakan untuk evaluasi kebijakan atau pengambilan langkah strategis lebih lanjut.



UNIVERSITAS
Dinamika

BAB II

LANDASAN TEORI

2.1 Sampah

Permasalahan sampah menjadi salah satu tantangan utama dalam menjaga kebersihan lingkungan, terutama di kawasan perkotaan yang memiliki tingkat populasi tinggi dan aktivitas industri yang padat. Pembuangan sampah sembarangan oleh manusia semakin memperburuk kondisi ini. Sampah yang tidak terkelola dengan baik dapat menyebabkan berbagai dampak negatif, seperti pencemaran lingkungan, penyumbatan saluran air, dan gangguan kesehatan masyarakat. Contoh pembuangan sampah sembarangan oleh manusia terdapat pada Gambar 2.1.



Gambar 2.1 Pembuangan sampah sembarangan
(Sumber: <https://www.rri.co.id/daerah/313542/banyak-sampah-di-pinggir-jalan-perlu-kesadaran-bersama>)

Di Indonesia, permasalahan sampah juga semakin bertambah. Data dari Kementerian Lingkungan Hidup dan Kehutanan menunjukkan bahwa pada tahun 2016, jumlah timbulan sampah di Indonesia telah mencapai 65.2 juta ton per tahun. Timbulan sampah ini juga berkontribusi secara signifikan terhadap bencana banjir yang terus meningkat serta kejadian luar biasa diare dengan angka kematian mencapai 3.04% (Verawati, 2022).

Untuk membantu mengatasi masalah ini, salah satu solusi inovatif yang dapat diterapkan adalah pemanfaatan teknologi kecerdasan buatan *AI* (*Artificial Intelligence*) untuk mendeteksi dan memantau perilaku pembuangan sampah

sembarangan. Dengan adanya sistem berbasis AI yang terintegrasi dengan *IP Camera*, proses pemantauan dapat dilakukan secara otomatis, sehingga membantu pihak terkait dalam meningkatkan efektivitas pengawasan dan mendukung pengambilan keputusan terkait pengelolaan kebersihan lingkungan.

2.2 *IP Camera*

IP Camera (Internet Protocol Camera) adalah kamera pengawas yang dapat mengirimkan data video melalui jaringan internet atau lokal tanpa memerlukan sistem kabel tertutup seperti CCTV konvensional. *IP Camera* banyak digunakan dalam berbagai sektor, seperti keamanan, transportasi, dan pemantauan lingkungan, karena kemampuannya untuk diakses secara remote dan fleksibilitas dalam integrasi sistem. Gambar 2.2 merupakan contoh *IP Camera* yang dipakai untuk memperoleh rekaman video.



Gambar 2.2 *IP Camera*

Meskipun berbeda dalam teknologi transmisi data, *IP Camera* memiliki fungsi yang serupa dengan CCTV dalam hal pemantauan dan perekaman aktivitas. Dalam hukum pidana Indonesia, rekaman CCTV dapat digunakan sebagai alat bukti elektronik. Hal ini diatur dalam Undang-Undang No. 19 Tahun 2016 tentang Informasi dan Transaksi Elektronik, yang menyatakan bahwa alat bukti elektronik, termasuk rekaman CCTV, diakui sebagai alat bukti sah sesuai hukum acara yang berlaku (Ponto et al., 2024). Oleh karena itu, dalam penelitian ini, *IP Camera* dimanfaatkan sebagai alat untuk merekam video secara *real-time*

2.3 *Computer Vision*

Computer Vision merupakan salah satu cabang dari *AI (Artificial Intelligence)* yang berfokus pada bagaimana komputer dapat memahami dan menafsirkan informasi dari gambar atau video. Pada dasarnya, *Computer Vision* berusaha meniru cara kerja sistem penglihatan manusia. Human vision sesungguhnya sangat kompleks, manusia melihat objek dengan indera penglihatan lalu obyek citra diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya (Afni et al., 2021).

Dengan prinsip yang serupa, *Computer Vision* memproses gambar atau video melalui serangkaian tahap yang meniru mekanisme penglihatan manusia. Gambar yang diperoleh akan diubah menjadi data digital. Kemudian, tahap pra-pemrosesan dilakukan untuk meningkatkan kualitas gambar, seperti mengurangi noise atau mengubah skala warna. Setelah itu, algoritma pemrosesan citra mengekstraksi fitur penting dari gambar, seperti bentuk, tekstur, atau warna. Fitur-fitur ini kemudian dianalisis menggunakan teknik *Deep Learning*, di mana model yang telah dilatih dapat mengenali pola dan mengklasifikasikan objek sesuai dengan kategori yang telah ditentukan.

2.4 YOLO

YOLO (*You Only Look Once*) adalah salah satu metode deteksi objek berbasis *Deep Learning* yang dikenal karena kecepatannya dalam mengenali objek dalam gambar atau video. Berbeda dengan metode deteksi objek R-CNN dan Faster R-CNN yang memproses gambar secara bertahap, YOLO melakukan deteksi dan klasifikasi dalam satu tahap, sehingga menghasilkan proses yang jauh lebih cepat dan efisien. Oleh karena itu, YOLO sering digunakan dalam aplikasi *real-time* yang membutuhkan deteksi objek secara instan, seperti sistem pengawasan, dan kendaraan otonom.

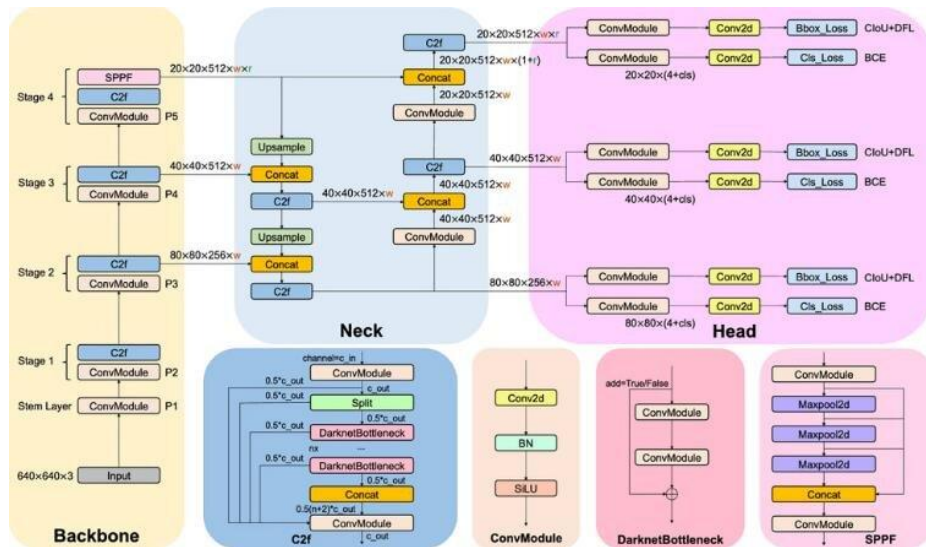
YOLO bekerja dengan membagi gambar menjadi beberapa grid, di mana setiap grid bertanggung jawab untuk mendeteksi objek yang terdapat di dalamnya. Model ini menggunakan jaringan saraf CNN untuk menganalisis fitur dalam gambar dan menghasilkan prediksi dalam bentuk *bounding box* (kotak pembatas) serta confidence score (tingkat keyakinan model terhadap objek yang terdeteksi).

Sejak pertama kali diperkenalkan, YOLO telah mengalami beberapa pengembangan versi untuk meningkatkan akurasi dan efisiensi deteksi objek. Versi-versi utama YOLO meliputi:

1. YOLOv1: Model pertama yang memperkenalkan konsep deteksi objek dalam satu tahap dengan kecepatan tinggi, tetapi memiliki keterbatasan dalam akurasi deteksi objek kecil.
2. YOLOv2: Peningkatan pada akurasi dengan menggunakan batch normalization dan anchor boxes.
3. YOLOv3: Peningkatan dalam deteksi multi-skala dengan fitur prediksi di berbagai resolusi.
4. YOLOv4: Peningkatan dalam efisiensi dan akurasi dengan penggunaan CSPDarknet53 sebagai *backbone*.
5. YOLOv5: Model yang lebih ringan dan lebih cepat dengan optimasi pada ukuran dan arsitektur.
6. YOLOv6 & YOLOv7: Peningkatan lebih lanjut pada kecepatan dan akurasi dengan teknik seperti anchor-free detection.
7. YOLOv8: Versi terbaru yang menggabungkan berbagai inovasi dari versi sebelumnya serta menambahkan teknik *Deep Learning* terbaru untuk meningkatkan performa deteksi.

Penelitian ini menggunakan YOLOv8 karena beberapa keunggulan utama, antara lain:

1. YOLOv8 memiliki optimasi pada arsitektur jaringan yang memungkinkan deteksi objek dengan presisi lebih baik dibandingkan versi sebelumnya.
2. Model ini mampu mencapai keseimbangan antara kecepatan inferensi dan akurasi, menjadikannya cocok untuk aplikasi *real-time*.
3. YOLOv8 dikembangkan dan didukung oleh *Ultralytics*, yang menyediakan pustaka, dokumentasi, dan alat bantu untuk mempermudah implementasi dalam berbagai aplikasi.



Gambar 2.3 Arsitektur YOLOv8
(Sumber: Zhorif et al., 2024)

Gambar 2.3 menunjukkan arsitektur YOLOv8, yang terdiri dari tiga bagian utama: *Backbone*, *Neck*, dan *Head*. Berikut adalah penjelasan dari masing-masing komponen dalam arsitektur ini:

1. *Backbone (Ekstraksi Fitur)* : *Backbone* bertugas mengekstrak fitur penting dari gambar masukan ($640 \times 640 \times 3$). Pada YOLOv8, *backbone* ini menggunakan *C2f* (*Cross-Stage Partial with Fusion*) yang merupakan pengembangan dari *CSP* (*Cross Stage Partial Network*) yang lebih efisien.
2. *Neck (Penguat Fitur)* : bertanggung jawab untuk menggabungkan fitur dari berbagai resolusi agar model mampu mendeteksi objek dalam berbagai ukuran.
3. *Head (Prediksi Kelas dan Bounding Box)* : bertugas menghasilkan *bounding box* dan kelas objek dari fitur yang telah diproses oleh *Backbone* dan *Neck*.

2.5 Object Tracking

Object *tracking* adalah salah satu cabang dalam *Computer Vision* yang bertujuan untuk melacak pergerakan suatu objek dari satu frame ke frame lainnya dalam sebuah video. Object *Tracking* ada beberapa jenis, contohnya: *SORT* (*Simple Online and Realtime Tracker*), *DeepSORT* (*Deep Simple Online and Realtime Tracker*), dan *ByteTrack*.

Menurut penelitian yang dilakukan oleh (Abouelyazid, 2023), hasil menunjukkan bahwa *ByteTrack* mengungguli *SORT* dan *DeepSORT* dengan *MOTA*

(*Multiple Object Tracking Accuracy*) sebesar 77,3% dan *MOTP (Multiple Object Tracking Precision)* sebesar 82,6%. Selain itu, *ByteTrack* memiliki jumlah kesalahan identifikasi terendah, yaitu 558, serta kecepatan pemrosesan tinggi, mencapai 171 FPS (Frame Per Second), lebih cepat dibandingkan dengan *SORT* dan *DeepSORT*.

Oleh karena itu, penelitian ini akan menggunakan *ByteTrack* untuk melacak manusia dan objek yang terdeteksi, karena algoritma ini memiliki akurasi tinggi dalam mempertahankan identitas objek (MOTA 77,3%), presisi tinggi dalam menentukan posisi objek (MOTP 82,6%), serta kecepatan pemrosesan yang optimal (171 FPS). Kemampuan ini menjadikan *ByteTrack* sebagai pilihan yang tepat untuk aplikasi pemantauan *real-time*.

2.6 Threading

Dalam pengembangan aplikasi, pengiriman data menjadi salah satu aspek penting, terutama saat data hasil pemrosesan perlu ditransfer ke sistem lain, seperti server atau basis data. Di sinilah konsep threading berperan. Threading memungkinkan eksekusi beberapa tugas secara konkuren dalam sebuah program melalui penggunaan thread, yaitu alur eksekusi independen. Dalam konteks Python, yang sering digunakan bersama *Ultralytics*, *threading* sangat berguna untuk menangani tugas-tugas yang bersifat I/O, seperti membaca data dari kamera, memprosesnya dengan YOLOv8, dan mengirim hasilnya ke sistem eksternal. Dalam penelitian ini, threading dimanfaatkan untuk memastikan bahwa proses pengiriman data hasil deteksi ke Google Sheets, Google Drive, dan penyimpanan lokal berjalan secara konkuren tanpa menghentikan sistem deteksi, sehingga aplikasi tetap beroperasi secara *real-time*.

2.7 Python

Python adalah bahasa pemrograman Guido van Rossum dan pertama kali dirilis pada tahun 1991. Python bersifat interpreted, yang berarti kode dapat dieksekusi langsung tanpa perlu dikompilasi terlebih dahulu. Python memiliki ekosistem pustaka yang luas, seperti NumPy dan Pandas untuk analisis data, TensorFlow dan PyTorch untuk kecerdasan buatan, serta OpenCV untuk

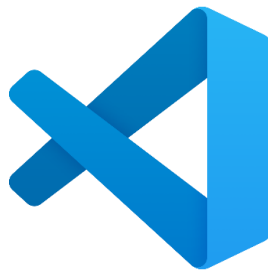
pemrosesan citra. Dengan fleksibilitas dan kemampuannya yang luas, Python telah menjadi bahasa yang populer dalam berbagai bidang, termasuk pengembangan web, data science, machine learning, dan otomasi sistem. Bahasa python dipilih oleh peneliti karena kemudahan penggunaannya, fleksibilitas, dan ekosistem pustaka yang luas. Sintaksnya yang sederhana memungkinkan peneliti fokus pada logika dan analisis tanpa harus terhambat oleh kompleksitas bahasa pemrograman. Logo python terdapat pada Gambar 2.4.



Gambar 2.4 Logo Python
(Sumber: <https://www.python.org/>)

2.8 Visual Studi Code (VS Code)

Visual Studio Code (VS Code) adalah editor kode sumber yang dikembangkan oleh Microsoft, dirancang untuk mendukung berbagai bahasa pemrograman dan menyediakan fitur canggih seperti penyorotan sintaks, debugging, IntelliSense (penyelesaian kode otomatis), terminal terintegrasi, dan sistem ekstensi yang fleksibel. Pada penelitian ini VS Code dipilih sebagai lingkungan pengembangan dalam pembuatan sistem deteksi pembuangan sampah sembarangan berbasis *computer vision* yang dapat berjalan secara *real-time*. Logo VS Code terdapat pada Gambar 2.5.



Gambar 2.5 Logo VS Code
(Sumber: <https://code.visualstudio.com/brand/>)

2.9 Google Colaboratory (Google Colab)

Google Colaboratory (Google Colab) adalah layanan berbasis cloud yang memungkinkan pengguna menulis dan menjalankan kode Python langsung dari

browser tanpa perlu instalasi tambahan. Platform ini sangat populer di kalangan peneliti, terutama dalam bidang machine learning, *Deep Learning*, dan analisis data, karena menyediakan akses ke GPU dan TPU gratis, yang mempercepat komputasi berbasis AI. Dalam penelitian ini, *Google Colab* digunakan untuk melatih dan menguji model dengan YOLOv8 dalam mendeteksi keberadaan manusia dan kantong sampah. Logo *Google Colab* terdapat pada Gambar 2.6.



Gambar 2.6 Logo *Google Colab*
(Sumber: <https://colab.google/>)

2.10 Roboflow

Roboflow adalah platform yang dirancang untuk mempermudah pengolahan dan manajemen dataset dalam pengembangan model *Computer Vision*. Platform ini memungkinkan pengguna untuk mengunggah, mengelola, serta melakukan augmentasi data guna meningkatkan performa model. Dalam penelitian ini, *Roboflow* digunakan untuk mempersiapkan dataset yang akan digunakan dalam pelatihan model YOLOv8. Logo *Roboflow* terdapat pada Gambar 2.7.



Gambar 2.7 Logo *Roboflow*
(Sumber: <https://roboflow.com/brand/>)

2.11 Google Drive

Google Drive adalah layanan penyimpanan berbasis cloud yang memungkinkan pengguna untuk menyimpan, mengelola, dan berbagi berbagai jenis file secara online. Dalam penelitian ini, Google Drive digunakan sebagai salah satu tempat menyimpan hasil tangkapan layar dari *IP Camera* yang merekam aktivitas pembuangan sampah sembarangan. Logo Google Drive terdapat pada Gambar 2.8.



Gambar 2.8 Logo Google Drive

(Sumber: <https://icon-icons.com/id/icon/google-drive-logo/159334/>)

2.12 Google Sheets

Google Sheets adalah layanan spreadsheet berbasis cloud yang disediakan oleh Google sebagai bagian dari Google Workspace. Layanan ini memungkinkan pengguna untuk membuat, mengedit, dan berbagi spreadsheet secara *real-time* dengan kolaborator lain. Dalam penelitian ini, Google Sheets digunakan sebagai salah satu tempat menyimpan hasil tangkapan layar dari *IP Camera* yang merekam aktivitas pembuangan sampah sembarangan, sekaligus mencatat data waktu kejadian untuk mempermudah analisis dan pelacakan. Logo Google Sheets terdapat pada Gambar 2.9.



Google Sheets

Gambar 2.9 Logo Google Sheets

(Sumber: <https://seeklogo.com/vector-logo/332740/google-sheets/>)

BAB III

METODOLOGI PENELITIAN

3.1 Studi Literatur

Pada tahap ini, peneliti melakukan kajian literatur dari berbagai sumber informasi yang relevan dengan topik dan judul penelitian. Proses ini melibatkan pembacaan dan analisis berbagai referensi untuk memperoleh pemahaman yang mendalam serta menentukan metode dan algoritma yang tepat agar sistem dapat berjalan dengan optimal dan sesuai dengan tujuan penelitian.

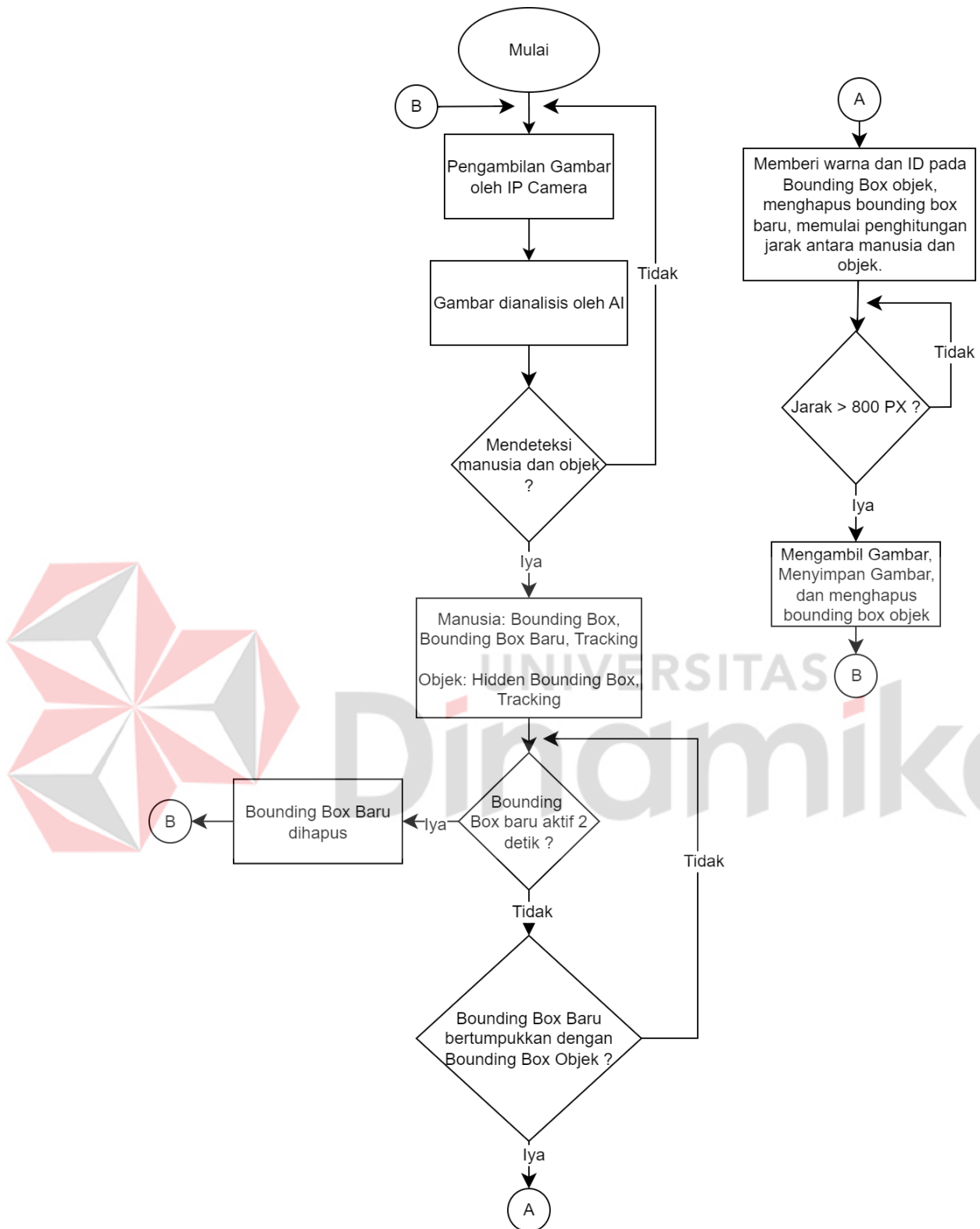
3.2 Pendekatan Penelitian Terhadap Perilaku Membuang Sampah

Membuang sampah sembarangan didefinisikan sebagai tindakan seseorang meletakkan, menjatuhkan, atau membuang sampah di area yang bukan diperuntukkan untuk pembuangan, seperti jalanan, taman, selokan, atau ruang publik lainnya. Dalam penelitian ini, perilaku membuang sampah sembarangan dikenali melalui ciri utama, yaitu tindakan melepas objek yang dipegang secara sengaja atau lalai di luar lokasi pembuangan sampah.

Berdasarkan karakteristik tersebut, sistem yang dikembangkan menggunakan metode pendeteksian objek berbasis YOLOv8 untuk mengidentifikasi keberadaan manusia dan objek yang dibawanya. Setelah objek terdeteksi, sistem menghitung jarak antara manusia dan objek yang dibawa dengan perbandingan piksel menggunakan *Euclidean Distance*. Jika sistem mendeteksi jarak antara manusia dan objek yang dibawa sebelumnya melebihi dari batas yang sudah ditetapkan, maka tindakan tersebut dikategorikan sebagai pembuangan sampah sembarangan.

3.3 Desain Sistem

Sistem ini dirancang untuk mendeteksi aktivitas pembuangan sampah menggunakan *IP Camera* dan teknologi AI berbasis YOLOv8. Algoritma sistem dapat dilihat pada *flowchart pada Gambar 3.1*.



Gambar 3.1 Flowchart Algoritma sistem

Penentuan jarak piksel bergantung pada kedekatan antara *IP Camera* dan objek manusia yang dideteksi. Dalam penelitian ini, jarak sebesar 800 piksel digunakan karena jarak tersebut telah sesuai untuk memastikan bahwa orang tersebut telah benar-benar membuang objek yang dibawa. Contoh jarak 800 piksel dapat dilihat pada Gambar 3.2.



Gambar 3.2 Contoh jarak 800 piksel



Gambar 3.3 Perancangan alat



Gambar 3.4 Device yang digunakan

Gambar 3.3 merupakan gambar perancangan alat untuk mendeteksi pelanggaran pembuangan sampah sembarangan, area warna hijau merupakan area yang diawasi IP Camera. Pengambilan video secara *real-time* akan diambil dari *IP Camera* yang dipasang dengan tinggi dan sudut yang disesuaikan. Area yang dideteksi ditentukan berdasarkan area yang tertangkap oleh *IP Camera*, sehingga cakupan deteksi bergantung pada sudut dan posisi pemasangan *IP Camera*. *IP Camera* dan device yang digunakan untuk proses deteksi harus terhubung ke jaringan internet yang sama. Gambar 3.4 merupakan device yang digunakan penulis untuk menjalankan sistem, device yang digunakan juga harus memiliki RTSP *IP Cam* yang sesuai agar dapat terhubung dengan *IP Camera*. Pendeteksian pelanggaran akan dilakukan melalui laptop, dan hasil deteksi pelanggaran akan disimpan ke Folder lokal, Google Drive, dan Google Sheets.

3.4 Pencarian Dataset

Dataset yang akan dikumpulkan terdiri dari gambar manusia dan kantong sampah pada kondisi siang hari dan malam hari. Untuk dataset manusia di dapatkan dari *Roboflow Universe* sebanyak 9665 buah dan untuk dataset objek akan sebanyak 9168 buah, didapat secara mandiri menggunakan kamera pribadi. Proses pengumpulan dataset objek dilakukan dengan menfoto kantong plastik berwarna putih tersebut dari berbagai arah dan kondisi, memvideo dan diubah ke foto ketika membawa kantong plastik dari pandangan *IP Camera* dan memvideo diri sambil

melempar objek untuk mendapatkan kondisi nyata ketika sampah dilempar. Untuk dataset manusia dapat dilihat pada Gambar 3.5 dan untuk dataset objek pada Gambar 3.6.



Gambar 3.5 Dataset manusia

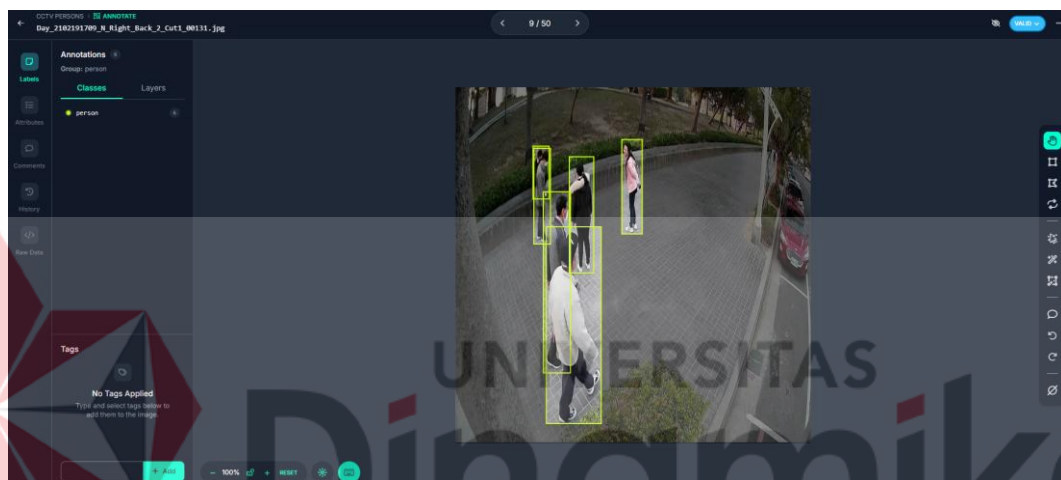


Gambar 3.6 Dataset objek

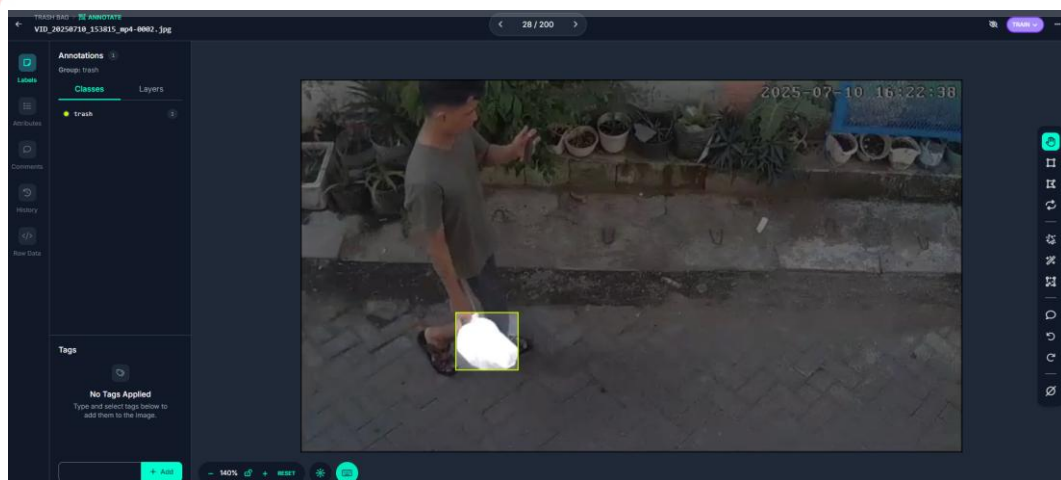
Dataset yang terkumpul akan diproses lebih lanjut di Roboflow, sebelum digunakan dalam pelatihan model YOLOv8 di Google Colab. Proses ini bertujuan untuk meningkatkan akurasi model dalam mendeteksi manusia dan kantong sampah dengan lebih optimal.

3.5 Pengolahan Dataset

Pada tahap ini, semua dataset yang telah dikumpulkan akan diproses lebih lanjut menggunakan *Roboflow*, sebuah platform berbasis cloud yang digunakan untuk pelabelan, augmentasi, dan preprocessing dataset sebelum digunakan dalam pelatihan model YOLOv8. Tahapan pertama dalam pengolahan dataset adalah pelabelan, di mana setiap gambar akan diberi *bounding box* untuk menandai objek yang relevan, seperti manusia dan kantong sampah. Proses ini bertujuan agar model dapat mengenali dan membedakan objek dengan lebih akurat. Contoh pelabelan manusia dapat dilihat pada Gambar 3.7 dan pelabelan objek pada Gambar 3.8.



Gambar 3.7 Pelabelan dataset manusia



Gambar 3.8 Pelabelan dataset objek

Dataset yang telah dilabeli selanjutnya akan dibagi melalui proses Train/Test Split. Tahap ini membagi dataset menjadi tiga bagian, yaitu train set, valid set, dan

test set, dengan proporsi masing-masing 70%, 20%, dan 10%. Proses pembagian Train/Test Split ini dapat dilihat pada gambar 3.9.

2 Train/Test Split
Here is how you split your images when you added them to the dataset:

TRAIN SET	VALID SET	TEST SET
6188 Images	1741 Images	922 Images

70% 20% 10%

Continue Rebalance

Gambar 3.9 Proses Train/Test Split

Lalu berikutnya, dataset akan diproses lebih lanjut melalui preprocessing, yang mencakup resize agar gambar seragam dan sesuai dengan standar yang digunakan dalam pelatihan model YOLOv8, sehingga mempercepat proses komputasi tanpa kehilangan informasi penting, dan Auto-Orient untuk Menyesuaikan orientasi gambar secara otomatis agar semua gambar memiliki posisi yang seragam, dan menghindari kesalahan deteksi. Proses preprocessing dapat dilihat pada gambar 3.10.

Create New Version
Prepare your images and data for training by compiling them into a version. Experiment with different configurations to achieve better training results.

Source Images
Images: 7,015
Classes: 1
Unannotated: 0

Train/Test Split
Training Set: 5.5k images
Validation Set: 719 images
Testing Set: 836 images

3 Preprocessing
What can preprocessing do?
Decrease training time and increase performance by applying image transformations to all images in this dataset.

Auto-Orient Edit ×

Resize
Stretch to 640×640 Edit ×

+ Add Preprocessing Step

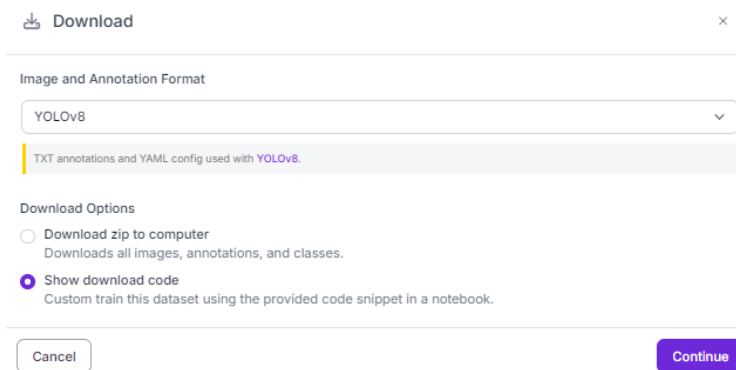
Continue

4 Augmentation @

5 Create

Gambar 3.10 Proses Preprocessing

Selanjutnya download dataset sesuai format model yang digunakan. Pada penelitian ini akan digunakan model YOLOv8 seperti pada gambar 3.11.



Gambar 3.11 Tahap Pemilihan Format Model

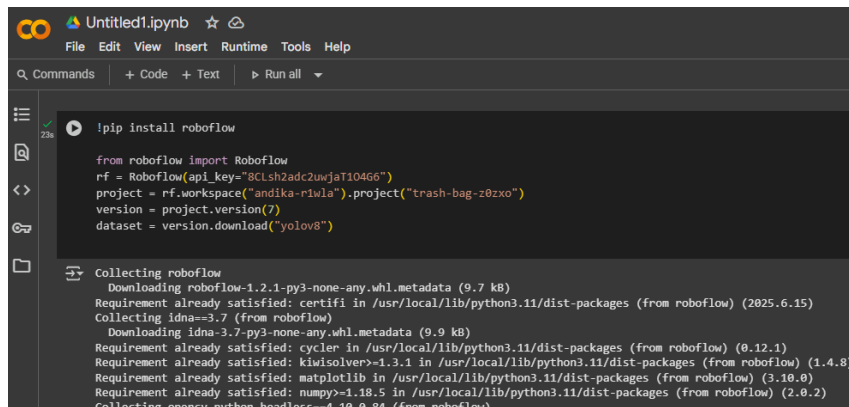
Roboflow akan memberi dua pilihan, yaitu download dalam bentuk file zip atau melalui API key. Pada penelitian ini akan memilih melalui API key, API key tersebut yang akan digunakan untuk mengunduh dataset ke *Google Colab*. Contoh API key dapat dilihat pada gambar 3.12.



Gambar 3.12 Contoh API Key

3.6 Proses Pelatihan Model

Pada tahap ini, dataset yang sudah diproses di *Roboflow akan didownload* ke *Google Colab* untuk tahap pelatihan seperti pada gambar 3.13.



```

!pip install roboflow

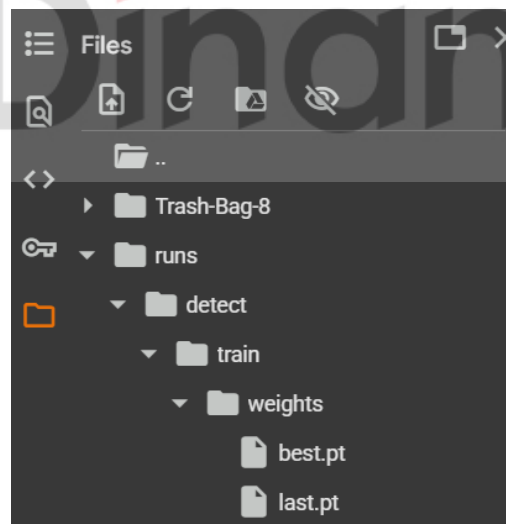
from roboflow import Roboflow
rf = Roboflow(api_key="8CLsh2adc2uwjaT104G6")
project = rf.workspace("andika-r1w1a").project("trash-bag-z0zxo")
version = project.version(7)
dataset = version.download("yolov8")

Collecting roboflow
  Downloading roboflow-1.2.1-py3-none-any.whl.metadata (9.7 kB)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from roboflow) (2025.6.15)
Collecting idna==3.7 (from roboflow)
  Downloading idna-3.7-py3-none-any.whl.metadata (9.9 kB)
Requirement already satisfied: cytoolz in /usr/local/lib/python3.11/dist-packages (from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from roboflow) (1.4.8)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from roboflow) (3.10.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.11/dist-packages (from roboflow) (2.0.2)
Collecting onenumpy-numpy-headless==4.10.0.84 (from roboflow)

```

Gambar 3.13 Download Dataset ke *Google Colab*

Pelatihan akan dilakukan secara terpisah untuk mendeteksi objek manusia dan kantong sampah dengan 3 learning rate yang berbeda, namun tetap menggunakan metode yang sama, yaitu model YOLOv8 dengan framework *Ultralytics*. Hasil pelatihan model akan berupa file dengan nama *best.pt* yang akan digunakan untuk *real-time* deteksi di VS Code. Untuk contoh hasil pelatihan dapat dilihat pada Gambar 3.14.

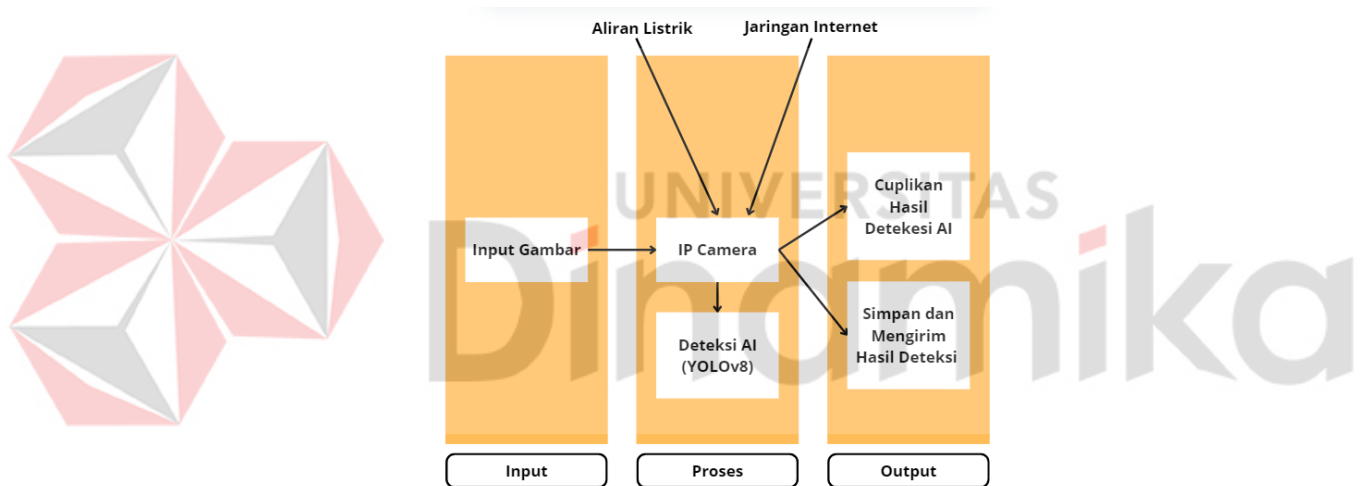


Gambar 3.14 Hasil Training di *Google Colab*

Selama pelatihan, GPU gratis dan Berbayar dari Google Colab akan digunakan untuk mempercepat proses komputasi.

3.7 Diagram Blok Sistem

Pada diagram blok Gambar 3.15, sistem ini dirancang untuk mendeteksi keberadaan manusia dan objek dalam area pengawasan *IP CAMERA* menggunakan teknologi AI secara *real-time*. pada tahap input, *IP Camera* yang dialiri listrik dan terhubung jaringan internet yang sama akan menangkap video secara *real-time* untuk memantau aktivitas di sekitarnya, khususnya dalam pengawasan terhadap pembuangan sampah. Selanjutnya, pada tahap proses, video yang diambil akan dianalisis menggunakan AI (YOLOv8) untuk mendeteksi manusia dan objek. Selanjutnya pada tahap Output, jika terdeteksi adanya manusia yang membuang sampah di area *IP Camera*, maka sistem akan mengambil gambar video tersebut yang kemudian akan disimpan ke Penyimpanan Lokal, Google Drive, dan Google Sheets untuk keperluan dokumentasi.



Gambar 3.15 Diagram blok sistem

3.8 Pembuatan Sistem

Pada tahap ini, sistem akan dikembangkan di VS Code dengan berbagai komponen utama untuk mendukung deteksi pembuangan sampah sembarangan. Pengembangan sistem mencakup deteksi manusia, deteksi objek yang dibawa manusia, penghapusan *bounding box* merah, penghitungan jarak antara manusia dan objek yang dibawa, *tracking* manusia dan objek, serta identifikasi tindakan pembuangan sampah sembarangan. Sistem juga akan menyimpan hasil deteksi dalam bentuk gambar ke penyimpanan lokal, Google Drive dan Google Sheets untuk dokumentasi dan analisis lebih lanjut.

3.9 Pengujian dan Evaluasi Sistem

Setelah pengembangan sistem selesai dilakukan di *Visual Studio Code* sesuai dengan rancangan pada tahap pembuatan sistem, tahap ini berfokus pada pengujian untuk memverifikasi bahwa setiap komponen berfungsi optimal. Pengujian mencakup kinerja model YOLOv8 dalam mendeteksi manusia dan objek yang dibawa, penghapusan *bounding box* merah, kemampuan sistem dalam menghitung jarak antara manusia dan objek untuk mengidentifikasi tindakan pembuangan sampah sembarangan, serta keakuratan object *tracking* untuk melacak pergerakan manusia dan objek. Hasil deteksi juga dipastikan apakah tersimpan dengan baik ke penyimpanan lokal, Google Drive, dan Google Sheets. Jika ditemukan kendala dalam proses pengujian, maka dilakukan perbaikan dan optimasi guna meningkatkan keandalan sistem sebelum implementasi penuh.

3.10 Implementasi dan Uji Coba Sistem

Setelah melalui tahap pengujian dan perbaikan, sistem kemudian diimplementasikan dalam lingkungan yang sesuai untuk menguji kinerjanya dalam kondisi nyata. Pengujian dilakukan menggunakan *IP Camera* yang dipasang pada ketinggian dan sudut yang disesuaikan dengan kondisi lingkungan. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem dapat beroperasi secara *real-time* dalam mendeteksi perilaku pembuangan sampah sembarangan serta melacak pergerakan objek dengan algoritma *ByteTrack*.

Pada tahap uji coba, sistem akan dioperasikan dalam kondisi siang hari dan malam hari, guna menilai keandalan sistem. Selain itu, dilakukan pengamatan terhadap proses penyimpanan dan pengiriman data untuk memastikan bahwa gambar hasil deteksi berhasil tersimpan di penyimpanan lokal, Google Drive, dan Google Sheets tanpa kendala.

Hasil dari implementasi ini dianalisis untuk mengidentifikasi kelebihan dan kekurangan sistem. Jika masih ditemukan hambatan, maka dilakukan perbaikan akhir hingga sistem dapat beroperasi lebih efektif dan optimal.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Validasi Proses Training Data

Bagian ini menjelaskan proses validasi yang dilakukan untuk mengevaluasi performa model YOLOv8 yang telah dilatih menggunakan platform *Google Colab*. Proses ini bertujuan untuk mengukur akurasi, ketepatan, dan keandalan model dalam mendeteksi objek berdasarkan data pelatihan yang digunakan. Evaluasi dilakukan pada setiap epoch untuk memantau perkembangan performa model selama proses pelatihan, sehingga dapat diidentifikasi apakah model mengalami peningkatan atau memerlukan penyesuaian parameter untuk mencapai hasil yang lebih baik.

4.1.1 Langkah-Langkah Validasi Proses Training Data

Berikut merupakan langkah-langkah dari proses validasi data:

1. Membuka web *Roboflow* lalu download dataset yang sudah diberi label menggunakan format sesuai dengan model yang digunakan pada pelatihan di *Google Colab*.
2. Meng-copy baris kode yang diberikan untuk mengunduh dataset melalui API (Application Programming Interface).
3. Membuka web *Google Colab* lalu tekan file dan buat folder baru dengan menekan new notebook in drive.
4. Mem-paste baris kode tersebut ke *Google Colab* lalu jalankan kode tersebut untuk mengunduh dataset ke *Google Colab*.
5. Selanjutnya meng-install environment yang diperlukan untuk training model
6. Lalu men-training model dengan jumlah epoch yang dimau, pada penelitian ini untuk manusia dilakukan 50 epoch dan untuk objek dilakukan 100 epoch.

4.1.2 Hasil Validasi Training Data

Setelah pelatihan selesai, kinerja model YOLOv8 akan dievaluasi performanya dalam mendeteksi manusia dan objek. Hasil evaluasi akan disajikan dalam dua tabel, yaitu tabel evaluasi loss untuk menilai akurasi prediksi dan tabel

performa model untuk mengukur ketepatan, kelengkapan, serta keandalan model secara keseluruhan. Hasil validasi untuk manusia dapat dilihat pada tabel 4.1 dan 4.3, lalu untuk hasil validasi objek dapat dilihat pada tabel 4.2 dan 4.4, dan untuk akurasi dapat dilihat pada tabel 4.5 untuk manusia dan 4.6 untuk objek.

Tabel 4.1 Tabel evaluasi loss pelatihan model YOLOv8 untuk manusia

Epoch	Learning Rate	Box_loss	Cls_loss	Dfl_loss
10	0.01	1.319	0.905	1.23
	0.005	1.306	0.8887	1.219
	0.0001	1.306	0.8887	1.219
20	0.01	1.17	0.7341	1.143
	0.005	1.166	0.7279	1.142
	0.001	1.166	0.7279	1.142
30	0.01	1.068	0.6388	1.092
	0.005	1.051	0.6254	1.081
	0.0001	1.051	0.6254	1.081
40	0.1	0.9704	0.5581	1.046
	0.005	0.9666	0.5577	1.045
	0.0001	0.9666	0.5577	1.045
50	0.1	0.8059	0.422	0.9737
	0.005	0.8042	0.4229	0.976
	0.0001	0.8042	0.4229	0.976

Tabel 4.2 Tabel evaluasi loss pelatihan Model YOLOv8 untuk objek

Epoch	Learning Rate	Box_loss	Cls_loss	Dfl_loss
20	0.001	0.897	0.4501	1.097
	0.0005	0.897	0.4501	1.097
	0.0001	0.897	0.4501	1.097
40	0.001	0.8016	0.3782	1.045
	0.0005	0.8016	0.3782	1.045
	0.0001	0.8016	0.3782	1.045
60	0.001	0.7268	0.3371	1.006
	0.0005	0.7268	0.3371	1.006
	0.0001	0.7268	0.3371	1.006
80	0.001	0.6483	0.2932	0.9686
	0.0005	0.6483	0.2932	0.9686
	0.0001	0.6483	0.2932	0.9686
100	0.001	0.4849	0.1971	0.8913
	0.0005	0.4849	0.1971	0.8913
	0.0001	0.4849	0.1971	0.8913

Tabel 4.3 Tabel performa deteksi model YOLOv8 untuk manusia

Epoch	Learning Rate	Precision	Recall	mAP50	mAP50-95	F1-Score
10	0.01	0.884	0.804	0.893	0.57	0.84
	0.005	0.889	0.814	0.897	0.577	0.85
	0.0001	0.889	0.814	0.897	0.577	0.85
20	0.01	0.919	0.845	0.925	0.633	0.88
	0.005	0.923	0.852	0.928	0.639	0.89
	0.0001	0.923	0.852	0.928	0.639	0.89
30	0.01	0.94	0.884	0.95	0.68	0.91
	0.005	0.936	0.881	0.949	0.681	0.91
	0.0001	0.936	0.881	0.949	0.681	0.91
40	0.01	0.947	0.907	0.961	0.717	0.93
	0.005	0.941	0.912	0.962	0.72	0.93
	0.0001	0.941	0.912	0.962	0.72	0.93
50	0.01	0.955	0.921	0.968	0.746	0.94
	0.005	0.957	0.912	0.912	0.749	0.93
	0.0001	0.957	0.912	0.968	0.749	0.93

Tabel 4.4 Tabel performa deteksi model YOLOv8 untuk objek

Epoch	Learning Rate	Precision	Recall	mAP50	mAP50-95	F1-Score
20	0.001	0.994	0.986	0.994	0.787	0.99
	0.0005	0.994	0.986	0.994	0.787	0.99
	0.0001	0.994	0.986	0.994	0.787	0.99
40	0.001	0.991	0.987	0.994	0.805	0.99
	0.0005	0.991	0.987	0.994	0.805	0.99
	0.0001	0.991	0.987	0.994	0.805	0.99
60	0.001	0.995	0.993	0.994	0.824	0.99
	0.0005	0.995	0.993	0.994	0.824	0.99
	0.0001	0.995	0.993	0.994	0.824	0.99
80	0.001	0.995	0.989	0.994	0.817	0.99
	0.0005	0.995	0.989	0.994	0.817	0.99
	0.0001	0.995	0.989	0.994	0.817	0.99
100	0.001	0.992	0.993	0.994	0.82	0.99
	0.0005	0.992	0.993	0.994	0.82	0.99
	0.0001	0.992	0.993	0.994	0.82	0.99

Tabel 4.5 Tabel akurasi training deteksi manusia

Learning Rate	Accuracy
0.01	0.87
0.005	0.86
0.0001	0.87

Tabel 4.6 Tabel akurasi training deteksi objek

Learning Rate	Accuracy
0.001	0.99
0.0005	0.99
0.0001	0.99

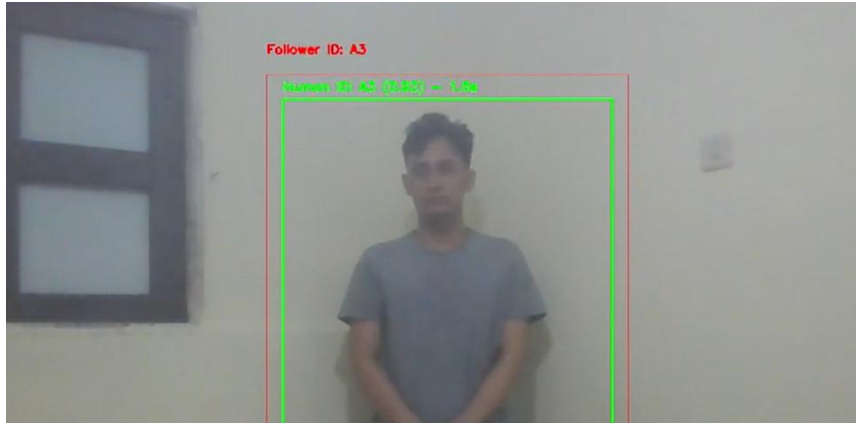
4.2 Pengujian Sistem Dalam Mendeteksi Manusia dan Objek

Pada tahap ini, sistem akan diuji secara *real-time* untuk menilai kemampuannya dalam mendeteksi manusia dan objek yang dibawa. Pengujian akan dilakukan dengan cara berjalan ke area *IP Camera* sampai keluar dari area *IP Camera*, pengujian dilakukan pada siang hari dan malam hari, guna mengevaluasi sejauh mana sistem dapat bekerja secara optimal di lingkungan yang berbeda.

4.2.1 Metode Mendeteksi Manusia dan Objek

Pengujian dilakukan dengan dua metode, yaitu pada siang hari dan malam hari. Cara pengujian sama, yaitu berjalan sambil membawa kantong sampah plastik berwarna putih ke area pengawasan *IP Camera* sampai keluar dari area pengawasan *IP Camera*.

Saat deteksi, sistem akan menghasilkan dua *bounding box* untuk manusia, yaitu *bounding box* berwarna hijau dan merah, seperti ditunjukkan pada Gambar 4.1. *Bounding box* hijau menandakan posisi manusia, sedangkan *bounding box* merah adalah *bounding box* tambahan yang 5% lebih besar dari *bounding box* hijau dan mengikuti pergerakan manusia. *Bounding box* merah akan aktif selama 2 detik sejak manusia terdeteksi dan berfungsi untuk memeriksa apakah manusia membawa objek. Jika *bounding box* merah bertumpang tindih dengan *bounding box* objek, maka manusia tersebut dianggap membawa objek. Selanjutnya, *bounding box* objek yang sebelumnya tidak berwarna akan diberi warna kuning dan diberi *ID* sesuai dengan *ID* manusia tersebut, sementara *bounding box* manusia juga akan berubah warna menjadi kuning, seperti ditunjukkan pada Gambar 4.3, setelah itu, *bounding box* merah akan dihapus. Huruf A pada *ID* ditambahkan untuk membedakan antara *ID* yang digunakan sistem dan *ID* dari ByteTrack.



Gambar 4.1 Dua *Bounding Box* manusia

Untuk metode pengujian siang hari dapat dilihat pada Gambar 4.2 dan pada malam hari pada Gambar 4.3.



Gambar 4.2 Pengujian pada siang hari



Gambar 4.3 Pengujian pada malam hari

4.2.2 Prosedur Pengujian Mendeteksi Manusia dan Objek

Berikut merupakan langkah-langkah dalam menguji model untuk mendeteksi manusia dan objek yang dibawa:

1. Berjalan ke area pengawasan *IP Camera*.
2. Melihat apakah manusia dan objek berhasil di tandai *bounding box*.
3. Mencatat hasil deteksi manusia dan objek.
4. Menghitung nilai akurasi dari manusia dan objek yang berhasil terdeteksi.

4.2.3 Hasil Pengujian Mendeteksi Manusia dan Objek

Hasil pengujian diperoleh menggunakan *IP Camera* yang terhubung dengan laptop yang telah terintegrasi dengan model AI untuk mendeteksi manusia dan objek yang dibawa. Agar hasil pengujian konsisten, pengujian akan langsung dilakukan selama 30 kali, berjalan ke area pengawasan *IP Camera* sampai keluar area *IP Camera*. Jika selama berjalan manusia maupun objek yang dibawa tidak terdeteksi, maka deteksi akan dianggap gagal. Penghitungan akurasi dihitung menggunakan rumus 4.1.

$$Akurasi = \left(\frac{Jumlah\ Terdeteksi}{Banyak\ Data} \right) \times 100\% \quad (4.1)$$

Jumlah terdeteksi dihitung saat manusia dan objek berhasil terdeteksi, sedangkan banyak data adalah total percobaan yang dilakukan selama pengujian. Model AI yang digunakan pada pengujian adalah YOLOv8, untuk mendeteksi manusia, model dilatih sebanyak 50 epoch dengan learning rate sebesar 0.01, dan untuk mendeteksi objek, model telah dilatih sebanyak 100 epoch dengan learning rate 0.0005.

A. Hasil Pengujian Pada Siang Hari

Pengujian ini dilakukan untuk menghitung akurasi sistem dalam mendeteksi manusia dan objek kantong plastik putih yang dibawa pada siang hari. Hasil pengujian menunjukkan jika sistem mampu mendeteksi manusia sekaligus objek yang dibawa dengan sangat baik. Hasil pengujian dapat dilihat pada Tabel 4.7.

Tabel 4.7 Tabel pengujian deteksi pada siang hari

Orang ke-	SIANG HARI			
	Manusia		Objek	
	Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi
1	✓		✓	
2	✓		✓	
3	✓		✓	
4	✓		✓	
5	✓		✓	
6	✓		✓	
7	✓		✓	
8	✓		✓	
9	✓		✓	
10	✓		✓	
11	✓		✓	
12	✓		✓	
13	✓		✓	
14	✓		✓	
15	✓		✓	
16	✓		✓	
17	✓		✓	
18	✓		✓	
19	✓		✓	
20	✓		✓	
21	✓		✓	
22	✓		✓	
23	✓		✓	
24	✓		✓	
25	✓		✓	
26	✓		✓	
27	✓		✓	

SIANG HARI				
Orang ke-	Manusia		Objek	
	Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi
28	✓		✓	
29	✓		✓	
30	✓		✓	
Total	30	0	30	0

Pada Tabel 4.7 menyajikan 30 hasil pengujian model dalam mendeteksi manusia dan objek yang dibawahnya pada siang hari. Penghitungan akurasi dilakukan dengan rumus 4.1 sebagai berikut:

$$\text{Akurasi Manusia Siang Hari} = \left(\frac{30}{30}\right) \times 100\% = 100\%$$

$$\text{Akurasi Objek Siang Hari} = \left(\frac{30}{30}\right) \times 100\% = 100\%$$

Hasil dari pengujian menghasilkan nilai akurasi sebesar 100% untuk deteksi manusia dan 100% untuk deteksi objek.

B. Hasil Pengujian Pada Malam Hari

Pengujian ini dilakukan untuk menghitung akurasi sistem dalam mendeteksi manusia dan objek kantong plastik putih yang dibawa pada malam hari. Hasil pengujian menunjukkan jika sistem mampu mendeteksi manusia sekaligus objek yang dibawa dengan baik. Hasil pengujian dapat dilihat pada Tabel 4.8.

Tabel 4.8 Tabel pengujian deteksi pada malam hari

MALAM HARI				
Orang ke-	Manusia		Objek	
	Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi
1	✓		✓	
2	✓		✓	
3	✓		✓	
4		✓		✓
5	✓		✓	
6	✓		✓	
7		✓		✓
8	✓		✓	
9	✓		✓	
10	✓		✓	
11	✓		✓	
12	✓		✓	
13	✓		✓	

Orang ke-	MALAM HARI			
	Manusia		Objek	
	Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi
14	✓		✓	
15		✓	✓	✓
16		✓	✓	✓
17	✓		✓	
18	✓		✓	
19	✓		✓	
20	✓		✓	
21	✓		✓	
22	✓		✓	
23		✓		✓
24	✓		✓	
25	✓		✓	
26	✓		✓	
27	✓		✓	
28	✓		✓	
29		✓		✓
30	✓		✓	
Total	24	6	24	6

Pada Tabel 4.8 menyajikan 30 hasil pengujian model dalam mendeteksi manusia dan objek yang dibawanya. Penghitungan akurasi dilakukan dengan rumus 4.1 sebagai berikut:

$$\text{Akurasi Manusia Malam Hari} = \left(\frac{24}{30}\right) \times 100\% = 80\%$$

$$\text{Akurasi Objek Malam Hari} = \left(\frac{24}{30}\right) \times 100\% = 80\%$$

Hasil dari pengujian menghasilkan nilai akurasi sebesar 80% untuk deteksi manusia dan 80% objek pada malam hari.

4.3 Pengujian Deteksi Pembuangan Sampah Sembarangan

Pada tahap ini, sistem akan diuji untuk mengevaluasi kemampuannya dalam mendeteksi aktivitas pembuangan sampah sembarangan pada siang hari dan malam hari. Pengujian dilakukan dengan cara melempar sampah secara terkendali sebagai simulasi.

4.3.1 Metode Deteksi Pembuangan Sampah Sembarangan

Pengujian dilakukan dengan dua metode, yaitu pada siang hari dan malam hari. Cara pengujian sama, yaitu berdiri di area pengawasan *IP Camera* dan melempar kantong sampah plastik berwarna putih ke depan. Hasil deteksi siang hari dapat dilihat pada Gambar 4.4 dan pada malam hari pada Gambar 4.5.



Gambar 4.4 Hasil deteksi siang hari



Gambar 4.5 Hasil deteksi malam hari

4.3.2 Prosedur Pengujian Mendeteksi Pembuangan Sampah

Berikut merupakan langkah-langkah dalam menguji model untuk mendeteksi pembuangan sampah sembarangan:

1. Berdiri di area pengawasan *IP Camera*.
2. Melempar sampah kedepan secara terkendali.
3. Mencatat hasil deteksi pembuangan sampah sembarangan.
4. Menghitung nilai akurasi dari hasil deteksi.

4.3.3 Hasil Deteksi Pembuangan Sampah Sembarangan

Hasil pengujian diperoleh menggunakan *IP Camera* yang terhubung dengan laptop yang telah terintegrasi dengan model AI untuk mendeteksi pembuangan sampah sembarangan. Agar hasil pengujian konsisten, pengujian akan langsung dilakukan selama 30 kali, melempar sampah, lalu keluar area pengawasan *IP Camera* sekaligus mengambil sampah, lalu berdiri lagi ke area pengawasan *IP Camera*. Jika selama percobaan pelemparan tidak berhasil tersimpan ke google sheet, maka deteksi akan dianggap gagal. Penghitungan akurasi dihitung menggunakan rumus 4.2.

$$Akurasi = \left(\frac{Jumlah\ Terdeteksi}{Banyak\ Data} \right) \times 100\% \quad 4.2$$

Jumlah terdeteksi dihitung saat deteksi berhasil tersimpan ke google sheet, sedangkan banyak data adalah total percobaan yang dilakukan selama pengujian. Model AI yang digunakan pada pengujian adalah YOLOv8, untuk mendeteksi manusia, model dilatih sebanyak 50 epoch dengan learning rate sebesar 0.01, dan untuk mendeteksi objek, model telah dilatih sebanyak 100 epoch dengan learning rate 0.0005.

A. Pengujian Pada Siang Hari

Pengujian ini dilakukan untuk menghitung akurasi sistem dalam mendeteksi pembuangan sampah sembarangan pada siang hari. Hasil pengujian menunjukkan jika sistem mampu mendeteksi pembuangan sampah sembarangan dengan cukup baik. Hasil pengujian dapat dilihat pada Tabel 4.9.

Tabel 4.9 Tabel pengujian pada siang hari

SIANG HARI		
Orang Buang Sampah ke-	Terdeteksi	Tidak Terdeteksi
1	✓	
2		✓
3	✓	
4		✓
5		✓
6		✓
7		✓
8	✓	
9	✓	

SIANG HARI		
Orang Buang Sampah ke-	Terdeteksi	Tidak Terdeteksi
10	✓	
11		✓
12	✓	
13	✓	
14		✓
15	✓	
16	✓	
17	✓	
18		✓
19	✓	
20	✓	
21	✓	
22	✓	
23	✓	
24	✓	
25	✓	
26	✓	
27	✓	
28	✓	
29	✓	
30	✓	
Total	22	8

Pada Tabel 4.9 menyajikan 30 hasil pengujian model dalam mendeteksi pembuangan sampah sembarangan. Penghitungan akurasi dilakukan dengan rumus 4.2 sebagai berikut

$$\text{Akurasi Siang Hari} = \left(\frac{22}{30} \right) \times 100\% = 73,33\%$$

Hasil dari pengujian menghasilkan nilai akurasi sebesar 73,33% dengan 22 terdeteksi dan 8 tidak terdeteksi.

B. Pengujian Pada Malam Hari

Pengujian ini dilakukan untuk menghitung akurasi sistem dalam mendeteksi pembuangan sampah sembarangan pada Malam hari. Hasil pengujian dapat dilihat pada Tabel 4.10.

Tabel 4.10 Tabel pengujian pada malam hari

MALAM HARI		
Orang Buang Sampah ke-	Terdeteksi	Tidak Terdeteksi
1	✓	
2		✓
3		✓
4	✓	
5	✓	
6	✓	
7	✓	
8		✓
9		✓
10	✓	
11	✓	
12		✓
13		✓
14		✓
15	✓	
16	✓	
17	✓	
18		✓
19	✓	
20	✓	
21	✓	
22	✓	
23	✓	
24	✓	
25	✓	
26		✓
27		✓
28	✓	
29	✓	
30	✓	
Total	20	10

Pada Tabel 4.10 menyajikan 30 hasil pengujian model dalam mendeteksi pembuangan sampah sembarangan pada malam hari. Penghitungan akurasi dilakukan dengan rumus 4.2 sebagai berikut

$$Akurasi Malam Hari = \left(\frac{20}{30}\right) \times 100\% = 66,7\%$$

Hasil dari pengujian menghasilkan nilai akurasi sebesar 66,7% dengan 20 terdeteksi dan 10 tidak terdeteksi.

4.4 Pengujian Kecepatan Pengiriman Data

Pada tahap ini, sistem akan diuji untuk mengukur waktu yang dibutuhkan sejak terdeteksi aktivitas pembuangan sampah wahingga hasil pengambilan gambar berhasil disimpan dan dikirim ke Google Drive dan Google Sheets.

4.4.1 Metode Pengujian Kecepatan Pengiriman Data

Pengujian dilakukan dengan cara menghitung selisih waktu terdeteksi pada jam *IP Camera* dengan Jam saat berhasil terkirim ke google sheet. Contoh waktu pelanggaran terdeteksi dapat dilihat pada Gambar 4.6 dan waktu berhasil tersimpan di google sheet dapat dilihat pada Gambar 4.7.



Gambar 4.6 Waktu pelanggaran terdeteksi

2025-07-10 14:51:04
2025-07-10 14:51:16
2025-07-10 14:51:20
2025-07-10 14:51:49

Gambar 4.7 Waktu berhasil tersimpan ke Google Sheet

4.4.2 Prosedur Pengujian Kecepatan Pengiriman Data

Berikut merupakan langkah-langkah dalam menguji kecepatan pengiriman data:

1. Mencatat jam saat pembuangan sampah sembarangan terdeteksi.
2. Mencatat jam saat deteksi berhasil disimpan di google sheet.
3. Menghitung selisih antara waktu terdeteksi dan waktu tersimpan.
4. Menghitung rata-rata selisih antara waktu deteksi dan waktu tersimpan.

4.4.3 Hasil Pengujian Kecepatan Pengiriman Data

Hasil pengujian diperoleh dengan cara menghitung selisih waktu terdeteksi pada jam *IP Camera* dengan Jam saat berhasil terkirim ke google sheet. Pengambilan data akan dilakukan dengan mengambil dari data deteksi pada siang hari dan malam hari. Penghitungan akurasi dihitung menggunakan rumus 4.3.

$$Rata - rata = \left(\frac{\text{Jumlah Selisih waktu}}{\text{Banyak Data}} \right) \quad 4.3$$

Jumlah selisih waktu merupakan hasil dari penghitungan selisih antara waktu terdeteksi dan waktu tersimpan, sedangkan banyak data adalah total percobaan yang dilakukan selama pengujian. Hasil pengujian dapat dilihat pada tabel 4.11.

Tabel 4.11 Pengujian pengiriman data

Orang Buang Sampah ke-	Waktu Terdeteksi	Waktu Tersimpan	Selisih Waktu (detik)
1	14:50:58	14:51:04	6
2	14:51:14	14:51:20	6
3	14:51:43	14:51:49	6
4	14:52:10	14:52:16	6
5	14:55:20	14:55:26	6
6	14:57:42	14:57:49	7
7	14:58:09	14:58:15	6
8	14:58:29	14:58:36	7
9	14:59:23	14:59:29	6
10	14:59:45	14:59:52	7
11	15:00:44	15:00:54	10
12	15:01:04	15:01:11	7
13	15:01:30	15:01:38	8
14	15:01:52	15:01:59	7
15	15:02:15	15:02:21	6
16	15:02:40	15:02:48	8
17	15:03:04	15:03:10	6
18	15:03:30	15:03:36	6
19	15:03:59	15:04:05	6
20	15:04:22	15:04:28	6

Orang Buang Sampah ke-	Waktu Terdeteksi	Waktu Tersimpan	Selisih Waktu (detik)
21	15:04:43	15:04:49	6
22	15:05:05	15:05:11	6
23	00:35:55	00:36:02	7
24	00:36:22	00:36:28	6
25	00:37:03	00:37:09	6
26	00:37:12	00:37:18	6
27	00:37:20	00:37:26	6
28	00:37:36	00:37:43	7
29	00:38:29	00:38:35	6
30	00:39:21	00:39:27	6
Total			195

Pada Tabel 4.11 menyajikan 30 hasil pengujian pengiriman data, pengujian dilakukan dengan cara menghitung selisih antara waktu terdeteksi dan waktu data telah tersimpan ke Google Drive dan Google Sheet. Penghitungan rata-rata waktu terkirim dilakukan dengan rumus 4.3 sebagai berikut:

$$Rata - rata = \left(\frac{195}{30} \right) = 6,5 \text{ detik}$$

Hasil Menunjukkan jika rata-rata waktu yang dibutuhkan untuk sistem untuk berhasil menyimpan hasil deteksi adalah 6,5 detik.



UNIVERSITAS
Dinamika

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian dan pengujian yang telah dilakukan, terdapat beberapa kesimpulan yang dapat diambil:

1. Sistem telah berhasil mendeteksi aktivitas pembuangan sampah sembarangan dengan menggunakan perbandingan piksel (*Euclidean Distance*).
2. Sistem berbasis *computer vision* telah berhasil dikembangkan menggunakan Visual Studio Code, dan mampu beroperasi secara *real-time*, dengan akurasi mencapai 77,33% pada siang hari dan 66,7% pada malam hari.
3. Sistem telah berhasil mempertahankan *ID* terhadap manusia dan objek yang dibawa pada setiap *frame* menggunakan algoritma *ByteTrack*.
4. Sistem telah berhasil menyimpan dan mengelola data hasil deteksi pembuangan sampah sembarangan pada Google Sheets dan penyimpanan lokal dengan rata-rata waktu pengiriman sebesar 6,5 detik.
5. Sistem telah berhasil dikembangkan untuk menandai manusia dan objek yang dibawanya menggunakan *bounding box*, dengan memanfaatkan *bounding box* merah yang diperbesar 5% dari *bounding box* manusia, sehingga mampu membedakan objek yang dibawa dari objek berserakan di lingkungan dengan akurasi untuk mendeteksi manusia dan objek masing-masing sebesar 100% pada siang hari dan 80% pada malam hari.
6. Hasil pengujian menunjukkan bahwa sistem beroperasi secara optimal pada kondisi lingkungan dengan pencahayaan yang merata, sementara pencahayaan tidak merata, seperti pada malam hari, dapat menurunkan performa deteksi karena berkurangnya kejernihan gambar.
7. Kecepatan internet tinggi berpengaruh signifikan terhadap performa sistem, karena *IP Camera* memerlukan koneksi internet yang stabil dan cepat untuk mengirimkan data video secara *real-time*. Koneksi internet yang lambat dapat menyebabkan video menjadi blur, mengurangi kualitas gambar, dan berdampak pada penurunan akurasi deteksi karena detail objek dan manusia

menjadi kurang jelas. Sehingga penggunaan *IP Camera* untuk pengambilan video menjadi kurang maksimal jika tidak didukung koneksi internet yang cepat dan stabil.

8. *Threading* sangat berperan penting dalam sistem deteksi *real-time* ini, karena memungkinkan proses pembacaan video dan pengunggahan data berjalan secara paralel. Dengan demikian, sistem tidak terhenti saat mengambil frame dari kamera atau saat mengunggah screenshot ke Google Drive dan Google Sheets.

5.2 Saran

Untuk pengembangan lebih lanjut, terdapat beberapa saran yang dapat dipertimbangkan:

1. Menambah jumlah dan variasi dataset untuk mencakup lebih banyak skenario, seperti kondisi pencahayaan, cuaca, dan sudut kamera yang beragam, guna meningkatkan akurasi sistem deteksi.
2. Mengembangkan sistem untuk mendeteksi keberadaan tempat sampah di area pengawasan, sehingga sistem dapat membedakan pembuangan sampah sembarangan dengan pembuangan pada tempatnya.
3. Mengintegrasikan sistem dengan mekanisme notifikasi *real-time*, seperti peringatan melalui email atau aplikasi, untuk memungkinkan respons cepat dari pihak berwenang terhadap pelanggaran.
4. Mengintegrasikan sistem dengan website untuk menyediakan antarmuka pengguna yang lebih rapi, intuitif, dan menarik, memungkinkan visualisasi hasil deteksi, laporan pelanggaran, dan data analisis secara terorganisir, serta memudahkan akses dan pengelolaan data.
5. Meningkatkan performa sistem terhadap kondisi lingkungan yang bervariasi, seperti hujan atau kabut, dengan melatih model pada dataset yang mencakup kondisi cuaca ekstrem.
6. Penelitian ini dapat dikembangkan lebih lanjut dengan menggunakan metode deep learning yang dapat berjalan secara *real-time* lainnya, seperti *Faster-RCNN* dan *SSD (Single Shot Detection)*.

DAFTAR PUSATAKA

- Abouelyazid, M. (2023). Comparative Evaluation of *SORT*, *DeepSORT*, and *ByteTrack* for Multiple Object Tracking in Highway Videos. *Nternational Journal of Sustainable Infrastructure for Cities and Societies*, 8(11), 42–52.
- Afni, S. V. N., Esi Putri Silmina, & Irwanda Budi Pangestu. (2021). *Computer Vision* Used to Monitor The Youth during The Pandemic Covid-19. *Procedia of Engineering and Life Science*, 1(2). <https://doi.org/10.21070/pels.v1i2.1019>
- Aisyah, R. (2023). *Sistem Identifikasi Aktivitas Manusia dalam Pembuangan Sampah Sembarangan Secara Real-Time Menggunakan Vision Sensing Berbasis Data Website*. UPT Perpustakaan POLSRI.
- Basyuni, R. A., & Sugandi, B. (2022). Queue Distancing Based On Image Processing. *Seminar Nasional Terapan Riset Inovatif (SENTRINOV)*, 8(1), 286–293.
- Honnainah, H., & Enggar Pawening, R. (2023). Deteksi Otomatis Terhadap Pelanggaran Pembuang Sampah Menggunakan Metode You Only Look Once (YOLO). *TRILOGI: Jurnal Ilmu Teknologi, Kesehatan, Dan Humaniora*, 4(2), 98–105. <https://doi.org/10.33650/trilogi.v4i2.6676>
- Kristianto, R. P., Putra, H. A., Andrian, D., & Danang, Y. (2025). *Model Estimasi Object Measurements untuk Pengukuran Objek Material Otomatis Menggunakan YOLOv5 dan OpenCV Automated Object Material Measurement Using YOLOv5 and OpenCV: An Object Measurements Dimentional Estimation Model*. 14(105), 166–172.
- Ponto, V. P., Makkulawuzar, K., & Haritsa. (2024). Efektivitas Hukum Penegakan CCTV Sebagai Alat Bukti Dalam Penegakan Hukum Tindak Pidana Pencurian di Kota Gorontalo. *Unisan Law Research*, 1(5), 1–15.
- Verawati, P. (2022). Kebijakan Extended Producer Responsibility Dalam Penanganan Masalah Sampah Di Indonesia Menuju Masyarakat Zero Waste. *JUSTITIA: Jurnal Ilmu Hukum Dan Humaniora*, 9(1), 189–197.
- Yuskar, A. F. (2024). *PENGUNAAN MODUL ESP32 CAM FAEDAH PEMBUANGAN SAMPAH SEMBARANGAN GUNA PENERTIBAN DAN KEPATUHAN MASYARAKAT MENGGUNAKAN ARDUINO MEGA 2560 DAN INTERNET OF THINGS*. Universitas Putra Indonesia YPTK.
- Zhorif, N. N., Anandyto, R. K., Rusyadi, A. U., & Irwansyah, E. (2024). Implementation of Slicing Aided Hyper Inference (SAHI) in YOLOv8 to Counting Oil Palm Trees Using High-Resolution Aerial Imagery Data. *International Journal of Advanced Computer Science and Applications*, 15(7), 869–874. <https://doi.org/10.14569/IJACSA.2024.0150786>