

**SISTEM PERHITUNGAN KENDARAAN MENGGUNAKAN
ALGORITMA YOLOV5 DAN DEEPSORT**

(Skripsi)

Oleh

**PUTRI ANGGIA CAHYANI
NPM 1915061042**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

**SISTEM PERHITUNGAN KENDARAAN MENGGUNAKAN
ALGORITMA YOLOV5 DAN DEEPSORT**

Oleh
PUTRI ANGGIA CAHYANI

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

ABSTRAK

SISTEM PERHITUNGAN KENDARAAN MENGGUNAKAN ALGORITMA YOLOV5 DAN DEEPSORT

Oleh

PUTRI ANGGIA CAHYANI

Pencemaran udara menjadi isu serius di kota-kota besar, seperti di kota Bandar Lampung. Hal ini diakibatkan oleh tingginya aktivitas transportasi menggunakan kendaraan bermotor. Data tahun 2021 menunjukkan peningkatan 4,30% jumlah kendaraan bermotor di Indonesia, berdampak pada emisi karbon. Sebagai respons terhadap permasalahan tersebut, *Greenmetric* Universitas Lampung memiliki program kerja transportasi hijau yang berfokus dalam mengurangi emisi kendaraan bermotor. Untuk mendukung tujuan tersebut dilakukan pemantauan lalu lintas secara otomatis dengan menerapkan bidang *Computer Vision*, yaitu *object tracking*. Pembuatan sistem *object tracking* dalam melakukan pemantauan lalu lintas menggunakan kombinasi algoritma YOLOv5 dan DeepSORT. Metode yang digunakan dalam penelitian ini adalah metode Scrum yang dilakukan sebanyak tiga kali *sprint*, dan dibagi menjadi tiga tahap yaitu *pre-game*, *game*, dan *post-game*. Hasil dari penelitian ini adalah sebuah sistem *object tracking* yang berhasil membedakan dan menghitung tiga jenis kendaraan (motor, mobil, dan bis) secara otomatis, dan telah dilakukan pengujian secara *realtime* dengan nilai rata-rata *precision* sebesar 99%, *recall* sebesar 97%, *F1 score* sebesar 97.2%, *accuracy* sebesar 96.8%, dan rata-rata ketepatan perhitungan sistem sebesar 97.65%.

Kata kunci: emisi karbon, *computer vision*, *object tracking*, yolov5, deepsort.

ABSTRACT

SISTEM PERHITUNGAN KENDARAAN MENGGUNAKAN ALGORITMA YOLOV5 DAN DEEPSORT

By

PUTRI ANGGIA CAHYANI

Air pollution is a serious issue in big cities, such as Bandar Lampung. This is caused by high transportation activities using motorized vehicles. Data from 2021 shows a 4.30% increase in the number of motorized vehicles in Indonesia, impacting carbon emissions. In response to this problem, Greenmetric Lampung University has a green transportation work program that focuses on reducing motor vehicle emissions. To support this goal, automatic traffic monitoring is carried out by applying the field of Computer Vision, namely object tracking. The making of an object tracking system in monitoring traffic uses two combinations of the YOLOv5 and DeepSORT algorithms. The method used in this research is the Scrum method which is carried out in three sprints and divided into three stages, namely pre-game, game, and post-game. The result of this research is an object tracking system that successfully distinguishes and counts three types of vehicles (motorcycle, car, and bus) automatically, and has been tested in real-time with an average precision value of 99%, recall of 97%, F1 score of 97.2%, accuracy of 96.8%, and average accuracy of system calculation of 97.65%.

Keywords: carbon emissions, computer vision, object tracking, yolov5, deepsort.

Judul Skripsi : **SISTEM PERHITUNGAN KENDARAAN
MENGUNAKAN ALGORITMA YOLOV5
DAN DEEPSORT**

Nama Mahasiswa : **Putri Anggia Cahyani**

Nomor Pokok Mahasiswa : **1915061042**

Program Studi : **Teknik Informatika**

Jurusan : **Teknik Elektro**

Fakultas : **Teknik**



1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.
NIP 19720316 199903 2 002

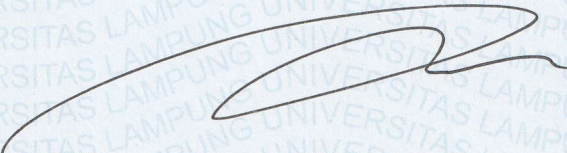
Puput Budi Wintoro, S.Kom., M.T.I.
NIP 19841031 201903 1 004

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

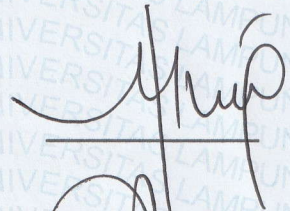

Herlinawati, S.T., M.T.
NIP 19710314 199903 2 001


Mona Arif Muda, S.T., M.T.
NIP 19711112 200003 1 002

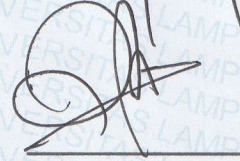
MENGESAHKAN

1. Tim Penguji

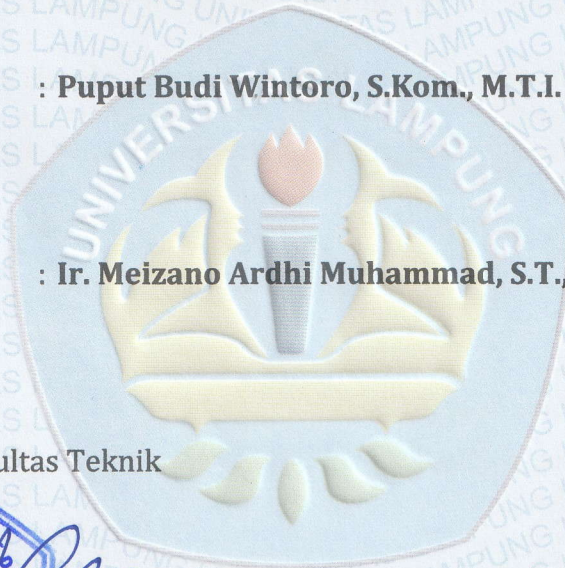
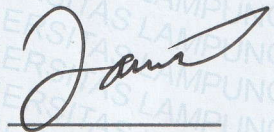
Ketua : Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.



Sekretaris : Puput Budi Wintoro, S.Kom., M.T.I.



Penguji : Ir. Meizano Ardhi Muhammad, S.T., M.T.



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. }

NIP.19750928 200112 1 002


Tanggal Lulus Ujian Skripsi : 10 Agustus 2023

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “Sistem Perhitungan Kendaraan Menggunakan Algoritma YOLOv5 dan DeepSORT” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 01 September 2023

Pembuat pernyaa



Putri Anggia Cahyani

NPM. 1915061042



RIWAYAT HIDUP



Penulis dilahirkan di Tangerang pada tanggal 30 Oktober 2001. Putri pertama dari dua bersaudara, dari (*Alm*). Bapak Makwan Mahuning dan Ibu Riana Sagala. Pendidikan formal yang pernah diselesaikan oleh penulis adalah Sekolah Dasar Negeri Cengkareng Barat 18 pada tahun 2013. Sekolah Menengah Pertama Negeri 248 Jakarta SSN pada tahun 2016. Sekolah Menengah Atas Negeri 84 Jakarta pada tahun 2019. Lalu pada tahun 2019, penulis terdaftar sebagai Mahasiswa Jurusan S1 Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis aktif dalam beberapa kegiatan sebagai berikut:

1. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Pendidikan dan Pengembangan Diri, Divisi MIKAT periode 2019/2020 hingga periode 2020/2021.
2. Komunitas Forum Beasiswa Kartu Jakarta Mahasiswa Unggul (KJMU) sebagai relawan dalam kegiatan Pengabdian Masyarakat setiap tahunnya.
3. Melaksanakan kegiatan Kuliah Kerja Nyata (KKN) di Kel. Kalideres, Kota Jakarta Barat pada Januari – Februari 2022.
4. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil Kelas *AI Mastery* di PT. Orbit Ventura Indonesia pada tahun 2022.
5. Praktek Kerja Lapangan (PKL) di PT. Whitesky Aviation sebagai *Software Developer Intern* pada tahun 2022 dengan membuat sistem *geofencing* pada aplikasi *mobile*.

MOTTO

“Maka sesungguhnya bersama kesulitan ada kemudahan”

(Q.S. AL-INSYIRAH 94:5)

“Your beliefs become your thoughts, your thoughts become your words, your words become your actions, your actions become your habits, your habits become your values, your values become your destiny”

-M. K. Gandhi

“Around here, however, we don't look backwards for very long. We keep moving forward, opening up new doors and doing new things, because we're curious... and curiosity keeps leading us down new paths”

-Walt Disney

PERSEMBAHAN

Puji dan syukur saya ucapkan kepada Allah SWT atas segala nikmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan.

KUPERSEMBAHKAN KARYA KECILKU INI UNTUK:

“Ibu Riana Sagala, ibuku tersayang yang telah melahirkanku, merawatku, membesarkanku, dan yang telah sepenuh hati mendidikku.”

“(Alm). Bapak Makwan Mahuning, ayahku tercinta yang telah membesarkanku dengan seluruh kasih dan sayangnya, serta memberikan pengetahuannya.”

“Adik lelakiku Dewangga Erlang Putra, yang telah menjadi semangat dan alasan supaya aku menjadi sosok kakak panutan baginya.”

“Diriku sendiri, terima kasih telah berjuang dan terus bangkit sampai pada titik ini. Terima kasih telah menjadi sosok yang kuat, maaf atas banyak malam panjang yang dilalui karena memikirkan beban yang sedang ditanggung hingga terkadang menguras air mata.”

Serta, Alamamater yang kubanggakan.

“UNIVERSITAS LAMPUNG”

SANWACANA

Bismillahirrahmanirrahim...

Puji syukur ke hadirat Allah SWT, karena atas segala rahmat, hidayah, serta nikmat-Nya, penulis dapat menyelesaikan skripsi ini tepat pada waktunya. Shalawat serta salam tercurah kepada Nabi Muhammad SAW sebagai tauladan umat manusia di dunia.

Skripsi dengan judul **“SISTEM PERHITUNGAN KENDARAAN MENGGUNAKAN ALGORITMA YOLOV5 DAN DEEPSORT”** disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada program Studi Teknik Informatika Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.

Dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
2. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Bapak Mona Arif Muda, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Jurusan Teknik Elektro Universitas Lampung.
4. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M selaku Pembimbing Utama dan Dosen Pembimbing Akademik, yang telah bersedia meluangkan waktu

untuk memberikan pengarahan dan bimbingan dalam pembuatan skripsi ini hingga selesai.

5. Bapak Puput Budi Wintoro, S.Kom., M.T.I. selaku Pembimbing Kedua yang telah banyak membantu, meluangkan waktu dan memberikan saran serta nasihat dalam mengerjakan skripsi hingga selesai.
6. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T. selaku Penguji Utama yang telah membantu sehingga skripsi ini menjadi lebih baik melalui masukan yang diberikan beliau.
7. Seluruh jajaran staf administrasi atas bantuannya dalam menyelesaikan urusan administrasi di Jurusan Teknik Elektro Universitas Lampung.
8. Kedua Orang tua, yang sangat penulis cintai dan sayangi yang telah memberikan do'a, dukungan, nasihat, semangat, serta pengorbanannya, sehingga penulis mampu menyelesaikan skripsi ini.
9. Kepada sahabat ECT yang telah menemani dari awal maba hingga penulisan skripsi ini selesai. Nadia Miranti, Nilam Cahya, Salma Irena dan Sarah Mustika, terima kasih telah menjadi sahabat yang selalu ada dalam senang maupun duka penulis.
10. Kepada sahabat sekamar, Alfiyah Widiyaningsih yang selalu ada disaat saya membutuhkan serta menjadi penyemangat dalam pengerjaan skripsi.
11. Seorang *partner* yang penulis sayangi.

Akhir kata, penulis menerima kritik dan saran yang membangun demi kemajuan di masa depan. Semoga Allah SWT membalas kebaikan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini.

Bandar Lampung, 01 September 2023

Penulis,

Putri Anggia Cahyani

DAFTAR ISI

	Halaman
DAFTAR ISI.....	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR.....	viii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian.....	3
1.3 Perumusan Masalah.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan Skripsi	4
II. TINJAUAN PUSTAKA	6
2.1 <i>Computer Vision</i>	6
2.2 <i>Object Detection</i>	7
2.3 <i>Object Tracking</i>	8
2.4 <i>Deep Learning</i>	9
2.5 <i>You Only Look Once (YOLO)</i>	10
2.6 YOLOv3.....	12
2.7 YOLOv5.....	12
2.8 DeepSORT	14
2.9 <i>Image Labeling</i>	16
2.10 Roboflow	17
2.11 Google Colab.....	17

2.12	Visual Studio Code.....	18
2.13	Bahasa Pemrograman Python.....	19
2.14	Django	20
2.15	<i>Confusion Matrix</i>	21
2.16	<i>Black-box Testing</i>	22
2.17	<i>Greenmetric</i>	23
2.18	Metode Scrum	23
2.19	Penelitian Terkait	25
III.	METODOLOGI PENELITIAN	30
3.1	Waktu dan Tempat	30
3.2	Alat dan Bahan	31
3.3	Tahapan Penelitian	33
3.3.1	<i>Pregame</i>	34
3.3.2	<i>Game</i>	35
3.3.3	<i>Post-game</i>	38
IV.	HASIL DAN PEMBAHASAN	39
4.1	Hasil.....	39
4.1.1	Pre-game	39
4.1.2	<i>Game</i>	43
4.1.3	Post-game.....	96
V.	KESIMPULAN DAN SARAN	98
5.1	Kesimpulan.....	98
5.2	Saran	99
	DAFTAR PUSTAKA	101

DAFTAR TABEL

Tabel	Halaman
3.1 Jadwal penelitian.....	30
3.2 Alat penelitian.....	31
3.3 Skenario uji pada pengujian sistem.....	37
4.1 Topik yang diajukan selama <i>daily scrum</i> pada <i>sprint I</i>	61
4.2 Topik yang diajukan selama <i>daily scrum</i> pada <i>sprint II</i>	67
4.3 Data <i>confusion matrix</i> hasil <i>training</i>	70
4.4 Data hasil perhitungan <i>confusion matrix</i> hasil <i>training</i>	70
4.5 Data perbandingan hasil perhitungan <i>testing</i> pada hari Senin.....	71
4.6 <i>Confusion matrix testing</i> pada hari Senin.....	72
4.7 Data hasil perhitungan <i>confusion matrix testing</i> hari Senin.....	73
4.8 Data perbandingan hasil perhitungan <i>testing</i> pada hari Selasa.....	74
4.9 <i>Confusion matrix testing</i> pada hari Selasa.....	75
4.10 Data hasil perhitungan <i>confusion matrix testing</i> hari Selasa.....	75
4.11 Data perbandingan hasil perhitungan <i>testing</i> pada hari Rabu.....	76
4.12 <i>Confusion matrix testing</i> pada hari Rabu.....	77
4.13 Data hasil perhitungan <i>confusion matrix testing</i> hari Rabu.....	78
4.14 Data perbandingan hasil perhitungan <i>testing</i> pada hari Kamis.....	78
4.15 <i>Confusion matrix testing</i> pada hari Kamis.....	79
4.16 Data hasil perhitungan <i>confusion matrix testing</i> hari Kamis.....	80
4.17 Data perbandingan hasil perhitungan <i>testing</i> pada hari Jumat.....	80
4.18 <i>Confusion matrix testing</i> pada hari Jumat.....	81
4.19 Data hasil perhitungan <i>confusion matrix testing</i> hari Jumat.....	82
4.20 Data hasil perhitungan rata-rata <i>confusion matrix testing realtime</i>	82
4.21 Data persentase kesalahan perhitungan sistem pada seluruh kelas.....	82

4.22 Topik yang diajukan selama daily scrum pada sprint III	91
4.23 Hasil skenario uji <i>black-box testing</i>	93

DAFTAR GAMBAR

Gambar	Halaman
2.1 Susunan waktu topik penelitian <i>computer vision</i>	7
2.2 Proses pengaplikasian CNN dalam <i>object detection</i>	7
2.3 Ilustrasi pemberian ID pada suatu objek.....	8
2.4 Ilustrasi masalah <i>occlusion</i> suatu objek menutupi objek lain	9
2.5 Cara kerja <i>Deep Learning</i>	10
2.6 Proses pendeteksian dalam YOLO.....	10
2.7 Lapisan pada Algoritma YOLO	11
2.8 Perbandingan performa YOLOv3 dengan algoritma lain	12
2.9 Perbandingan performa kerja pada tiap model YOLOv5	13
2.10 Performa YOLOv5 dengan YOLO versi terbaru	14
2.11 Algoritma SORT	15
2.12 Algoritma Deep SORT.....	15
2.13 Proses pelabelan pada citra menggunakan Label Studio	16
2.14 Tampilan Google Colab	18
2.15 <i>Confusion Matrix</i>	21
2.16 Kerangka kerja Scrum.....	23
3.1 Diagram alur penelitian menggunakan metode Scrum	33
4.1 <i>User's Story</i>	40
4.2 Daftar fitur sistem	41
4.3 <i>Product backlog</i>	42
4.4 Pemilihan <i>item product backlog</i> pada sprint I.....	43
4.5 <i>Sprint I: Sprint Backlog</i>	44
4.6 Pengumpulan <i>dataset</i> kendaraan.....	45
4.7 <i>Labeling</i> menggunakan Label Studio.....	46

4.8 Folder hasil anotasi menggunakan Label Studio	46
4.9 Proses pengunggahan hasil anotasi pada Roboflow.....	47
4.10 Pembagian <i>dataset</i> pada Roboflow	48
4.11 <i>Pre-processing</i> pada Roboflow.....	48
4.12 <i>Augmentation</i> pada Roboflow.....	49
4.13 <i>Training dataset</i> menggunakan model yolov5s.....	50
4.14 <i>Summary training</i> menggunakan YOLOv5s.....	51
4.15 Model arsitektur YOLOv5s.....	52
4.16 Proses <i>training</i> sebanyak 1000 <i>epochs</i>	56
4.17 Grafik kerugian hasil <i>training</i>	57
4.18 Grafik <i>metrics</i> hasil <i>training</i>	59
4.19 <i>Confusion matrix</i> hasil <i>training</i>	60
4.20 <i>Progress</i> pekerjaan pada <i>sprint I</i>	63
4.21 Pemilihan <i>item product backlog</i> pada <i>sprint II</i>	64
4.22 <i>Sprint II: Sprint Backlog</i>	65
4.23 Penambahan algoritma Deep SORT pada track.py.....	65
4.24 Penambahan <i>weight</i> YOLOv5 dengan algoritma Deep SORT.....	66
4.25 Algoritma perhitungan pada sistem	66
4.26 <i>Progress</i> pekerjaan pada <i>sprint II</i>	69
4.27 Hasil perhitungan nilai TP, TN, FP, dan FN <i>training</i> YOLOv5.....	70
4.28 Hasil <i>testing</i> sistem secara <i>realtime</i> pada hari Senin	71
4.29 Hambatan <i>overlapping object</i> saat pendeteksian.....	72
4.30 Hasil perhitungan nilai TP, TN, FP, dan FN <i>testing</i> hari Senin.....	73
4.31 Hasil <i>testing</i> sistem secara <i>realtime</i> pada hari Selasa	73
4.32 Objek mobil yang terdeteksi sebagai bis.....	74
4.33 Hasil perhitungan nilai TP, TN, FP, dan FN <i>testing</i> hari Selasa.....	75
4.34 Hasil <i>testing</i> sistem secara <i>realtime</i> pada hari Rabu.....	76
4.35 <i>Sample</i> hasil <i>testing</i> pada hari Rabu.....	77
4.36 Hasil perhitungan nilai TP, TN, FP, dan FN <i>testing</i> hari Rabu	77
4.37 <i>Sample</i> hasil <i>testing</i> pada hari Kamis.....	79
4.38 Hasil perhitungan nilai TP, TN, FP, dan FN <i>testing</i> hari Kamis	79
4.39 <i>Sample</i> hasil <i>testing</i> pada hari Jumat	81

4.40 Hasil perhitungan nilai TP, TN, FP, dan FN <i>testing</i> hari Jumat	81
4.41 <i>Progress</i> pekerjaan <i>sprint II</i> yang telah selesai.....	83
4.42 Pemilihan <i>item product backlog</i> pada <i>sprint III</i>	84
4.43 <i>Sprint III: Sprint Backlog</i>	85
4.44 Struktur tabel kendaraan pada <i>database data_kendaraan</i>	86
4.45 <i>Source code models.py</i>	86
4.46 <i>Source code connector MySQL</i>	87
4.47 <i>Source code insert data sistem tracking</i>	87
4.48 <i>Source code request render</i> pada file <i>views.py</i>	88
4.49 Tampilan halaman <i>report</i>	89
4.50 Fitur cetak halaman <i>report</i>	90
4.51 <i>Progress</i> pekerjaan pada <i>sprint III</i>	92
4.52 <i>Progress</i> pekerjaan <i>sprint III</i> yang telah selesai	95
4.53 <i>Progress</i> tiga <i>sprint</i> pada <i>board Trello</i>	97

I. PENDAHULUAN

1.1 Latar Belakang

Pencemaran udara menjadi permasalahan yang seringkali dijumpai di kota-kota besar. Hal ini disebabkan oleh tingginya intensitas aktivitas transportasi yang melibatkan penggunaan kendaraan individu maupun sarana transportasi publik. Berdasarkan data yang bersumber dari situs *web* dataindonesia.id pada tahun 2021, jumlah kendaraan bermotor di Indonesia meningkat sebesar 4,30% dari tahun sebelumnya. Meningkatnya jumlah kendaraan bermotor dapat berdampak langsung pada pembuangan emisi karbon ke lingkungan. Kota Bandar Lampung sendiri tergolong sebagai salah satu kota besar yang mengalami peningkatan pencemaran udara yang signifikan. Sebagai respons terhadap permasalahan ini, maka dibentuk *greenmetric* oleh Universitas Indonesia yaitu sebuah indeks yang digunakan untuk menilai kinerja berkelanjutan dalam mengurangi dampak negatif terhadap lingkungan, termasuk mengurangi emisi karbon dan pencemaran udara.

Greenmetric Universitas Lampung merupakan salah satu tim di Universitas Lampung yang mendorong peran serta seluruh *civitas* untuk melakukan kolaborasi dan kontribusi dalam pembangunan komunitas kampus yang berkelanjutan sehingga terbentuk kampus hijau yang berfokus pada kelestarian lingkungan hidup. Salah satu program kerja *greenmetric* adalah transportasi hijau, yaitu program kerja yang berfokus pada pengurangan emisi kendaraan bermotor. Untuk menunjang hal tersebut, perlu dilakukan pemantauan kepadatan lalu lintas di jalan kampus Universitas Lampung. Pemantauan kepadatan lalu lintas dapat dilakukan secara otomatis dengan memanfaatkan pengaplikasian bidang studi *Computer Vision* yaitu *object tracking*.

Object tracking melakukan pelacakan lintasan gerak suatu objek dalam video secara tepat.[1][2] Kombinasi algoritma DeepSORT dan YOLOv5 menjadi salah satu kombinasi algoritma yang dapat digunakan dalam pembuatan sistem *object tracking*. Pemilihan kedua algoritma ini berdasarkan studi literatur yang telah dilakukan. DeepSORT merupakan algoritma lanjutan dari SORT yang dibuat untuk menutupi kekurangan pada SORT dengan mengurangi permasalahan tertukarnya ID sebesar 45%.[3] Sedangkan algoritma *object detection* yang terkenal ialah YOLO. YOLO memiliki banyak versi, dengan versi terbaru YOLO saat ini ialah YOLOv8 dengan tingkat akurasi yang menyaingi YOLOv5. Meskipun begitu, performa YOLOv5 dalam pendeteksian sistem *realtime* sangat stabil dan lebih cepat dibandingkan versi YOLO lainnya.

Dengan memanfaatkan kedua algoritma tersebut, dibangun sebuah sistem *object tracking* yang dapat membantu dalam pemantauan awal kepadatan jalan kampus Universitas Lampung terkait permasalahan yang dihadapi. Cara kerja sistem ialah menghitung tiga jenis kendaraan penyumbang emisi karbon (mobil, motor, dan bus) yang paling sering melintasi jalan kampus Universitas Lampung. Data perhitungan yang didapatkan dari sistem disimpan dalam sebuah *database*, dan ditampilkan melalui halaman data kendaraan yang disajikan dalam bentuk tabel. Data perhitungan tersebut dapat membantu tim *greenmetric* Universitas Lampung dalam pemantauan jumlah kendaraan yang melintasi jalan kampus Universitas Lampung. Selain itu, tujuan lain pembuatan sistem ini ialah sebagai langkah awal dalam pengembangan sistem yang memiliki fungsi untuk menghitung pembuangan emisi karbon secara otomatis.

1.2 Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Membuat sebuah sistem *tracking* dan *counting* tiga jenis kendaraan (mobil, motor, dan bus) secara akurat dengan menggunakan algoritma DeepSORT dan YOLOv5.
2. Melakukan pembuatan halaman *website* yang digunakan untuk menampilkan data hasil deteksi kendaraan menggunakan *framework* Django.

1.3 Perumusan Masalah

Berdasarkan latar belakang, kajian masalah yang mendasari penelitian ini adalah:

1. Bagaimana sistem *object tracking* dan *counting* dapat melacak tiga jenis kendaraan (mobil, motor, dan bus) menggunakan algoritma DeepSORT dan YOLOv5 untuk membantu dalam penyelesaian permasalahan lalu lintas?
2. Bagaimana halaman *website* dapat menampilkan data hasil deteksi kendaraan menggunakan *framework* Django?

1.4 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

1. Manfaat aplikatif, penelitian ini diharapkan dapat membantu *stakeholders* terkait dalam pemantauan jumlah kendaraan yang melintasi jalan kampus Universitas Lampung melalui data jumlah kendaraan yang dihasilkan oleh sistem.
2. Manfaat keilmuan, penelitian ini diharapkan dapat menjadi referensi bagi penelitian selanjutnya dan bisa dikembangkan menjadi lebih sempurna.

3. Manfaat institusi, penelitian ini diharapkan dapat memperluas wawasan dan minat pada bidang *Computer Vision* bagi mahasiswa.

1.5 Batasan Masalah

Dalam penelitian ini, pembatasan masalah meliputi hal-hal sebagai berikut:

1. Penambahan variasi *dataset* menggunakan proses *augmentation* pada Roboflow dengan lisensi *free* hanya dapat dilakukan sebanyak 3 kali dari jumlah *dataset training* yang dimiliki.
2. Pengujian sistem dilakukan di luar ruangan, sehingga objek dapat muncul dalam berbagai skala dan jarak yang bervariasi selama proses pengujian.

1.6 Sistematika Penulisan Skripsi

Sistematika penulisan skripsi ini terdiri dari 5 (lima) bab sebagai berikut:

I. PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, tujuan penelitian, perumusan masalah, manfaat penelitian, batasan masalah, dan sistematika penulisan.

II. TINJAUAN PUSTAKA

Bab ini memuat tinjauan pustaka mengenai:

1. *Computer Vision*
2. *Object Detection*
3. *Object Tracking*
4. *Deep Learning*
5. *You Only Look Once (YOLO)*
6. *DeepSORT*
7. *Image Labeling*

8. Roboflow
9. Google Colab
10. Visual Studio Code
11. Bahasa pemrograman Python
(opencv-python, Pillow, PyTorch, tb-nightly, imageio)
12. Django
13. *Confusion Matrix*
14. *Black-box Testing*
15. Metode Scrum
16. Penelitian terkait

III. METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai waktu dan tempat pelaksanaan penelitian, alat dan bahan, serta tahapan penelitian (*Pre-game*, *Game*, dan *Post-game*) yang dilakukan.

IV. HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil penelitian berdasarkan tahapan penelitian (*Pre-game*, *Game*, dan *Post-game*) yang dilakukan, serta pembahasan secara lengkap mengenai *progress* pada Trello.

V. KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan berdasarkan hasil penelitian yang dilakukan dan saran untuk pengembangan dalam penelitian kedepannya.

DAFTAR PUSTAKA

Bagian ini berisi daftar literatur yang digunakan untuk penelitian.

LAMPIRAN

Bagian ini berisi lampiran-lampiran yang menunjang laporan.

II. TINJAUAN PUSTAKA

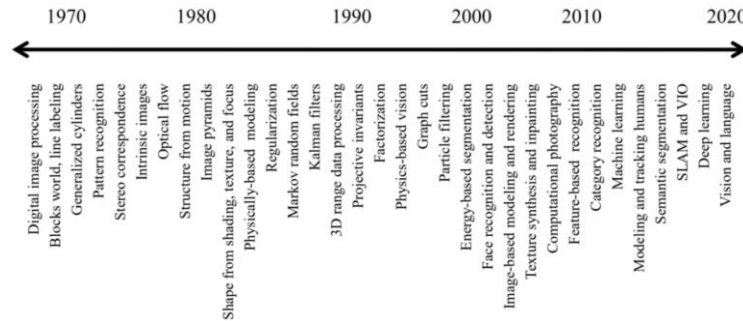
2.1 *Computer Vision*

Computer Vision atau Visi komputer adalah bidang studi yang berfokus pada masalah dalam membantu komputer untuk mereproduksi kemampuan penglihatan seperti layaknya manusia. Secara luas, *computer vision* merupakan sub bidang dari *Artificial Intelligence (AI)* dan *Machine Learning (ML)*, yang bertujuan untuk memahami konten gambar digital.[4] Salah satu elemen penting dalam *computer vision* adalah citra. Citra adalah gambaran visual yang merepresentasikan pantulan cahaya dalam suatu objek. Citra yang diciptakan oleh sebuah mesin digital disebut sebagai citra digital.[5] Dalam pemahaman konten gambar digital, *computer vision* dapat melibatkan penggalian deskripsi dari citra, berupa objek, deskripsi teks, model tiga dimensi, dan sebagainya.

Era *Computer Vision* dimulai pada awal tahun 1970 yang dipandang sebagai komponen persepsi visual untuk meniru kecerdasan manusia dan memberikan robot kecerdasan tingkah laku seperti manusia. Pada saat itu, beberapa pelopor *artificial Intelligence* dan *robotics* mempercayai bahwa dengan menyelesaikan permasalahan masukan pengelihatian atau '*visual input*', akan memudahkan langkah dalam menyelesaikan permasalahan tingkat tinggi lainnya, seperti '*reasoning and planning*' dalam *artificial intelligence*.[6]

Hal tersebut didasari kisah di tahun 1966, Marvin Minsky di MIT (Massachusetts Institute of Technology) meminta mahasiswanya, Gerald Jay Sussman untuk menghabiskan libur musim panas dengan menghubungkan kamera pada computer, dan membuat computer tersebut dapat mendeskripsikan apa yang dilihat. Akan tetapi, Sussman tidak dapat menyelesaikannya, dan dapat disimpulkan bahwa

pengelihatan atau *vision* merupakan salah satu tantangan terbesar dalam *artificial Intelligence*. [6]

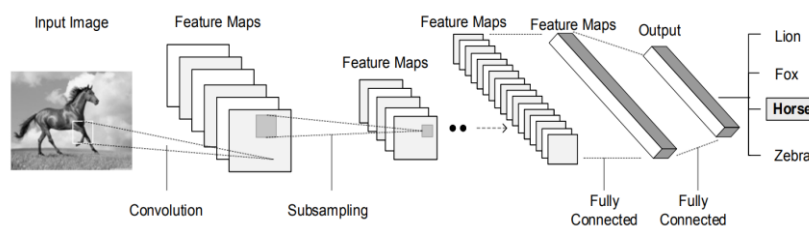


Gambar 2.1 Susunan waktu topik penelitian *computer vision*

2.2 Object Detection

Object Detection adalah prosedur yang dilakukan dalam bidang studi *computer vision* untuk menentukan kelas tiap objek dan mengestimasi lokasi dari objek dengan memberikan *bounding box* di sekitar objek. Pendeteksian sebuah kelas pada suatu citra dinamakan *single class object detection*, sedangkan pendeteksian kelas-kelas pada seluruh objek yang terdapat dalam suatu citra dikenal dengan sebutan *multi class object detection*. [7]

D-CNN atau *Deep Convolutional Neural Network* telah banyak digunakan dalam deteksi objek. [8] CNN adalah sebuah tipe dari jaringan syaraf tiruan *feed-forward* dan bekerja menggunakan prinsip *weight sharing* yaitu membagi bobot sama rata pada setiap filter. [7]



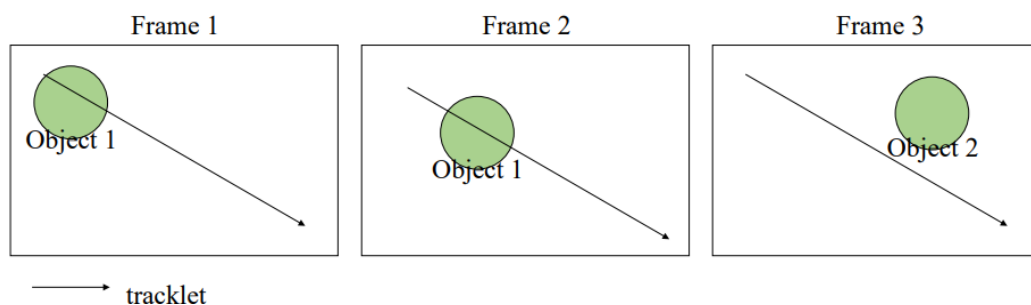
Gambar 2.2 Proses pengaplikasian CNN dalam *object detection*

Citra dilakukan konvolusi atau penggabungan dengan fungsi aktivasi untuk mendapatkan *feature maps*. *Feature maps* akan diproses menggunakan *pooling layer* yaitu lapisan penyatuan untuk menghasilkan *feature maps* yang terabstraksi dan bertujuan untuk mengurangi kompleksitas spasial citra. Proses tersebut terus diulang sebanyak jumlah filter dan *feature maps* yang dibuat.[9] Akhirnya, *feature maps* diproses dengan lapisan yang terhubung sepenuhnya untuk mendapatkan hasil berupa pengenalan citra yang menunjukkan tingkat skor kepercayaan (*confidence score*) untuk kelas yang terprediksi dari citra tersebut.[7]

2.3 Object Tracking

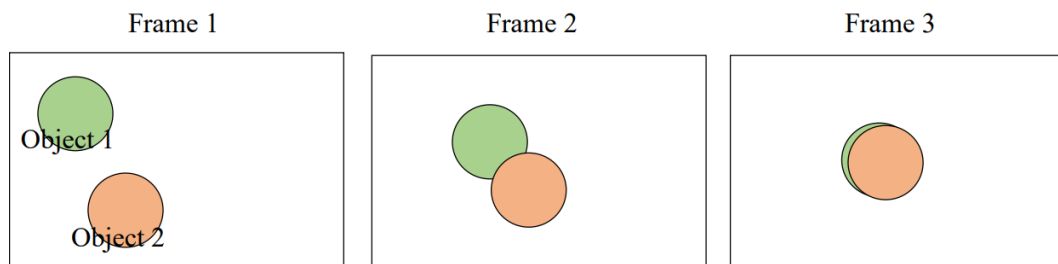
Object tracking atau pelacakan objek adalah salah satu pengaplikasian bidang studi visi komputer mendasar yang mengacu pada serangkaian metode untuk melacak lintasan gerak suatu objek dalam video secara tepat.[10] *Object tracking* dapat diaplikasikan untuk pengawasan lalu lintas, pengenalan tingkah laku dan sebagainya.

Multiple Object Tracking (MOT) merupakan sub bagian dari metode *object tracking*, yang memiliki tujuan untuk melakukan pelacakan banyak objek dalam satu video dan merepresentasikannya menjadi sekumpulan lintasan berakurasi tinggi.[10] Permasalahan dari MOT adalah ketika objek yang sama tidak diberikan ID yang sama dalam seluruh *frame*. Permasalahan tersebut disebabkan karena pertukaran ID dan hambatan.[11]



Gambar 2.3 Ilustrasi pemberian ID pada suatu objek

Pertukaran ID merupakan suatu kondisi ketika objek 1 yang memiliki ID A, diberikan ID berbeda yaitu ID B. Kondisi tersebut disebabkan oleh banyak skenario seperti, ketika pelacak memberikan objek 2 dengan ID A karena objek 2 menyerupai objek 1. Pada Gambar 2.3, objek di *frame* 1 dan objek di *frame* 2 dianggap memiliki ID yang sama karena masih berada di *tracklet* (lintasan). Di sisi lain, *frame* 3 dianggap sebagai objek yang berbeda karena objek keluar dari *tracklet*, dan dengan begitu objek tersebut diberikan ID baru.[11]



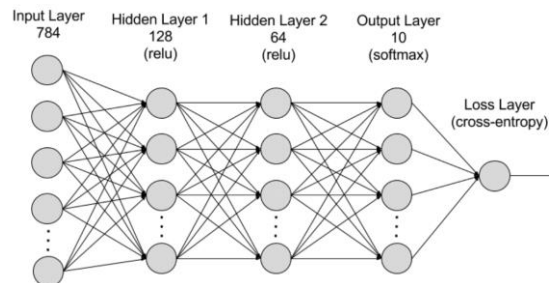
Gambar 2.4 Ilustrasi masalah *occlusion* suatu objek menutupi objek lain

Masalah selain pertukaran ID adalah *occlusion* yaitu ketika objek lain menutupi suatu objek sebagian maupun seluruhnya selama durasi tertentu[11], seperti ilustrasi pada Gambar 2.4.

2.4 Deep Learning

Deep learning adalah teknik dalam *artificial intelligence* (AI) yang mengajarkan komputer atau mesin untuk mengolah data seperti cara kerja otak manusia. Model *deep learning* dapat mengidentifikasi pola yang rumit dalam sebuah citra, teks, suara dan data lain untuk menghasilkan pengetahuan dan prediksi secara akurat.[12] Teknik *deep learning* dapat digunakan untuk mengotomatiskan pekerjaan yang memerlukan kecerdasan manusia, seperti pendeskripsian citra atau kendaraan autonomous. Cara kerja *deep learning* menirukan cara kerja otak manusia seperti jaringan neural. Pada *deep learning*, terdapat jaringan neural yang tercipta dari

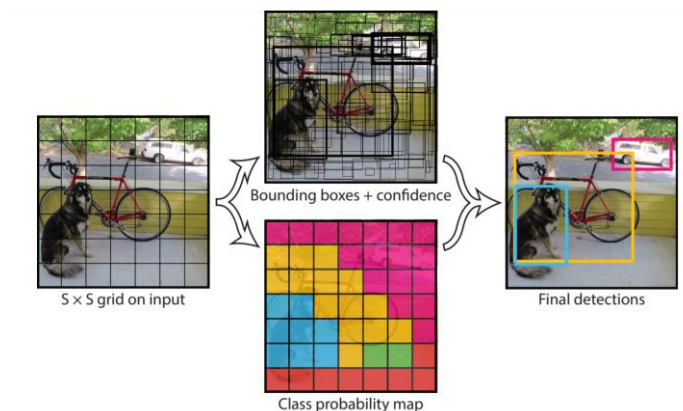
banyak lapisan neuron buatan yang saling bekerja sama di dalam komputer. Jaringan neural ini merupakan algoritma dari *deep learning* yang memanfaatkan neuron dalam memecahkan permasalahan kompleks.[12]



Gambar 2.5 Cara kerja *Deep Learning*

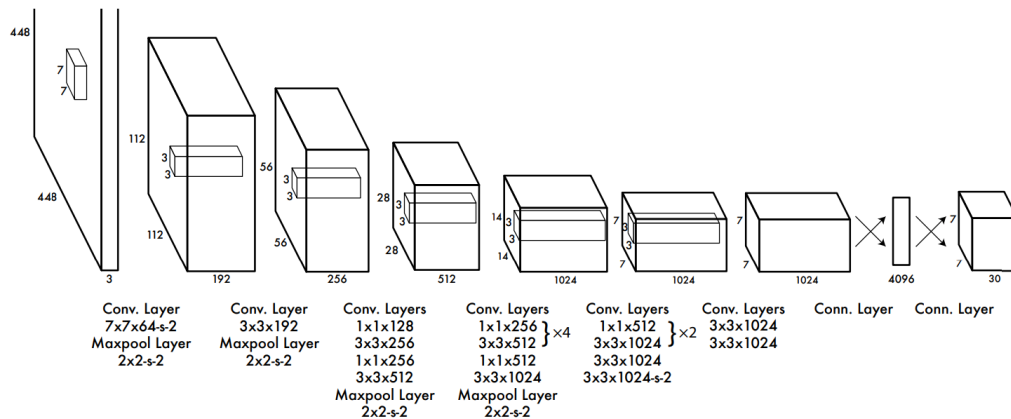
2.5 *You Only Look Once (YOLO)*

You Only Look Once atau YOLO merupakan model pendeteksian banyak objek (*multiple object detection*) dan segmentasi citra yang dikembangkan oleh Joseph Redmon dan Ali Farhadi.[13] Perilisan pertama YOLO pada tahun 2015 banyak menarik perhatian karena model ini dapat melakukan deteksi dengan tingkat kecepatan dan akurasi yang tinggi. Dengan tingkat akurasi yang tinggi, YOLO disebut sebagai salah satu algoritma pendeteksian objek berbasis *deep learning* dengan performa luar biasa.



Gambar 2.6 Proses pendeteksian dalam YOLO

YOLO mendeteksi objek pada citra dengan melakukan lokalisasi, kemudian melakukan pengkomputasian lokasi objek pada citra, dan melakukan klasifikasi untuk mengenali objek yang terdapat dalam citra. Algoritma YOLO dapat digunakan secara realtime melalui tahapan pemrosesan.[14]



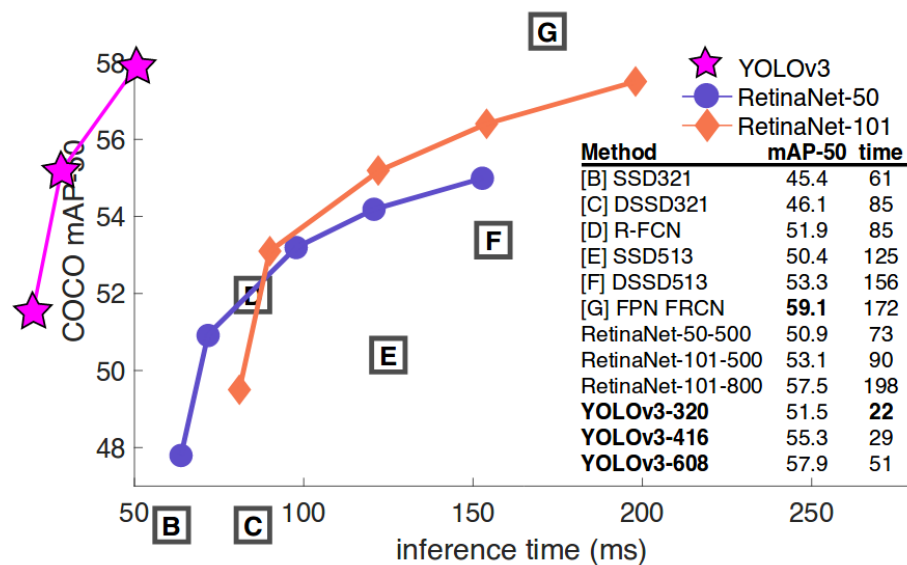
Gambar 2.7 Lapisan pada Algoritma YOLO

Algoritma YOLO mengambil citra sebagai masukan, kemudian mendeteksi objek pada citra menggunakan model *Deep Convolutional Neural Network* atau D-CNN dalam pembentukan *backbone* dari YOLO pada Gambar 2.6. Arsitektur jaringan YOLO terinspirasi oleh model klasifikasi citra GoogLeNet.[15] Pada arsitektur jaringan YOLO memiliki 24 lapisan *convolutional*, dilanjutkan dengan 2 lapisan yang terhubung sepenuhnya.

Dari modul awal GoogLeNet, YOLO hanya menggunakan lapisan reduksi 1 x 1 yang dilanjutkan dengan lapisan konvolusional 3 x 3. Terdapat versi lebih cepat dari YOLO bernama Fast YOLO yang dirancang untuk melakukan deteksi objek secara cepat menggunakan jaringan neural dengan lapisan konvolusional lebih sedikit dibanding YOLO versi biasa. Jika pada YOLO versi biasa menggunakan 24 lapisan, pada YOLO versi cepat menggunakan 9 lapisan konvolusional.[15]

2.6 YOLOv3

YOLOv3 merupakan algoritma deteksi banyak objek (*multiple object detection*) yang dikembangkan oleh Joseph Redmon. Pada YOLOv3, digunakan Darknet53 sebagai *backbone* untuk melakukan ekstraksi fitur dari sebuah masukan citra digital. Pada bagian *neck*, YOLOv3 menggunakan *Feature Pyramid Network* (FPN) yang berperan mengekstraksi *feature maps* dari tahapan yang berbeda. Terakhir, pada bagian *head* terdapat lapisan YOLO yang akan menghasilkan keluaran hasil dari pendeteksian.[13]



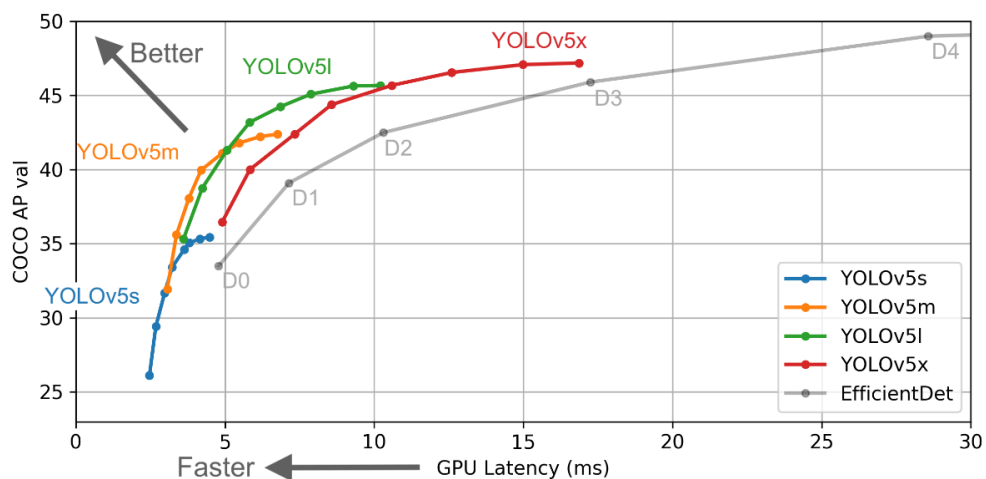
Gambar 2.8 Perbandingan performa YOLOv3 dengan algoritma lain

Berdasarkan Gambar 2.8, YOLOv3 memiliki nilai mAP_0.5 yang tinggi dengan tingkat kecepatan deteksi (waktu inferensi) yang terendah yaitu sekitar 51 ms hingga 22 ms dibandingkan dengan algoritma objek deteksi lainnya.

2.7 YOLOv5

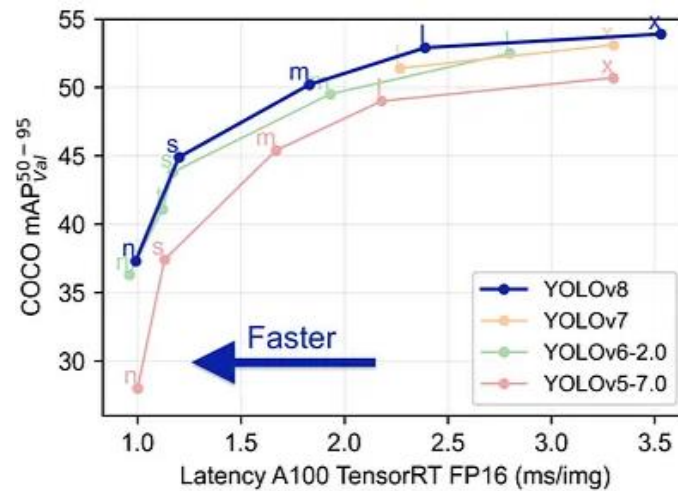
YOLOv5 merupakan algoritma deteksi banyak objek (*multiple object detection*) yang dikembangkan oleh Ultralytics. Secara keseluruhan, arsitektur YOLOv5

terdiri dari 3 bagian utama yaitu *backbone*, *neck*, dan *head*. Pada YOLOv5 dilakukan berbagai teknik augmentasi data untuk meningkatkan kemampuan model dalam melakukan generalisasi dan mengurangi *overfitting*. Teknik-teknik augmentasi tersebut terdiri dari *mosaic augmentation*, *copy-paste augmentation*, *random affine transformations*, *mix-up augmentation*, *albumentations*, *HVS augmentation*, dan *random horizontal flip*. Selain itu, diterapkan juga beberapa strategi *training* yang dapat meningkatkan performa model dalam melakukan deteksi seperti, *multiscale training*, *autoanchor*, *warmup and cosine LR scheduler*, *exponential moving average*, *mixed precision training*, dan *hyperparameter evolution*. Pada *training* menggunakan YOLOv5, nilai *loss* atau kerugian dikomputasikan menggunakan tiga kombinasi komponen yang terdiri dari *classes loss*, *objectness loss*, dan *location loss*.



Gambar 2.9 Perbandingan performa kerja pada tiap model YOLOv5

Dengan menggabungkan berbagai fitur baru, peningkatan, dan strategi pelatihan, YOLOv5 melampaui versi sebelumnya dari keluarga YOLO dalam hal performa dan efisiensi. [16] Berdasarkan Gambar 2.9, waktu inferensi yang dibutuhkan oleh YOLOv5 dalam melakukan deteksi berkisar pada rentang waktu 5 ms hingga 20 ms.

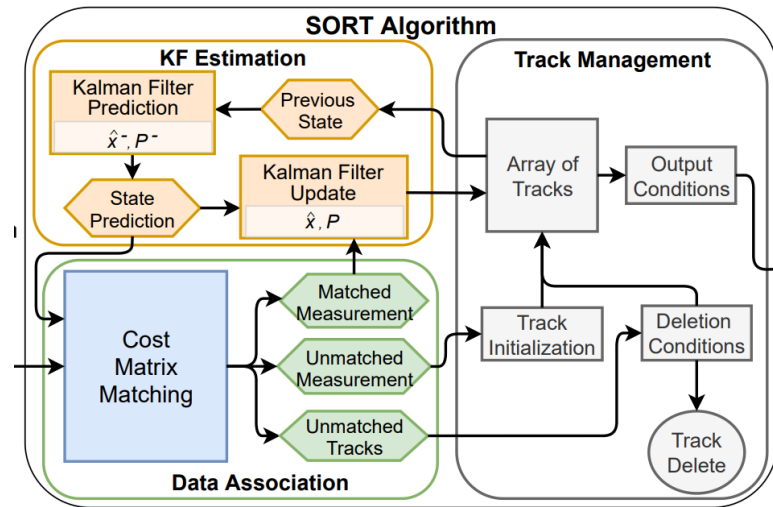


Gambar 2.10 Performa YOLOv5 dengan YOLO versi terbaru

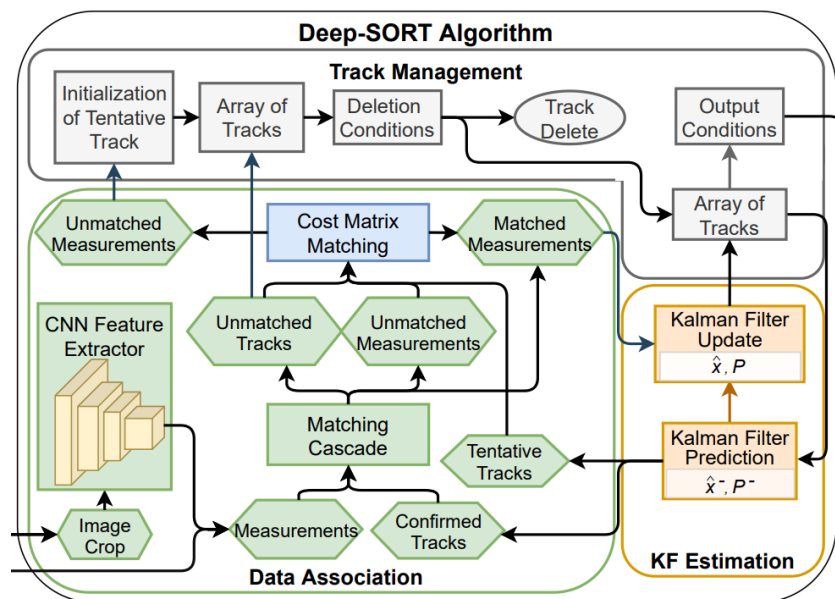
Gambar 2.10 menunjukkan grafik perbandingan waktu inferensi yang dibutuhkan dan akurasi mAP_{0.5:95} dalam melakukan deteksi menggunakan algoritma YOLOv5, YOLOv6, YOLOv7, dan YOLOv8. Berdasarkan grafik, YOLOv8 unggul dalam nilai akurasi dibandingkan YOLO versi lainnya. Akan tetapi, YOLOv5 lebih cepat dalam kecepatan deteksi, karena membutuhkan waktu inferensi yang lebih rendah dibandingkan YOLO versi lainnya dengan rentang waktu 1 ms hingga 3.25 ms.

2.8 DeepSORT

SORT (Simple Online and Realtime Tracking) adalah pendekatan efisien untuk melakukan *tracking* atau pelacakan pada banyak objek dan berfokus pada penggunaan algoritma yang sederhana serta efektif.[17] Pada kinerja SORT, dilakukan pemfilteran Kalman dalam ruang citra dan asosiasi data bingkai demi bingkai menggunakan metode Hungarian dengan metrik asosiasi untuk mengukur *bounding box* yang saling tumpang tindih. Algoritma SORT memiliki kinerja yang baik pada *frame rate* yang tinggi. Akan tetapi, SORT hanya akurat jika tingkat ketidakpastian estimasi objek rendah, karena SORT mengabaikan tampilan fitur objek saat melakukan deteksi.[17]



Gambar 2.11 Algoritma SORT



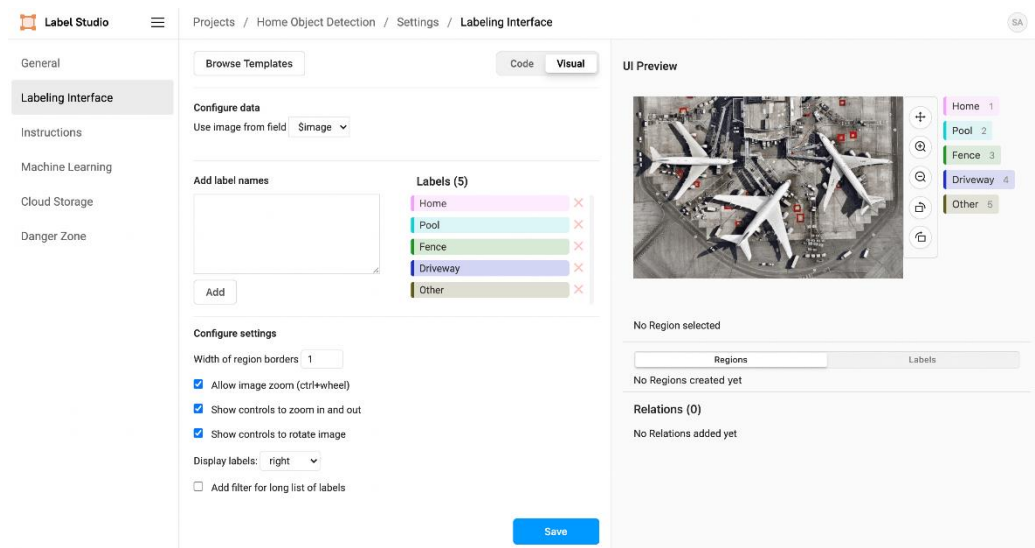
Gambar 2.12 Algoritma Deep SORT

Deep SORT pada dasarnya sama dengan SORT, karena Deep SORT merupakan ekstensi dari algoritma SORT. Hal yang membedakan antara Deep SORT dan SORT ialah pada Deep SORT ditambahkan model CNN untuk ekstraksi fitur pada citra yang dibatasi oleh detektor. Untuk menutupi kekurangan algoritma SORT,

algoritma Deep SORT menambahkan informasi tampilan fitur objek dan melakukan peminjaman model ReID untuk mengekstrak fitur tampilan, serta mengurangi permasalahan tertukarnya ID sebesar 45%. [10]

2.9 Image Labeling

Image labeling adalah suatu teknik untuk melakukan anotasi suatu objek tertentu atau fitur dalam sebuah citra. Dengan melakukan *image labeling*, suatu model *computer vision* dapat mengidentifikasi suatu objek dalam citra. Sebagai contoh, dalam suatu citra, dilakukan anotasi semua objek sebagai mobil, maka model dapat mengerti objek mana saja yang merupakan mobil. [18]



Gambar 2.13 Proses pelabelan pada citra menggunakan Label Studio

Proses *labeling* dapat dilakukan dengan berbagai macam alat anotasi. Menggunakan alat anotasi tersebut, pengguna dapat membuat Batasan tertentu di sekitar objek. Batasan yang dibuat ini merupakan *bounding boxes* atau kotak pembatas. Setiap kotak pembatas diberikan label supaya model dapat membedakan tiap objek yang berbeda. [18]

2.10 Roboflow

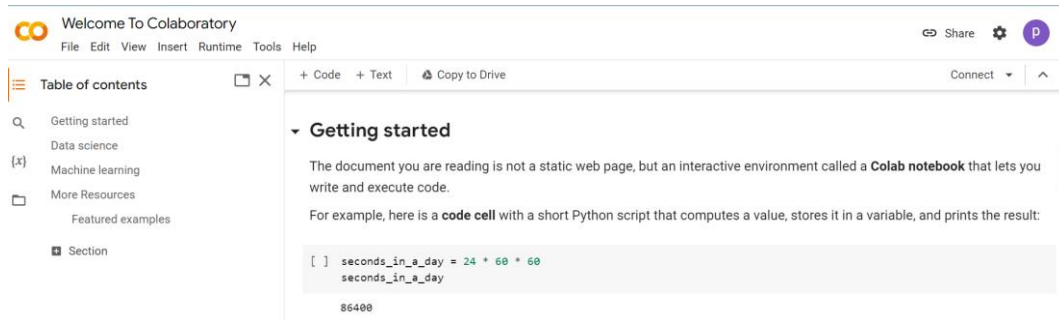
Roboflow merupakan *platform* yang diluncurkan pada Januari 2020, dan menyediakan layanan untuk membangun aplikasi *computer vision* kepada pengembang. Roboflow menyederhanakan proses dalam pemberian label (*labeling*) pada data hingga pelatihan model (*training*).[19]

Pengembang dapat menggunakan Roboflow untuk melakukan beberapa hal sebagai berikut[19]:

1. Anotasi citra atau mengunggah anotasi yang telah ada
2. Melakukan konversi file anotasi VOC XML ke COCO JSON
3. Melakukan pengecekan pelabelan
4. Melakukan *pre-processing* citra
5. Meningkatkan kualitas citra sebagai data untuk dilatih: *flip, rotate, brighten/darken, chop, shear, blur, dan add random noise*.
6. Menghasilkan format anotasi seperti: TFRecords, Create ML dan Turi Create, dan custom YOLOv3 implementations.
7. Memudahkan dalam mengakses kualitas *datasets*
8. Mendapat dan membagikan *dataset* umum

2.11 Google Colab

Colaboratory atau disingkat Colab, merupakan produk penelitian dari Google. Colab memungkinkan penggunaannya untuk menulis dan mengeksekusi kode Python melalui peramban, dan sangat cocok untuk penggunaan dalam *machine learning, data analysis* dan edukasi. Secara teknis, Colab adalah layanan jupyter notebook terhosting yang tidak memerlukan penysetelan untuk digunakan, serta memberikan akses gratis untuk melakukan sumber daya komputasi termasuk penggunaan GPU.[20]



Gambar 2.14 Tampilan Google Colab

Colab notebooks memudahkan pengguna untuk mengkombinasikan antara *executable code* (bahasa mesin) dan *rich text* dalam sebuah dokumen, bersamaan dengan penggunaan gambar, HTML, LaTeX, dan banyak lagi. Hasil dari Colab Notebooks akan secara otomatis tersimpan di dalam akun Google Drive. Selain itu, pengguna dapat membagikan Colab Notebooks dengan orang lain untuk dapat melakukan kolaborasi Bersama.[20]

2.12 Visual Studio Code

Visual Studio Code atau biasa disebut VS Code, adalah sebuah editor kode gratis, *open-source*, dan *cross-platform* yang dikembangkan oleh Microsoft. Berdasarkan survei Stack Overflow Developer, VS Code merupakan environment pengembang paling populer di tahun 2019, serta memiliki banyak fitur yang dapat dikostumisasi dan tidak hanya untuk melakukan pengeditan source code saja, tapi juga memiliki dukungan built-in untuk kolaborasi dan environments cloud-hosted. Dukungan built-in ini hanya untuk Javascript, Typescript, HTML, dan CSS. Meskipun begitu, VS Code juga mendukung banyak bahasa tambahan, seperti Python melalui extensions.[21]

2.13 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman yang sering diaplikasikan pada pengembangan situs web, perangkat lunak, ilmu data, dan *machine learning*. Dengan keunggulan Python yang efisien dan mudah dipelajari serta dapat dijalankan di berbagai *platform*, menjadikan Python sebagai salah satu bahasa pemrograman populer di kalangan pengembang.[22]

Dalam penggunaannya, python dibantu dengan berbagai *library*. *Library* adalah sekumpulan kode yang digunakan oleh pengembang dengan tujuan untuk mempercepat dalam penulisan kode dari awal.[22] Pada *project* YOLOv5 dan Deep SORT digunakan beberapa *library* pendukung sebagai berikut:

2.13.1 opencv-python

OpenCV atau *Open Source Computer Vision* merupakan *library* Python yang digunakan pengembang dalam pembuatan aplikasi *computer vision*. Penggunaan *library* ini dapat dilakukan pada berbagai bahasa pemrograman yaitu C, C++, Java, dan Python. Pada *library* opencv memuat ratusan algoritma *computer vision*, serta memiliki struktur modular.[23]

2.13.2 Pillow

Pillow merupakan *fork* dari PIL yang dibuat oleh Alex Clark dan para kontribusi. *Fork* yaitu repositori baru yang membagikan kode dan penyetelan visibilitas dengan repositori asli. PIL atau *Python Imaging Library* ciptaan Fredrik Lundh dan para contributor, adalah sebuah *library* yang digunakan untuk menambah kapabilitas dalam pemrosesan citra pada interpreter Python.[24]

2.13.3 PyTorch

PyTorch adalah *library* tensor yang telah teroptimasi untuk keperluan *deep learning* menggunakan GPU dan CPU yang dikembangkan oleh Facebook. Pada *library*,

terdapat *package* torch yang berisi struktur data untuk tensor multi-dimensi dan pendefinisian operasi matematika pada tensor.[25]

2.13.4 tb-nightly

Tb-nightly adalah *package* Python sebagai TensorBoard yang bertujuan untuk memeriksa dan memahami proses dan grafik dari tensorflow pengembang.[26]

2.13.5 imageio

Imageio adalah *library* Python yang menyediakan antarmuka yang dengan mudah membaca dan menulis data citra secara luas, termasuk citra animasi, data volumetric, dan format ilmiah.[27]

2.14 Django

Django adalah kerangka kerja *web* gratis, yang bersifat *open source* dan dapat mempercepat pengembang dalam mengembangkan sebuah aplikasi *web* yang sedang dibangun menggunakan bahasa pemrograman Python. Pada Django, tabel *database* (basis data) diubah menjadi kelas Python. Selanjutnya, aplikasi *web* mengakses dan mengatur data melalui *models* pada Django. Django menawarkan *shortcut* terintegrasi penuh dengan basis data aplikasi pengembang, yaitu menyediakan CRUD (*create, read, update, delete*), *HttpResponse* dan *cross-site scripting*, menyediakan kapabilitas manajemen bagi pengguna, fitur perangkat lunak administrasi dan masih banyak lagi.[28]

Kerangka kerja Django menjadi pilihan yang tepat ketika pengembang ingin membuat aplikasi *web* yang melibatkan *cross-site scripting* (XSS) dan mengharapkan aplikasi dapat menangani pengguna dengan jumlah yang besar atau kumpulan fitur bersifat kompleks, seperti sebuah konektivitas API maupun autentikasi pengguna.[28]

2.15 Confusion Matrix

Confusion Matrix adalah representasi dari pengklasifikasian kinerja sebuah model dengan format biner yang mewakili nilai performa model yang dibuat.[29] *Confusion Matrix* merupakan *array* persegi yang berdimensi $n \times n$, dengan n yang menyatakan jumlah atau banyak kelas seperti pada Gambar 2.15.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2.15 *Confusion Matrix*

Kolom pada *Confusion Matrix* mewakili data uji, sedangkan baris pada *Confusion Matrix* mewakili label yang diberikan oleh pengklasifikasi.[30] Evaluasi tingkat efektivitas suatu model klasifikasi dapat dihitung dengan menggunakan metrik *precision*, *recall*, *F1 score*, dan *accuracy*, berdasarkan pada *Confusion Matrix*. [31]

Precision. Adalah perbandingan pengamatan positif yang diprediksi secara benar (*True Positive*), dengan jumlah pengamatan positif (*True Positive* dan *False Positive*). Tingginya nilai *precision* bergantung terhadap nilai *False Positive*. Semakin rendah nilai FP, maka semakin tinggi nilai *precision*. [31] *Precision* didefinisikan dengan rumus sebagai berikut:

$$Precision = \frac{True\ Positive}{Actual\ Results} = \frac{TP}{TP + FP}$$

Recall. Adalah perbandingan pengamatan positif yang diprediksi benar (*True Positive*), dengan keseluruhan pengamatan pada kelas sesungguhnya (*True Positive*

dan *False Negative*). Tingginya nilai *recall* bergantung terhadap nilai *False Negative*. Semakin rendah nilai FN, maka semakin tinggi nilai *recall*. [31] *Recall* didefinisikan dengan rumus sebagai berikut:

$$Recall = \frac{True\ Positive}{Predicted\ Results} = \frac{TP}{TP + FN}$$

F1 Score. Adalah rata-rata harmonik dari *precision* dan *recall*. Sehingga, skor dari *F1 score* memperhitungkan *False Positive* (FP) dan *False Negative* (FN). *F1 score* memberikan bobot lebih tinggi pada nilai FN dan FP, dengan tidak membiarkan nilai *True Negative* (TN) mempengaruhi skor. [31] *F1 score* didefinisikan dengan rumus sebagai berikut:

$$F1\ Score = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Accuracy. Metrik *accuracy* adalah perbandingan dari masukan yang diklasifikasikan secara benar pada seluruh kumpulan data. Metrik *accuracy* merupakan pengukuran terbaik ketika menghadapi kumpulan data yang memiliki nilai FP dan FN serupa. [31] *Accuracy* didefinisikan dengan rumus sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

2.16 *Black-box Testing*

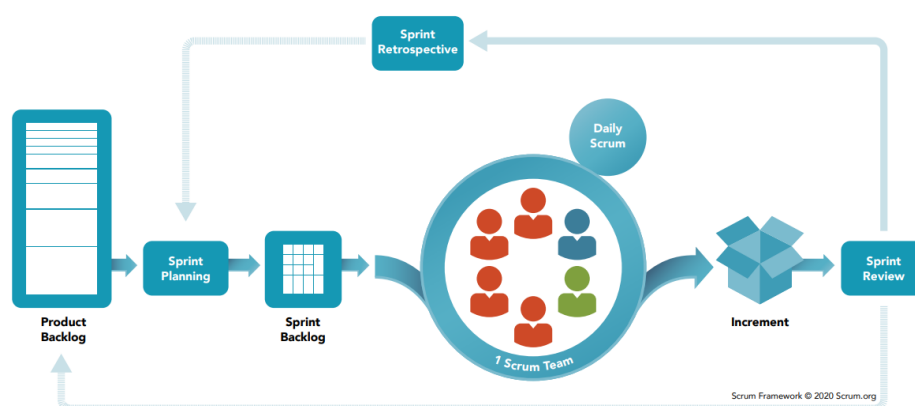
Black-box testing adalah pengujian sistem yang bertujuan untuk memeriksa apakah sebuah sistem atau program berjalan sesuai dengan kebutuhan yang ditentukan. Pengujian ini tidak membutuhkan pengetahuan mengenai cara kerja kode program (implementasi program). [32]

2.17 Greenmetric

Greenmetric adalah pemeringkatan kampus hijau dan kelestarian lingkungan yang pertama kali diperkenalkan oleh Universitas Indonesia pada tahun 2010. Pemeringkatan dilakukan dengan memperhatikan 39 indikator dalam 6 kriteria yang terdiri dari *Setting and Infrastructure*, *Energy and Climate Change*, *Waste, Water, Transportation*, dan *Education*. Universitas Lampung merupakan salah satu universitas yang berpartisipasi dalam program *greenmetric* dengan nama *Greenmetric Universitas Lampung*. Tim ini mendorong peran serta seluruh civitas untuk bersama berkolaborasi dan berkontribusi dalam mengembangkan komunitas kampus yang berkesinambungan. [33]

2.18 Metode Scrum

Scrum adalah kerangka proses yang telah digunakan untuk mengelola pengembangan produk kompleks sejak awal 1990-an. Scrum bukan proses maupun teknik yang digunakan dalam membuat suatu produk, akan tetapi Scrum merupakan kerangka kerja untuk menerapkan berbagai proses dan teknik.



Gambar 2.16 Kerangka kerja Scrum

Berdasarkan Gambar 2.16, siklus Scrum dimulai dari *Product Backlog*, yaitu tahapan yang menjabarkan keseluruhan pekerjaan yang perlu dilakukan, kemudian

terdapat tahapan *Sprint* berkisar 4 minggu. Dari *Sprint* dihasilkan *Sprint Backlog* yang di tahap ini juga *DoD* atau *Definition of Done* ditentukan. [34] Pada *Sprint Backlog* ini, dibagi menjadi beberapa bagian status seperti *In-Progress* dan *done*. Selanjutnya, terdapat *Daily Scrum* yang bertujuan untuk memantau progress pekerjaan yang telah dilakukan dan menentukan pekerjaan yang akan dilakukan nantinya. Proses selanjutnya ialah *Increment* yang dapat ditambahkan sewaktu-waktu jika memang terdapat pekerjaan yang perlu ditambahkan. Kemudian, *Sprint Review* yang dilakukan di akhir *Sprint*, yaitu pemeriksaan hasil *Sprint* yang telah dilakukan. Terakhir, *Sprint Retrospective*, yaitu mengevaluasi bagaimana hasil pekerjaan yang telah dilakukan dan mencari cara untuk dapat menjadi lebih baik lagi. Siklus tersebut kemudian kembali lagi pada *Sprint Planning*, dan seterusnya hingga *DoD* telah tercapai seluruhnya.[34]

Pada penerapannya dalam pengembangan perangkat lunak, *Scrum* dibagi menjadi tiga fase yaitu *pre-game*, *game*, dan *post-game*. [35] Pada *Scrum* terdapat 5 *event* yang dilakukan, berdasarkan Gambar 2.16, yaitu :

1. *Sprint*

Sprint dimulai dengan tahap *sprint planning*. Pada tahap ini dilakukan analisis pada masalah yang akan diselesaikan dan bagaimana pekerjaan itu akan dicapai atau diselesaikan.[36]

2. *Sprint Planning*

Sprint backlog merupakan daftar pekerjaan atau *tasks* hasil kesepakatan dan diskusi yang telah dilakukan pada *sprint planning* yang akan dikerjakan dalam satu *sprint*. [36]

3. *Daily Scrum*

Selama *sprint* berlangsung, diadakan *daily scrum* yang dilakukan dengan durasi 15 menit setiap harinya. Adapun topik pembahasan dalam *daily scrum* yang dilakukan ialah membahas mengenai *tasks* apa saja yang sudah dikerjakan di hari sebelumnya, kendala yang dialami selama pengerjaan, dan *tasks* yang akan dikerjakan

berikutnya. Selanjutnya, hasil dari *daily scrum* ini ditulis dalam catatan harian untuk dijadikan referensi dalam pengembangan sistem selanjutnya.[36]

4. *Sprint Review*

Sprint Review ialah melakukan peninjauan terhadap *progress sprint* yang telah dilakukan dalam satu sprint[36].

5. *Sprint Retrospective*.

Hasil yang didapatkan pada tahap *sprint review* dibahas kembali di tahap *sprint retrospective*. Pada tahap ini, dilakukan peninjauan mengenai hasil yang didapatkan pada *sprint review*. Adapun hasil *sprint retrospective* pada *sprint* ialah berupa penambahan *tasks* pada *sprint backlog*. [36]

2.19 Penelitian Terkait

Tabel 2.1 Penelitian Terkait

Judul	Metode	Hasil
Pengaturan Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Metode Yolo.[37]	YOLOv3 dan SORT (Simple Online Realtime Tracking)	Peneliti melakukan perhitungan kendaraan pada video untuk pengaturan lampu lalu lintas menggunakan algoritma YOLOv3 dan SORT. Pada penelitian digunakan 624 data latih dan 156 data uji, dengan parameter yang ditinjau adalah AP (<i>Average Precession</i>) dan akurasi. Hasil dari pengujian memperoleh nilai AP tertinggi sebesar 0.89 dan akurasi tertinggi sebesar 98.80%.

<p>Analisis Deteksi Objek Citra Digital Menggunakan Algoritma Yolo Dan CNN Dengan Arsitektur RepVGG pada Sistem Pendeteksian Dan Pengenalan Ekspresi Wajah.[38]</p>	<p>YOLOv5s dan RepVGG</p>	<p>Penelitian menggunakan algoritma YOLOv5s dan RepVGG untuk melakukan klasifikasi 7 ekspresi wajah yaitu <i>Neutral, Happy, Sad, Anger, Fear, Surprise, dan Disgust</i>. Pengujian dilakukan dengan jumlah epoch sebanyak 200 epoch dan menggunakan data citra sebanyak 35.887 ekspresi wajah manusia yang masing-masing dikelompokkan menjadi 7 kelas ekspresi wajah. Hasil dari pengujian algoritma YOLO mendapatkan tingkat keyakinan sebesar 89%, dan pengujian keseluruhan sistem mendapatkan tingkat keyakinan sebesar 99.3% untuk ekspresi <i>Happy</i>.</p>
<p>Implementasi Algoritma Region Of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan.[39]</p>	<p><i>Region Of Interest</i> (ROI)</p>	<p>Sistem yang dibangun adalah sistem deteksi dan klasifikasi kendaraan. Penelitian dilakukan menggunakan metode <i>Region Of Interest</i> (ROI) untuk mempercepat waktu pemrosesan citra dalam melakukan deteksi dan klasifikasi kendaraan. Pengujian dilakukan dengan mengumpulkan data video sesuai ketentuan, kemudian menganalisis waktu pemrosesan</p>

		<p>tanpa menggunakan ROI maupun ketika menggunakan ROI pada laptop, raspberry pi3 dan odroid xu4. Hasil yang diperoleh adalah waktu pemrosesan citra menggunakan ROI dengan metode segmentasi MOG2 dan <i>tracking</i> lebih cepat 0.026 s dibandingkan tanpa menggunakan ROI.</p>
<p>Implementasi Deteksi <i>Real Time</i> Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOv5.[40]</p>	YOLOv5	<p>Peneliti membangun sistem pendeteksi objek menggunakan metode YOLOv5 untuk mendeteksi jenis kendaraan di jalan raya. Penelitian menggunakan 1332 citra dengan 9 kelas (bajaj, becak, bus, mobil, mobil molen, mobil pik'up, sepeda, sepeda motor, dan truk. Hasil dari penelitian menggunakan metode YOLOv5 ialah sistem dapat mengenali objek secara konsisten dengan tingkat akurasi yang cukup tinggi sebesar 90%.</p>
<p><i>Counting Vehicle with High-Precision in Brazilian Roads Using YOLOv3 and Deep SORT.</i>[41]</p>	YOLOv3 dan Deep SORT	<p>Peneliti membangun sistem menggunakan algoritma YOLOv3 untuk mendeteksi objek dan Deep SORT untuk melakukan <i>tracking</i> banyak objek. Pada pengujian, peneliti</p>

		<p>menggunakan video dari GRAM <i>dataset</i> dan CD2014 <i>dataset</i>, dan video yang diambil pada jalan raya Brazil. Hasil pengujian ialah sistem mendapatkan nilai <i>precision</i> (presisi) 99.15% pada perhitungan yang dilakukan di GRAM <i>dataset</i> dan CD2014 <i>dataset</i>. Sedangkan pada pengujian video jalan raya Brazil, didapatkan nilai <i>precision</i> (presisi) diatas 90%.</p>
<p><i>Vision-based vehicle detection and counting system using deep learning in highway scenes.</i>[42]</p>	YOLOv3	<p>Peneliti menggunakan metode segmentasi untuk membagi citra jalan raya menjadi dua area, yaitu area <i>remotec</i> (terpencil) dan <i>proximal</i> (terdekat). Kemudian, peneliti menggunakan algoritma YOLOv3 untuk mendeteksi tipe dan lokasi kendaraan. Akhirnya, lintasan kendaraan didapatkan melalui algoritma ORB untuk mendapatkan arah tujuan kendaraan dan jumlah kendaraan. Hasil penelitian mendapatkan akurasi rata-rata arah tujuan kendaraan sebesar 92.3% dan akurasi rata-rata perhitungan kendaraan sebesar 93.2%.</p>

<p><i>Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network.</i>[43]</p>	<p><i>Convolutional Neural Network (CNN)</i></p>	<p>Peneliti melakukan pendeteksian buah menggunakan algoritma CNN. Pada penelitian digunakan 6783 total gambar sebagai <i>dataset</i>. Hasil dari pengujian memperoleh nilai akurasi tertinggi 97.4%.</p>
--	--	---

Berdasarkan tabel 2.1, penelitian sistem deteksi objek dilakukan dengan menggunakan model *Deep Convolutional Neural Network* (D-CNN). Faktor pembeda dari penelitian terletak pada objek sasaran deteksi dan pengimplementasiannya. Berdasarkan penelitian terkait, pendeteksian menggunakan model CNN mendapatkan nilai akurasi diatas 70%. Nilai akurasi tersebut menunjukkan bahwa model CNN sangat efektif dalam penerapannya pada sistem deteksi objek.

Dengan mempertimbangkan kelebihan dan kekurangan pada penelitian terkait, pada penelitian ini, dilakukan perancangan sebuah sistem perhitungan kendaraan yang dapat melakukan deteksi dan pelacakan 3 jenis kendaraan (motor, mobil, dan bis). Algoritma yang digunakan ialah YOLOv5 untuk melakukan pendeteksian objek karena algoritma YOLOv5 memiliki performa paling stabil untuk sistem *realtime*. Waktu inferensi yang dibutuhkan dalam melakukan deteksi berkisar pada rentang 5 ms hingga 20 ms, rentang waktu tersebut lebih rendah apabila dibandingkan dengan YOLO versi sebelumnya.

Selain itu, penggunaan YOLOv5 dalam sistem perhitungan kendaraan masih sedikit sehingga pada penelitian dipilih algoritma YOLOv5. Kemudian, algoritma Deep SORT untuk melakukan pelacakan karena algoritma ini memiliki performa lebih baik dibanding SORT, yaitu mengurangi masalah tertukar ID saat melakukan pelacakan objek. Selain itu, pada penelitian ini, ditambahkan penggunaan ROI (*Region of Interest*) *line* untuk mengoptimalkan kinerja perhitungan yang dilakukan oleh sistem.

Tempat penelitian : Laboratorium Terpadu jurusan Teknik Elektro Universitas Lampung dan Jalan kampus Universitas Lampung.

3.2 Alat dan Bahan

Alat dan bahan memiliki peranan sebagai eksekutor dalam penelitian yang dilakukan, mulai dari tahap perancangan sistem hingga penyelesaian sistem. Alat dan bahan yang digunakan penelitian ini yaitu:

Tabel 3.2 Alat penelitian

No	Jenis Perangkat	Perangkat	Spesifikasi	Keterangan
1	Perangkat Keras	Laptop	Intel® Core™ i7-1255U, RAM 8.00 GB	Perangkat utama yang digunakan dalam pembangunan dan pengujian <i>project</i>
		<i>Webcam</i>	<i>Camera</i> 4MP, 30Fps <i>Frame rate.</i>	Perangkat yang digunakan dalam pengambilan <i>sample</i> pada pengujian.
		<i>Tripod</i>	2 Meter	Perangkat yang digunakan dalam pengambilan <i>sample</i> pada pengujian.
2	Perangkat Lunak	Python	Versi 3.9.10	Bahasa pemrograman yang digunakan dalam pembuatan sistem deteksi dan pelacakan objek.
		YOLO	Model YOOLov5s	Algoritma deteksi objek yang digunakan pada <i>project</i> .

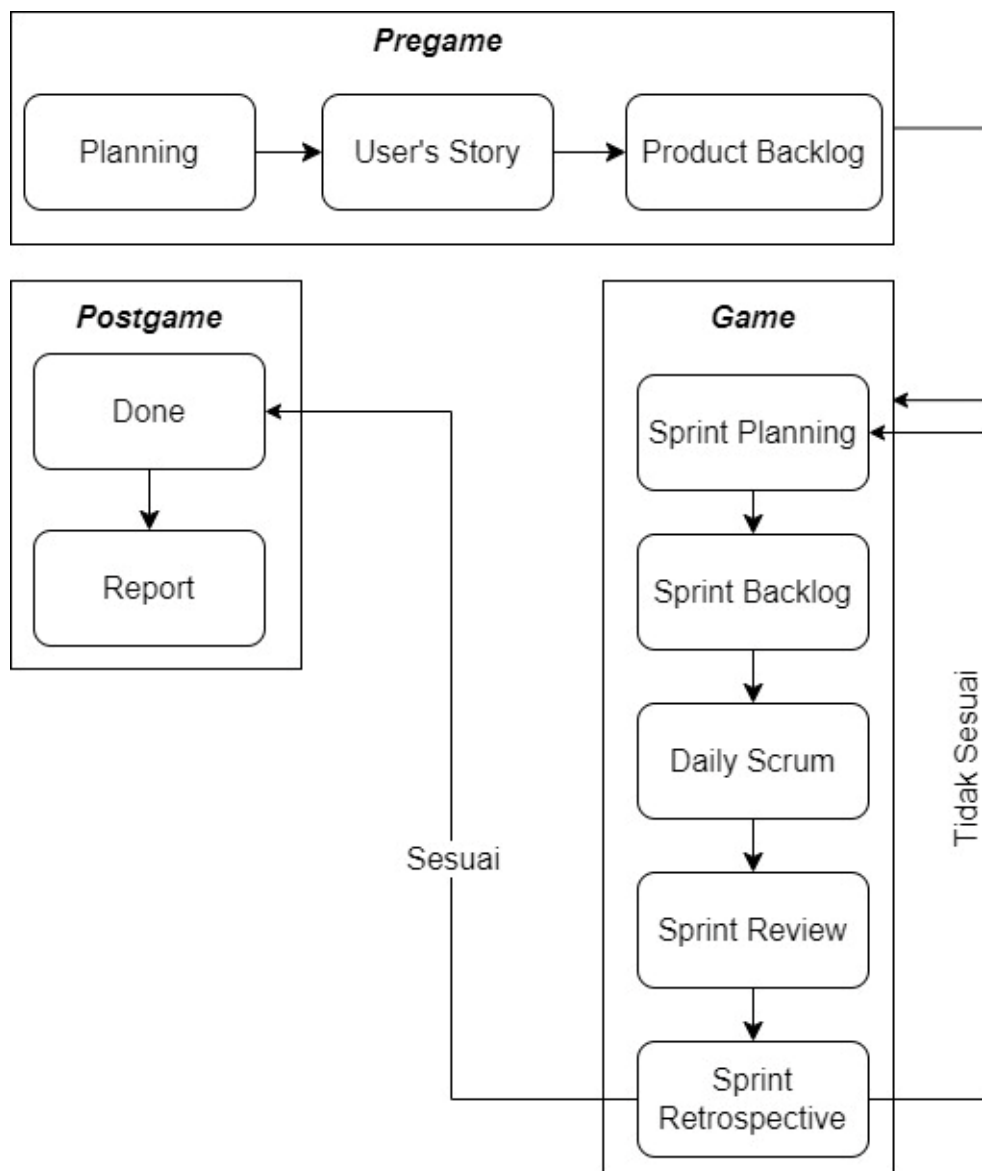
		Deep SORT	Model osnet_x0_2 5	Algoritma <i>multiple object tracking</i> (MOT) yang digunakan pada <i>project</i> .
		Django	Versi 4.1.7	Kerangka kerja <i>web</i> yang digunakan dalam pembangunan aplikasi <i>project</i> .
		Google Colaboration		<i>Code editor</i> yang digunakan untuk melatih data dan mendapatkan model latih deteksi.
		Roboflow		Aplikasi <i>web</i> yang digunakan untuk melakukan <i>pre-processing</i> dan <i>augmentation dataset</i> .
		<i>Database</i>	MySQL	<i>Database</i> yang digunakan untuk menyimpan hasil video yang diunggah melalui aplikasi <i>web</i> .
		Visual Studio Code	Versi 1.75.1	<i>Code editor</i> yang digunakan dalam pembangunan sistem <i>project</i> .

Bahan penelitian :

1. 3 objek kendaraan (motor, mobil, dan bus) sebanyak 199 gambar. Dengan rincian anotasi kelas motor sebanyak 213, mobil sebanyak 84, dan bis sebanyak 33.

3.3 Tahapan Penelitian

Metode penelitian yang digunakan dalam pembuatan sistem deteksi dan perhitungan kendaraan di Kawasan jalan kampus Universitas Lampung ialah metode Scrum. Metode Scrum adalah kerangka proses untuk mengelola pengembangan produk kompleks. Pada penelitian proses Scrum dibagi menjadi 3 fase yaitu, *Pregame*, *Game*, dan *Postgame* seperti pada Gambar 3.1.



Gambar 3.1 Diagram alur penelitian menggunakan metode Scrum

3.3.1 *Pregame*

Fase *pregame* berfokus pada aktivitas dalam mengidentifikasi permasalahan yang dihadapi dan melakukan perencanaan dalam penentuan solusi dari masalah yang dihadapi. Pada Fase *pregame* terdapat 3 tahapan yaitu *planning*, *user's story*, dan *product backlog*. Keluaran yang didapatkan dari fase ini berupa *product backlog*.

a. *Planning*

Tahap *planning* merupakan tahap pertama pada fase *pregame*, yang meliputi perencanaan terhadap rancangan sistem deteksi dan perhitungan. Pada penelitian, tahapan ini dimulai dengan melakukan studi banding melalui kajian literasi terkait penelitian. Studi banding dilakukan dengan tujuan untuk menambah wawasan baru dan untuk lebih memahami pengembangan sistem yang dibangun. Langkah selanjutnya ialah melakukan *user research* menggunakan metode wawancara dengan *stakeholder* terkait, yaitu tim *greenmetric*.

b. *User's Story*

Setelah menyelesaikan tahap *planning*, tahap selanjutnya yang dilakukan adalah pembuatan *user's story*. *User's story* dibuat menggunakan *tools* Trello, dengan format “Sebagai pengguna, saya menginginkan [kegunaan] sehingga saya dapat [tujuan]”. Setelah didapatkan *user's story* selanjutnya akan ditentukan beberapa fitur sistem berdasarkan *user's story* yang telah dibuat.

c. *Product Backlog*

Product backlog berisikan daftar *item* yang disusun dengan mempertimbangkan fitur sistem yang dibuat pada langkah sebelumnya. *Item* pada *product backlog* selanjutnya dijadikan acuan utama dalam pengerjaan sistem deteksi dan perhitungan. Pemberian level prioritas yaitu *high*, *medium*, dan *low* akan mempengaruhi urutan pengerjaan *item* pada setiap *sprint*. Semakin tinggi prioritas

suatu *item*, maka *item* tersebut akan dikerjakan terlebih dahulu dibandingkan *item* dengan prioritas lebih rendah.

3.3.2 Game

Tahapan *game* merupakan tahapan pengembangan kebutuhan sistem yang dilakukan berdasarkan *product backlog* yang telah dirancang. Pengembangan sistem pada penelitian ini menggunakan metode Scrum yang digunakan untuk mengelola pengembangan produk kompleks dengan 5 tahapan dalam Scrum sebagai berikut:

a. Sprint Planning

Sprint dimulai dengan tahap *sprint planning*. Adapun lama pengerjaan dalam setiap 1 *sprint* adalah 10 hari kerja dalam 2 minggu. Pada tahap ini dilakukan penentuan *item* yang dikerjakan pada suatu *sprint* dengan melihat level prioritas *item*. Selanjutnya, dilakukan analisis pada *item product backlog* untuk menentukan langkah bagaimana *item* tersebut akan diselesaikan.

b. Sprint Backlog

Hasil analisa yang dilakukan pada tahapan sebelumnya, menghasilkan daftar tahapan untuk mencapai *item product backlog* yang dipilih. Daftar tahapan tersebut merupakan tahapan yang harus dilakukan selama 1 *sprint*.

c. Daily Scrum

Selama *sprint* berlangsung, diadakan *daily scrum* yang dilakukan dengan durasi 15 menit setiap harinya. Pada penelitian ini, *daily scrum* dilakukan selama hari kerja setiap minggunya.

d. *Sprint Review*

Pada pengembangan sistem deteksi dan perhitungan kendaraan, peninjauan *progress* dibantu menggunakan *tools* Trello. Daftar pada Trello yang digunakan pada peninjauan adalah daftar *In-Progress* dan *Done*. Kedua daftar tersebut memberikan informasi tentang pekerjaan yang masih dalam proses dan pekerjaan yang telah selesai. Selain peninjauan, pada tahap ini dilakukan *testing* sistem secara *realtime*. Pengambilan *sample testing* menggunakan laptop, *tripod* dan *webcam* selama 1 jam setiap harinya dalam 5 hari. Hasil pengujian diukur menggunakan *confusion matrix* dan *black-box testing*.

1. *Testing* menggunakan *Confusion Matrix*

Pengujian *Confusion Matrix* dilakukan untuk mengetahui performa sistem dalam melakukan deteksi. Hasil dari pengujian ini digunakan sebagai bahan pertimbangan dan evaluasi mengenai kelayakan sistem dalam melakukan deteksi objek. Adapun nilai yang digunakan sebagai acuan dalam pengujian ini adalah *precision*, *recall*, *F1 score*, dan *accuracy* dengan persamaan sebagai berikut:

$$Precision = \frac{True\ Positive}{Actual\ Results} = \frac{TP}{TP + FP}$$

$$Recall = \frac{True\ Positive}{Predicted\ Results} = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Hasil perhitungan berdasarkan persamaan diatas akan diubah kedalam skala persen. Sehingga masing-masing hasil akan dikalikan dengan 100%.

2. *Testing* menggunakan *Black Box Testing*

Pengujian selanjutnya ialah *Black-box testing*. Pengujian ini dilakukan untuk menilai fungsi-fungsi sistem berdasarkan pendeskripsian *user's story*. Pada penelitian ini, pengujian dilakukan menggunakan salah satu tipe pada *black box testing*, yaitu *functional testing*. Tipe pengujian ini dilakukan berdasarkan kebutuhan fungsional sistem dengan scenario uji yang terdapat dalam tabel 3.3:

Tabel 3.3 Skenario uji pada pengujian sistem

No	Skenario Uji	Hasil yang diharapkan	Hasil Pengamatan	Kesimpulan
1	Sistem memiliki nilai akurasi >70%	Nilai akurasi sistem >70%		
2	Sistem dapat mendeteksi dan membedakan 3 jenis kendaraan (motor, mobil, bis) secara tepat.	Sistem dapat mendeteksi dan membedakan 3 jenis kendaraan (motor, mobil, bis) secara tepat.		
3	Sistem dapat menghitung tiap jenis kendaraan yang terdeteksi.	Sistem dapat menghitung tiap jenis kendaraan yang terdeteksi.		
4	Sistem dapat menyimpan hasil perhitungan tiap jenis kendaraan yang terdeteksi.	Sistem dapat menyimpan hasil perhitungan tiap jenis kendaraan yang terdeteksi.		

5	Sistem dapat menampilkan data hasil perhitungan tiap jenis kendaraan yang terdeteksi pada halaman <i>report</i> .	Sistem dapat menampilkan data hasil perhitungan tiap jenis kendaraan yang terdeteksi pada halaman <i>report</i> .		
---	---	---	--	--

e. *Sprint Retrospective*

Selanjutnya, setelah melakukan peninjauan *progress* pekerjaan pada *sprint review* dilakukan peninjauan dan refleksi Sumber Daya Manusia (SDM). Peninjauan ini dilakukan untuk meningkatkan kualitas dan efektivitas pengerjaan untuk *sprint* selanjutnya. Hasil dari tahapan ini berupa keputusan mengenai hal yang dilakukan pada *sprint* berikutnya.

3.3.3 *Post-game*

Fase terakhir dari penelitian ini adalah fase *postgame*. Pada fase *postgame*, sistem yang dibangun telah memenuhi kebutuhan pengguna berdasarkan permasalahan yang dihadapi.

a. *Report*

Report adalah tahapan yang terdapat dalam fase *postgame*. Pada tahapan ini dilakukan pembuatan laporan berdasarkan aktivitas penelitian yang telah dilakukan. Hasil laporan yang telah dibuat akan dianalisa untuk pengambilan kesimpulan dan saran terhadap pengembangan sistem yang telah dibangun.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut:

1. Sistem *tracking* dan perhitungan kendaraan menggunakan *object tracking* berhasil melacak dan menghitung tiga jenis kendaraan yaitu motor, mobil, dan bis dengan algoritma YOLOv5 dan DeepSORT. Hasil perhitungan yang didapat berhasil disimpan ke dalam *database*.
2. *Website report* berhasil menampilkan informasi data hasil *tracking* dan perhitungan menggunakan *framework Django*.
3. Berdasarkan perhitungan *confusion matrix* data hasil *testing* secara *realtime*, didapatkan nilai rata-rata *precision* sebesar 99%, *recall* sebesar 97%, *F1 Score* sebesar 97.2% dan *accuracy* sebesar 96.8%. Sehingga sistem memiliki performa yang cukup baik dalam mendeteksi kendaraan.
4. Berdasarkan hasil perhitungan kendaraan pada *testing* secara *realtime*, didapatkan persentase ketepatan sistem dalam melakukan perhitungan sebesar 97.65%. Hal ini dipengaruhi oleh faktor *overlapping object* dan kesalahan sistem dalam mendeteksi sebuah objek selama *tracking*.
5. Pembuatan sistem *tracking* dan perhitungan kendaraan yang dilaksanakan menggunakan kerangka proses Scrum selama 30 hari telah sesuai, dengan iterasi sebanyak tiga kali *sprint*. *Sprint I* berfokus pada pembuatan sistem dalam mendeteksi tiga jenis kendaraan yang berbeda. Lalu, *sprint II* berfokus pada penambahan algoritma supaya sistem dapat melakukan perhitungan otomatis. Terakhir, *sprint III* berfokus pada pembuatan halaman *report*.

5.2 Saran

Berdasarkan penelitian yang dilakukan, terdapat beberapa saran untuk pengembangan sistem selanjutnya, antara lain:

1. Melakukan pengembangan pada sistem perhitungan kendaraan supaya dapat melakukan perhitungan emisi karbon secara otomatis.
2. Menambahkan *dataset* kendaraan untuk meningkatkan akurasi sistem dalam mendeteksi tiap jenis kendaraan.
3. Melakukan pemindahan posisi atau lokasi kamera untuk mengurangi faktor *occlusion* selama dilakukan deteksi objek.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [1] Y. Park, L. M. Dang, S. Lee, D. Han, and H. Moon, "Multiple object tracking in deep learning approaches: A survey," *Electronics (Switzerland)*, vol. 10, no. 19. MDPI, Basel, Oct. 02, 2021. doi: 10.3390/electronics10192406.
- [2] Mardiana, M. A. Muhammad, and Y. Mulyani, "Library Attendance System using YOLOv5 Faces Recognition," in *2021 International Conference on Converging Technology in Electrical and Information Engineering (ICCTEIE)*, 2021, pp. 68–72. doi: 10.1109/ICCTEIE54047.2021.9650628.
- [3] F. Yang, X. Zhang, and B. Liu, "Video object tracking based on YOLOv7 and DeepSORT," *Journal of Latex Class Files*, vol. 14, no. 8, p. 2, Jul. 2022.
- [4] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*, V1.8. San Francisco: Machine Learning Mastery, 2019.
- [5] K. C. Kirana, *Pengolahan Citra Digital: Teori dan Penerapan Pengolahan Citra Digital pada Deteksi Wajah*. Malang: Ahlimedia Book, 2021.
- [6] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Springer International Publishing, 2022.
- [7] R. Kumar Rout, R. Naaz Mir, S. Umer, and V. Kumar Sharma, *Advancement of Deep Learning and Its Applications in Object Detection and Recognition*. Aalborg: River Publishers, 2023.

- [8] S. Valeev and N. Kondratyeva, *Process Safety and Big Data*. Oxford: Elsevier Science, 2021.
- [9] B. Bazeer Ahamed, J. Joshua Thomas, P. Vasant, and P. Karagoz, *Deep Learning Techniques and Optimization Strategies in Big Data Analytics*. Hershey: IGI Global, 2019.
- [10] P. Spagnolo, P. Luigi Mazzeo, and S. Ramakrishnan, *Visual Object Tracking with Deep Neural Networks*. London: IntechOpen, 2019.
- [11] A. K Roy-Chowdhury and B. Song, *Camera Networks: The Acquisition and Analysis of Videos Over Wide Areas*. Zug: Springer International Publishing, 2022. doi: 10.1007/978-3-031-01811-4.
- [12] I. Chatterjee, *Machine Learning and Its Application: A Quick Guide for Beginners*. Sharjah: Bentham Science Publishers, 2021.
- [13] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv*, 2018. <https://pjreddie.com/darknet/yolo/> (accessed Mar. 14, 2023).
- [14] Y. Liu and S. Mehta, *Hands-On Deep Learning Architectures with Python*. Birmingham: Packt Publishing, 2019.
- [15] S. Ranjan and S. Senthilarasu, *Applied Deep Learning and Computer Vision for Self-Driving Cars*. Birmingham: Packt Publishing, 2020.
- [16] G. Jocher, A. Chaurasia, and Laughing, "YOLO: A Brief History," *Ultralytics*, Dec. 05, 2022. <https://docs.ultralytics.com/> (accessed Mar. 14, 2023).
- [17] K. Kar, *Mastering Computer Vision with TensorFlow 2.x*. Birmingham: Packt Publishing, 2020.

- [18] S. J. D. Prince, *Computer Vision: Models, Learning, and Inference*. New York: Cambridge University Press, 2012.
- [19] Roboflow, “Overview - Roboflow,” *Roboflow*. <https://docs.roboflow.com/> (accessed Mar. 17, 2023).
- [20] P. G. Naik, *Conceptualizing Python in Google COLAB*. Bilaspur: Shashwat Publication, 2023.
- [21] A. Speight, *Visual Studio Code for Python Programmers*, 1st ed. Hoboken: Wiley, 2021.
- [22] J. Manuel Ortega, *Mastering Python for Networking and Security*. Birmingham: Packt Publishing, 2018.
- [23] Q. Nguyen, *Mastering Concurrency in Python*. Birmingham: Packt Publishing, 2018.
- [24] Ö. Teker, *Profound Python Libraries*, First Edition. Istanbul: Godoro Publishing, 2022.
- [25] H. Karau and B. Lublinsky, *Scaling Python with Ray*. Sebastopol: O’Reilly Media, 2022.
- [26] N. Hanum Harani and M. Hasanah, *Deteksi Objek dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Berbasis Python*. Bandung: Kreatif Industri Nusantara, 2020.
- [27] S. Singh, M. Tiwari, and Shrivastava, *Flexible Electronics for Electric Vehicles*. Jaipur: Springer Nature Singapore, 2022.
- [28] International Business Machines Corporation, “What is Django?” <https://www.ibm.com/topics/django> (accessed Mar. 17, 2023).

- [29] L. Sharma and M. Carpenter, *Computer Vision and Internet of Things*, First. Boca Raton: Chapman and Hall/CRC, 2022. doi: 10.1201/9781003244165.
- [30] P. Mather and B. Tso, *Classification Methods for Remotely Sensed Data*, Second. CRC Press, 2016.
- [31] R. Chakraborty, A. Ghosh, and J. K. Mandal, *Machine Learning Techniques and Analytics for Cloud Security*, First. Hoboken: Wiley, 2021.
- [32] R. Bierig, S. Brown, E. Galván, and J. Timoney, *Essentials of Software Testing*. Cambridge: Cambridge University Press, 2021. doi: 10.1017/9781108974073.
- [33] Universitas Indonesia, “UI Greenmetric.” <https://greenmetric.ui.ac.id/> (accessed Jul. 21, 2023).
- [34] G. Verheyen, *Scrum – A Pocket Guide – 3rd edition*, Third. Hertogenbosch: Van Haren Publishing, 2021.
- [35] F. Portela, M. Filipe Santos, and T. Guarda, *Advanced Research in Technologies, Information, Innovation and Sustainability*. Manhattan: Springer International Publishing, 2021.
- [36] K. Schwaber and J. Sutherland, “The Scrum Guide The Definitive Guide to Scrum: The rules of the Game,” 2017.
- [37] M. Irfan Hermawan, I. Iwut Tritoasmoro, and N. Ibrahim, “Pengaturan Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Metode Yolo,” *e-Proceeding of Engineering*, vol. 8, Feb. 2021.
- [38] A. Pratiwi Saputri, A. Taqwa, and S. Soim, “Analisis Deteksi Objek Citra Digital Menggunakan Algoritma Yolo dan CNN Dengan Arsitektur REPVGG Pada Sistem Pendeteksian dan Pengenalan Ekspresi Wajah,” *Jurnal Ilmiah Indonesia*, vol. 7, no. 9, Sep. 2022.

- [39] A. Hendiranto Pratomo, W. Kaswidjanti, and S. Mu'arifah, "Implementasi Algoritma Region Of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 7, no. 1, pp. 155–162, Feb. 2020.
- [40] D. Iskandar Mulyana and M. Ainur Rofik, "Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOv5," *Jurnal Pendidikan Tambusai*, vol. 6, no. 3, 2022.
- [41] A. M. Santos, C. J. A. Bastos-Filho, A. M. A. Maciel, and E. Lima, "Counting Vehicle with High-Precision in Brazilian Roads Using YOLOv3 and Deep SORT," in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, IEEE, Nov. 2020, pp. 69–76. doi: 10.1109/SIBGRAPI51738.2020.00018.
- [42] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *European Transport Research Review*, vol. 11, no. 1, p. 51, Dec. 2019, doi: 10.1186/s12544-019-0390-4.
- [43] R. S. Latha *et al.*, "Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, Jan. 2021, pp. 1–5. doi: 10.1109/ICCCI50826.2021.9402513.
- [44] X. Xiang, M. Zhai, N. Lv, and A. El Saddik, "Vehicle counting based on vehicle detection and tracking from aerial videos," *Sensors (Switzerland)*, vol. 18, no. 8, Aug. 2018, doi: 10.3390/s18082560.