

**BUKU TUGAS AKHIR
CAPSTONE DESIGN**



TRAFFIC VIOLATION DETECTION USING DRONE

Oleh :

Hadid Candra Diputra/1103194165

Muhammad Iqbal Dharmawan/1103194125

Zalva Ihilani Pasha/1103194182

**PRODI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
BANDUNG
2023**

LEMBAR PENGESAHAN

BUKU CAPSTONE DESIGN

DETEKSI PELANGGARAN LALU LINTAS MENGGUNAKAN DRONE

TRAFFIC VIOLATION DETECTION USING DRONE

Telah disetujui dan disahkan sebagai bagian dari Capstone Design
Program S1 Teknik Komputer
Fakultas Teknik Elektro
Universitas Telkom
Bandung

Disusun oleh:

Hadid Candra Diputra/1103194165

Muhammad Iqbal Dharmawan/1103194125

Zalva Ihilani Pasha/1103194182

Bandung, 16 Agustus 2023

Menyetujui,

Pembimbing 1

Pembimbing 2


Casi Setianingsih, S.T., M.T.
NIP. 19890019-11


Anggunmekka Luhur Prasasti, S.T., M.T.
NIP. 15900014-1

LEMBAR PERNYATAAN ORISINALITAS

Saya, yang bertanda tangan di bawah ini

Nama : Muhammad Iqbal Dharmawan
NIM : 11013194125
Alamat : Cimenyan Townhouse No.2 , Jl. Padasuka Atas, Cimenyan.
No. Telepon : 081280992872
Email : iqbal darm11@gmail.com

Menyatakan bahwa Buku Capstone Design ini merupakan karya orisinal saya sendiri bersama dengan kelompok Capstone Design saya, dengan judul:

DETEKSI PELANGGARAN LALU LINTAS MENGGUNAKAN DRONE

TRAFFIC VIOLATION DETECTION USING DRONE

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila di kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian karya ini.

Bandung, 5 Juli 2023



Muhammad Iqbal Dharmawan
11013194125

LEMBAR PERNYATAAN ORISINALITAS

Saya, yang bertanda tangan di bawah ini

Nama : Zalva Ihilani Pasha
NIM : 11013194182
Alamat : Jl. Ketang Ketang no.6 rawamangun
No. Telepon : 081519121716
Email : zalvapasha1@gmail.com

Menyatakan bahwa Buku Capstone Design ini merupakan karya orisinal saya sendiri bersama dengan kelompok Capstone Design saya, dengan judul:

DETEKSI PELANGGARAN LALU LINTAS MENGGUNAKAN DRONE

TRAFFIC VIOLATION DETECTION USING DRONE

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila di kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian karya ini.

Bandung, 5 Juli 2023



Zalva Ihilani Pasha

11013194182

LEMBAR PERNYATAAN ORISINALITAS

Saya, yang bertanda tangan di bawah ini

Nama : Hadid Candra Diputra
NIM : 11013194165
Alamat : ...
No. Telepon : 089638930451
Email : hadidcandra343@gmail.com

Menyatakan bahwa Buku Capstone Design ini merupakan karya orisinal saya sendiri bersama dengan kelompok Capstone Design saya, dengan judul:

DETEKSI PELANGGARAN LALU LINTAS MENGGUNAKAN DRONE

TRAFFIC VIOLATION DETECTION USING DRONE

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila di kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian karya ini.

Bandung, 5 Juli 2023



Hadid Candra Diputra

11013194165

ABSTRAK

Lalu lintas yang semakin padat dan kompleks di indonesia telah menyebabkan meningkatnya risiko pelanggaran lalu lintas yang mampu mengancam keselamatan pengguna jalan. Metode konvensional untuk mendeteksi pelanggaran lalu lintas, seperti kamera cctv, memiliki keterbatasan pada cakupan area dan mobilitasnya. Hal ini menyulitkan upaya penegakan hukum dan mengakibatkan tingginya angka pelanggaran yang tidak terdeteksi. Dalam konteks ini, penggunaan drone sebagai alat pemantauan lalu lintas menawarkan potensi untuk meningkatkan efisiensi dan efektivitas deteksi pelanggaran lalu lintas.

Studi ini mengusulkan sistem deteksi pelanggaran lalu lintas berbasis drone yang mengintegrasikan teknologi penginderaan visual dan kecerdasan buatan. Drone mampu mencakup area yang lebih luas dan dapat menjangkau lokasi yang sulit dijangkau oleh kamera tetap. Dengan menggabungkan teknologi drone dan *object detection* akan digunakan untuk mendeteksi pelanggaran seperti tidak menggunakan helm, berhenti di atas garis *zebra cross*, dan mampu mengenali digit plat nomor dari kendaraan tersebut dengan menggunakan teknologi *optical character recognition*. Sistem akan didukung oleh algoritma kecerdasan buatan yang telah dilatih untuk mengidentifikasi pola pelanggaran lalu lintas dari tangkapan gambar. Data yang diperoleh dari drone akan diproses secara real-time dan dapat diakses oleh pihak penegak hukum untuk ditindak lebih lanjut.

Pengujian akhir sistem deteksi pelanggaran lalu lintas menggunakan drone menunjukkan hasil yang cukup menjanjikan. Drone mampu mendeteksi jenis pelanggaran lalu lintas yang disebutkan di atas dengan akurasi yang cukup memuaskan. Kombinasi teknologi drone, *object detection* dan *optical character recognition* menghasilkan sistem yang dapat secara efektif mengidentifikasi pelanggaran lalu lintas dengan ketepatan waktu yang tinggi. Proses pelatihan model melibatkan pembagian data menjadi 88% untuk pelatihan dan 12% untuk validasi dan test dengan learning rate sebesar 0.0001, batch size 8, dan dijalankan selama 100 epoch. Hasil parameter yang dicapai dari model ini menunjukkan precision 0.742, recall 0.798, F1-score 0.768, mAP@[0.5] 0.81, dan mAP@[0.5:0.95] 0.592. Dalam konteks aplikasi, model ini menghasilkan tingkat akurasi sebesar 90.51% dalam mendeteksi pelanggaran tidak memakai helm, 69.1% untuk deteksi berhenti di zebra cross selama 3 detik, dan 11.92% untuk *Optical Character Recognition*. Dalam pengujian lapangan, sistem berhasil menangkap pelanggaran lalu lintas yang sebelumnya sulit diawasi oleh kamera tetap, membuktikan potensi drone sebagai solusi inovatif dalam upaya meningkatkan keamanan dan penegakan hukum lalu lintas. Dengan pengembangan lebih lanjut, sistem ini akan berpotensi menjadi alat yang sangat berguna bagi pihak berwenang dalam menghadapi keselamatan lalu lintas di masa depan.

Kata kunci : Drone, *Object Detection*, *Optical Character Recognition*, helm, zebra cross

ABSTRACT

Increasingly congested and complex traffic in Indonesia has led to an increased risk of traffic violations that can threaten the safety of road users. Conventional methods for detecting traffic violations, such as CCTV cameras, have limitations in their coverage area and mobility. This complicates law enforcement efforts and results in high rates of violations that go undetected. In this context, the use of drones as a traffic monitoring tool offers the potential to increase the efficiency and effectiveness of traffic violation detection.

This study proposes a drone-based traffic violation detection system that integrates visual sensing and artificial intelligence technologies. Drones are able to cover a wider area and can reach locations that are difficult to reach by fixed cameras. By combining drone and object detection technology, it will be used to detect violations such as not wearing a helmet, stopping above the zebra crossing line, and being able to recognize the license plate digits of the vehicle using optical character recognition technology. The system will be supported by an artificial intelligence algorithm that has been trained to identify patterns of traffic violations from captured images. Data obtained from drones will be processed in real-time and can be accessed by law enforcement agencies for further action.

Initial testing of a traffic violation detection system using drones has shown promising results. Drones are capable of detecting the types of traffic violations mentioned above with satisfactory accuracy. The combination of drone technology, object detection and optical character recognition produces a system that can effectively identify traffic violations with high timeliness. The model training process involves a data split of 88% for training and 12% for validation and testing, with a learning rate of 0.0001, a batch size of 8, and run over 100 epochs. The achieved parameters from this model indicate a precision of 0.742, recall of 0.798, F1-score of 0.768, mAP@[0.5] of 0.81, and mAP@[0.5:0.95] of 0.592. In terms of application, this model produces an accuracy level of 90.51% in detecting helmet violation, 69.1% for detecting stopping at crosswalks for 3 seconds, and 11.92% for Optical Character Recognition. In field tests, the system managed to capture traffic violations that were previously difficult to monitor with fixed cameras, proving the potential of drones as an innovative solution in efforts to improve traffic safety and law enforcement. With further development, this system will potentially become a very useful tool for authorities dealing with traffic safety in the future.

Keywords: Drone, *Object Detection*, *Optical Character Recognition*, helmet, zebra cross

KATA PENGANTAR



Puji dan syukur kita panjatkan kepada Allah SWT atas segala rahmat dan karunia-Nya. Tidak lupa shalawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW yang telah memperkenalkan kita pada jalan yang diridhoi Allah SWT. sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan judul "**“DETEKSI PELANGGARAN LALU LINTAS MENGGUNAKAN DRONE”**". Sebagai penulis, kami mengetahui bahwa penelitian ini masih memiliki keterbatasan dan kekurangan, namun kami berharap hasilnya dapat memberikan kontribusi ilmiah dan bermanfaat bagi perkembangan bidang teknologi. Penyusunan tugas akhir ditujukan sebagai syarat kelulusan program studi S1 Teknik Komputer, Fakultas Teknik Elektro, Telkom University.

Kami berharap studi ini dan uji coba ini akan memberikan kontribusi positif dan menginspirasi pembaca yang tertarik dengan studi serupa. Oleh karena itu diharapkan kritik dan saran dari pembaca dengan harapan Tugas Akhir Ini bisa bermanfaat bagi pembaca.

UCAPAN TERIMAKASIH

Dalam penyusunan Tugas Akhir ini tidak terlepas dari dukungan beberapa pihak. Penulis secara khusus mengucapkan terima kasih yang sebesar - besarnya kepada semua pihak yang telah membantu penyusunan Tugas Akhir ini. Penulis banyak menerima bimbingan, informasi dan bantuan dari berbagai pihak yang bersifat moral maupun materil. Pada kesempatan kali ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

1. **Allah SWT.** yang memberikan segalanya, menginginkan segalanya, mengizinkan segalanya, memberkati segalanya dan banyak hal baik lainnya yang bahkan tidak dapat dituliskan oleh penulis, berkat restunya, setiap Segalanya dapat direncanakan dan dilaksanakan dengan baik.
2. **Papah dan Mamah Qibal, Umi dan Aba Zalva** yang telah menjadi orang-orang terkuat yang melindungi penulis dan memberikan segala kasih sayangnya dalam bentuk apapun kepada penulis terutama pendidikan, dan berkat mereka penulis dapat menyelesaikan studinya hingga ke jenjang perguruan tinggi. Terimakasih juga dukungan alat dan dana. Penulis hanya bisa berharap semoga segala kebaikan, keberkahan, rahmat dan segalanya selalu bersama orang tua penulis, dan kita dapat menunjukkan kepada mereka bahwa kita lebih mampu daripada orang tua kita.
3. **Umi, Aba, Raffi, Rafka, Akbar** terima kasih atas kesabaran, dorongan, kesetiaannya mendampingi, memberi semangat, dan terkadang menjadi selingan humor di saat-saat saya merasa tertekan. Setiap kata dan tindakan kalian telah menjadi penerang jalan saya dalam menyelesaikan skripsi ini. rasa terima kasih yang mendalam atas keperluan yang telah diberikan. Dukungan berupa sumber daya ini bukan hanya memudahkan proses, tapi juga menjadi salah satu faktor utama yang memungkinkan saya untuk mencapai titik ini. Semoga segala jerih payah kalian dalam mendidik dan memberikan semangat untuk saya selama ini menjadi amal jariyah yang tidak pernah putus.
4. **Aunty Wawa dan keluarga**, terima kasih atas setiap saran, perhatian, dan canda tawamu yang selalu mampu menghangatkan hati. Terima kasih juga kepada seluruh anggota keluarga yang telah membuka pintu rumah dan hati untuk saya. Setiap momen yang kalian bagi, setiap makan malam bersama, hingga obrolan santai di sore hari, semuanya telah memberikan kenangan tak terlupakan dan mendalam. Sejauh apapun saya dari jakarta, kalian telah membuat saya merasa dekat dengan rumah.

Semoga Allah melimpahkan berkah, kesehatan, dan kebahagiaan kepada kalian semua. Terima kasih telah menjadi keluarga ke-2 bagi saya di luar kota.

5. **Casi Setianingsih, S.T., M.T., Anggunmeka Luhur Prasasti, S.T., M.T** selaku pembimbing 1 dan pembimbing 2, kami ucapkan rasa terima kasih yang tulus atas dedikasi dan bimbingan yang telah diberikan kepada kami selama proses penulisan laporan tugas akhir ini. Dukungan dan motivasi keduanya telah membantu kami mengatasi hambatan-hambatan yang dihadapi selama proses penulisan tugas akhir ini dan kami sangat berterima kasih atas kesempatan yang telah diberikan
6. **Ir. AGUS VIRGONO M.T.** selaku Wali Dosen kami, kami haturkan rasa terima kasih yang tak terhingga atas peran dan dukungan kalian selama perjalanan kami dalam menyelesaikan studi. Semoga segala upaya dan pengabdian bapak sebagai Wali Dosen selalu mendapat penghargaan yang setinggi-tingginya
7. Teman - teman kelas **TK-43-04** : Terima Kasih telah berjuang bersama dari awal perkuliahan hingga sekarang. Semoga diberikan kelancaran dalam karir dan sukses selalu.
8. PBB Blok I No.24 : **Sayed, Zalva, Difa, Bintang, Harry, Imam, Darry, Daffa, dan Pusaka.** Terima kasih selalu memberikan dukungan dan semangat, Kebersamaan kita telah membentuk sebuah tim yang kompak, saling mendukung satu sama lain dalam setiap kebahagiaan dan kesulitan yang kita hadapi. Terima kasih telah menjadi bagian penting dalam perjalanan ini, dan semoga persahabatan kita tetap kuat dan abadi.
9. **Yolo The Cat**, sebagai kucing kontrakan yang setia menemani semua kegiatan penyusunan dokumen TA. Yolo juga berjasa sebagai penghibur dikala stress melanda karena mampu menjadi mainan yang suka dijailin dan diangkat angkat.

DAFTAR ISI

LEMBAR PENGESAHAN.....	1
LEMBAR PERNYATAAN ORISINALITAS.....	2
ABSTRAK.....	5
ABSTRACT.....	6
KATA PENGANTAR.....	7
UCAPAN TERIMAKASIH.....	8
DAFTAR ISI.....	10
DAFTAR GAMBAR.....	12
DAFTAR TABEL.....	13
DAFTAR SINGKATAN.....	16
BAB 1 USULAN GAGASAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Informasi Pendukung Masalah.....	2
1.3 Analisis Umum.....	4
1.4 Kebutuhan yang Harus Dipenuhi.....	5
1.5 Solusi Sistem yang Diusulkan.....	6
1.5.1 Karakteristik Produk.....	6
1.5.2 Skenario Penggunaan.....	8
1.6 Kesimpulan dan Ringkasan CD-1.....	13
BAB 2 DESAIN KONSEP SOLUSI.....	14
2.1 Spesifikasi Produk.....	14
2.2 Verifikasi.....	16
2.2.1 Verifikasi Spesifikasi 1.....	16
2.2.2 Verifikasi spesifikasi 2.....	17
2.2.3 Verifikasi spesifikasi 3.....	19
2.2.4 Verifikasi Spesifikasi 4.....	20
2.2.5 Verifikasi spesifikasi 5.....	20
2.3 Kesimpulan dan Ringkasan CD-2.....	21
BAB 3 DESAIN RANCANGAN SOLUSI.....	22
3.1 Konsep Sistem.....	22
3.1.1. You Only Look Once (YOLO).....	22
3.1.2. Parameter Kinerja YOLO.....	24
3.1.3. Library Frontend dan Backend.....	25
3.1.4. Koneksi Drone dengan Perangkat keras.....	32
3.1.5. Arsitektur utama sistem.....	33
3.1.6. Analisis.....	34
3.1.7. Sistem yang akan Dikembangkan.....	36
3.2 Desain Sistem.....	37
3.3 Pengujian Komponen (Kalibrasi).....	41
3.3.1. Algoritma.....	41
3.3.2. Perangkat Drone.....	43

3.3.3. Perangkat Komputer.....	44
3.3.4. Desain Aplikasi Web.....	45
3.4 Jadwal Pengerjaan.....	46
3.5 Kesimpulan dan Ringkasan CD-3.....	46
BAB 4 IMPLEMENTASI.....	47
4.1 Implementasi Sistem.....	47
4.1.1 Sub-sistem 1 (Dataset).....	47
4.1.2 Mendeteksi Pelanggaran.....	49
4.1.3 Webapp.....	51
4.1.4 Mendeteksi Plat Nomor pada Kendaraan Bermotor.....	52
4.2 Analisis Pengerjaan Implementasi Sistem.....	53
4.2.1. Sub-sistem 1 (Dataset).....	53
4.2.2. Mendeteksi Pelanggaran.....	54
4.2.3. Webapp.....	54
4.2.4. Mendeteksi dan Membaca Plat Nomor.....	55
4.3 Hasil Akhir Sistem.....	55
4.4 Kesimpulan dan Ringkasan CD-4.....	56
BAB 5 PENGUJIAN SISTEM.....	57
5.1 Skema Pengujian Sistem.....	57
5.1.1 Pengujian Model.....	57
5.1.2 Pengujian Sistem.....	57
5.2 Proses Pengujian.....	58
5.2.1 Pengujian 1: Pengujian Model.....	58
5.2.2 Pengujian 2: Pengujian Model OCR.....	73
5.2.3 Pengujian 3: Pengujian Sistem.....	81
5.2.4 Pengujian 4: Pengujian Notifikasi.....	86
5.2.5 Pengujian 5: User Acceptance Test (UAT).....	86
5.3 Analisis Hasil Pengujian.....	88
5.3.1. Hasil analisis dari pengujian yang pertama (Learning Rate).....	88
5.3.2. Hasil analisis dari pengujian yang kedua (Learning Rate).....	88
5.3.3. Hasil analisis dari pengujian yang ketiga (Batch Size).....	89
5.3.4. Hasil analisis dari pengujian yang keempat (epoch).....	89
5.3.5. Hasil analisis dari pengujian yang kelima (Optical Character Recognition).....	90
5.3.6. Hasil analisis dari pengujian yang keenam (Pengujian Sistem).....	91
5.3.7. Uji Validitas Kuesioner User Acceptance Test (UAT).....	95
5.4 Kesimpulan dan Ringkasan CD-5.....	97
DAFTAR PUSTAKA.....	98
LAMPIRAN CD-1.....	101
LAMPIRAN CD-2.....	105
LAMPIRAN CD-3.....	106
LAMPIRAN CD-4.....	107
LAMPIRAN CD-5.....	110

DAFTAR GAMBAR

Gambar 3.1 ilustrasi oleh Redmon et al.....	22
Gambar 3.2 rumus confidence.....	22
Gambar 3.3 rumus Intersection of Union.....	23
Gambar 3.4 arsitektur YOLOv7.....	23
Gambar 3.5 arsitektur YOLOv7.....	24
Gambar 3.6 ilustrasi monaserver.....	34
Gambar 3.7 ilustrasi Fungsi sistem.....	35
Gambar 3.8 Use Case Diagram.....	35
Gambar 3.9 Diagram blok sistem.....	38
Gambar 3.10 Flowchart solusi sistem.....	40
Gambar 3.11 Diagram blok keseluruhan sistem.....	41
Gambar 3.12 Activity Diagram.....	42
Gambar 3.13 Sequence Diagram.....	43
Gambar 3.14 Car Detection Test.....	44
Gambar 3.15 Car Identification.....	44
Gambar 3.16 Tampilan UI sederhana pada website.....	47
Gambar 3.17 Tampilan UI indikasi pelanggaran.....	47
Gambar 4.1 Diagram block.....	49
Gambar 4.2 proses anotasi gambar yang akan di training.....	49
Gambar 4.3 Diagram alur pendekripsi.....	51
Gambar 4.4 source code untuk mendekripsi pelanggaran.....	52
Gambar 4.5 alur kerja sub-sistem.....	53
Gambar 4.6 Diagram Alur Sistem.....	54
Gambar 4.7 Foto awal mobil dengan plat nomor.....	55
Gambar 4.8 Foto mobil dengan plat nomor yang sudah dipotong.....	55
Gambar 4.9 Foto mobil dengan plat nomor yang sudah di invert color.....	55
Gambar 4.10 Foto halaman landing page dari website.....	57
Gambar 4.11 Foto halaman main page dari website.....	57
Gambar 4.12 Foto halaman landing page dari website.....	58
Gambar 5.1 nilai rata-rata pada rasio data 58% : 42%.....	61
Gambar 5.2 grafik nilai rata-rata pada rasio data 58% : 42%.....	61
Gambar 5.3 nilai rata-rata pada rasio data 68% : 32%.....	62
Gambar 5.4 grafik nilai rata-rata pada rasio data 68% : 32%.....	62
Gambar 5.5 nilai rata-rata pada rasio data 78% : 22%.....	63
Gambar 5.6 grafik nilai rata-rata pada rasio data 78% : 22%.....	63
Gambar 5.7 nilai rata-rata pada rasio data 88% : 12%.....	64
Gambar 5.8 grafik nilai rata-rata pada rasio data 88% : 12%.....	64
Gambar 5.9 nilai rata-rata pada rasio data 98% : 2%.....	65
Gambar 5.10 grafik nilai rata-rata pada rasio data 98% : 2%.....	65
Gambar 5.11 nilai rata-rata pada pengujian learning rate dengan nilai 0,001.....	66
Gambar 5.12 grafik nilai rata-rata pada pengujian learning rate dengan nilai 0,001.....	66
Gambar 5.13 nilai rata-rata pada pengujian learning rate dengan nilai 0,0001.....	67

Gambar 5.14 grafik nilai rata-rata pada pengujian learning rate dengan nilai 0,0001.....	67
Gambar 5.15 nilai rata-rata pada pengujian learning rate dengan nilai 0,00001.....	68
Gambar 5.16 grafik nilai rata-rata pada pengujian learning rate dengan nilai 0,00001.....	68
Gambar 5.17 nilai rata-rata pada pengujian learning rate dengan nilai 0,00001.....	69
Gambar 5.18 grafik nilai rata-rata pada pengujian learning rate dengan nilai 0,00001.....	69
Gambar 5.19 nilai rata-rata pada pengujian batch size dengan nilai batch size 2.....	70
Gambar 5.20 grafik nilai rata-rata pada pengujian batch size dengan nilai batch size 2.....	70
Gambar 5.21 nilai rata-rata pada pengujian batch size dengan nilai batch size 4.....	71
Gambar 5.22 grafik nilai rata-rata pada pengujian batch size dengan nilai batch size 4.....	71
Gambar 5.23 nilai rata-rata pada pengujian batch size dengan nilai batch size 8.....	72
Gambar 5.24 grafik nilai rata-rata pada pengujian batch size dengan nilai batch size 8.....	72
Gambar 5.25 nilai rata-rata pada pengujian epoch dengan nilai epoch 100.....	73
Gambar 5.26 grafik nilai rata-rata pada pengujian epoch dengan nilai epoch 100.....	73
Gambar 5.27 nilai rata-rata pada pengujian epoch dengan nilai epoch 200.....	74
Gambar 5.28 grafik nilai rata-rata pada pengujian epoch dengan nilai epoch 200.....	74
Gambar 5.29 nilai rata-rata pada pengujian epoch dengan nilai epoch 300.....	75
Gambar 5.30 nilai rata-rata pada pengujian epoch dengan nilai epoch 300.....	75
Gambar 5.31 Full code easyocr.....	76
Gambar 5.32 sample foto pertama.....	77
Gambar 5.33 hasil foto kedua.....	77
Gambar 5.34 sample foto kedua.....	77
Gambar 5.35 hasil foto kedua.....	78
Gambar 5.36 sample foto ketiga.....	78
Gambar 5.37 hasil foto ketiga.....	79
Gambar 5.38 full code pytesseract.....	80
Gambar 5.39 sample foto pertama.....	81
Gambar 5.40 hasil foto pertama.....	81
Gambar 5.41 sample foto kedua.....	81
Gambar 5.42 hasil foto kedua.....	82
Gambar 5.43 sample foto ketiga.....	82
Gambar 5.44 hasil sample foto ketiga.....	82
Gambar 5.45 diagram batang pada pengujian rasio data.....	90
Gambar 5.46 diagram batang pada pengujian learning rate.....	91
Gambar 5.47 diagram batang pada pengujian batch size.....	91
Gambar 6.1 manual book DJI mavic pro.....	112
Gambar 6.2 DJI mavic pro aircraft profile.....	113
Gambar 6.3 DJI mavic pro controller specification.....	114
Gambar 6.4 DJI mavic pro camera profile.....	115
Gambar 6.5 DJI mavic pro gimbal profile.....	116
Gambar 6.6 DJI mavic pro appendix specification.....	117
Gambar 6.7 DJI mavic pro appendix specification.....	118
Gambar 6.8 google collab receipt.....	118
Gambar 6.9 google drive receipt.....	119

DAFTAR TABEL

Tabel 1.1 Tabel Perbandingan Solusi.....	9
Tabel 2.1 Perbedaan waktu yang diperlukan dalam pemrosesan gambar.....	15
Tabel 2.2 Verifikasi Spesifikasi 1.....	16
Tabel 2.3 Verifikasi Spesifikasi 2.....	18
Tabel 2.4 Verifikasi Spesifikasi 3.....	19
Tabel 2.5 Verifikasi Spesifikasi 3.....	20
Tabel 2.6 Verifikasi Spesifikasi 5.....	20
Tabel 3.1 Tabel Kriteria.....	35
Tabel 3.2 Analisis Konsep.....	36
Tabel 3.3 UU yang terkait dengan pelaksanaan sistem.....	37
Tabel 3.4 Pengujian Algoritma.....	43
Tabel 3.5 Perbandingan Pengujian Algoritma.....	44
Tabel 3.6 Pilihan Drone.....	45
Tabel 3.7 Pilihan perangkat komputer.....	46
Tabel 3.8 Gantt chart.....	48
Tabel 4.1 pengujian model yolo.....	50
Tabel 5.1 nilai rata-rata pada rasio data 58% : 42%.....	60
Tabel 5.2 nilai rata-rata pada rasio data 68% : 32%.....	61
Tabel 5.3 nilai rata-rata pada rasio data 78% : 22%.....	62
Tabel 5.4 nilai rata-rata pada rasio data 88% : 12%.....	63
Tabel 5.5 nilai rata-rata pada rasio data 98% : 2%.....	64
Tabel 5.6 nilai rata-rata pada pengujian learning rate dengan nilai 0,001.....	65
Tabel 5.7 nilai rata-rata pada pengujian learning rate dengan nilai 0,0001.....	66
Tabel 5.8 nilai rata-rata pada pengujian learning rate dengan nilai 0,00001.....	67
Tabel 5.9 nilai rata-rata pada pengujian learning rate dengan nilai 0,00001.....	68
Tabel 5.10 nilai rata-rata pada pengujian batch size dengan nilai batch size 2.....	70
Tabel 5.11 nilai rata-rata pada pengujian batch size dengan nilai batch size 4.....	70
Tabel 5.12 nilai rata-rata pada pengujian batch size dengan nilai batch size 8.....	71
Tabel 5.13 nilai rata-rata pada pengujian epoch dengan nilai epoch 100.....	73
Tabel 5.14 nilai rata-rata pada pengujian epoch dengan nilai epoch 200.....	74
Tabel 5.15 nilai rata-rata pada pengujian epoch dengan nilai epoch 300.....	74
Tabel 5.16 hasil pengujian pada angle 45 derajat.....	83
Tabel 5.17 hasil pengujian pada angle 60 derajat.....	83
Tabel 5.18 hasil pengujian pada angle 90 derajat.....	83
Tabel 5.19 hasil pengujian pada angle 45 derajat.....	84
Tabel 5.20 hasil pengujian pada angle 60 derajat.....	84
Tabel 5.21 hasil pengujian pada angle 90 derajat.....	84
Tabel 5.22 hasil pengujian pada angle 45 derajat.....	85
Tabel 5.23 hasil pengujian pada angle 60 derajat.....	85
Tabel 5.24 hasil pengujian pada angle 90 derajat.....	85
Tabel 5.25 hasil pengujian pada kecepatan 0,7-1,1 km/j.....	86
Tabel 5.26 hasil pengujian pada kecepatan 1,4-1,8 km/j.....	86

Tabel 5.27 hasil pengujian pada kecepatan 3,2-4 km/j.....	86
Tabel 5.28 hasil pengujian pada intensitas cahaya.....	87
Tabel 5.29 hasil pengujian pada intensitas cahaya.....	87
Tabel 5.30 hasil pengujian pada intensitas cahaya.....	87
Tabel 5.31 hasil pengujian pada notifikasi.....	88
Tabel 5.32 keterangan dan bobot nilai kuisioner.....	89
Tabel 5.33 persentase hasil dari jawaban kuisioner.....	89
Tabel 5.34 hasil pengujian menggunakan easyocr.....	92
Tabel 5.35 hasil pengujian menggunakan pytesseract.....	92
Tabel 5.36 hasil pengujian pada ketinggian 3m.....	92
Tabel 5.37 hasil pengujian pada ketinggian 5m.....	93
Tabel 5.38 hasil pengujian pada ketinggian 7m.....	94
Tabel 5.39 hasil pengujian pada kecepatan.....	94
Tabel 5.40 hasil pengujian pada lux meter.....	95
Tabel 5.41 hasil pengujian pada lux meter.....	96

DAFTAR SINGKATAN

FTE	: Fakultas Teknik Elektro
UAV	: Unmanned Aerial Vehicle
Prodi	: Program Studi
NIM	: Nomor Induk Mahasiswa
NIP	: Nomor Induk Pegawai
CCTV	: Closed Circuit Television
ETLE	: Electronic Traffic Law Enforcement
OCR	: Optical Character Recognition
CPU	: Central Processing Unit
GPU	: Graphic Processing Unit
CNN	: Convolutional Neural Network
IOU	: Intersection of Union
AP	: Average Precision

BAB 1

USULAN GAGASAN

1.1 Latar Belakang Masalah

Bersamaan dengan adanya perkembangan dari teknologi yang semakin maju, seluruh aspek dalam kehidupan pun turut berkembang. Salah satunya yaitu pada bidang penindakan hukum. Seperti pada belakangan ini, pihak kepolisian indonesia sudah mulai menerapkan perkembangan teknologi ini dengan menggunakan teknologi kamera cctv yang banyak berada di lampu merah. Akan tetapi belakangan ini juga mulai menguji coba teknologi yang baru dengan menggunakan drone, sebagai pembantu atau bahkan pengganti dari cctv [1].

Disaat yang bersamaan, sistem yang tersedia saat ini masih bergantung sekali terhadap banyaknya tenaga kerja manusia untuk menjalankan dan mengoperasikan sistem tersebut. Pengambilan gambar dan video menggunakan cctv juga masih menghadapi berbagai masalah yang cukup serius, dimana terkadang plat nomor tidak terlihat jelas dikarenakan resolusi kamera cctv masih tergolong rendah. Memang, untuk pengambilan gambar dan video sudah menggunakan cctv akan tetapi proses identifikasi apabila pengendara tersebut melakukan pelanggaran atau tidak masih dilakukan secara manual oleh tenaga kerja manusia yang berada dibelakang layar.

Drone atau *unmanned aerial vehicle* (UAV) merupakan pesawat tanpa pilot. Pesawat ini dikendalikan secara otomatis melalui program komputer yang dirancang, atau melalui kendali jarak jauh dari pilot yang terdapat di dataran atau di kendaraan lainnya. Sang pilot bertugas untuk mengendalikan drone dari jauh melalui remote control yang sudah terkoneksi dengan UAV ini. Alat ini dilengkapi dengan item kamera yang dapat membuat pengguna (pilot) dapat memantau dan merekam dari ketinggian. Drone ini digerakkan melalui tenaga baterai yang terpasang di dalamnya [2].

Pada penelitian yang kami lakukan, kami menggunakan pengendara dan kendaraan bermotor sebagai objek yang dikumpulkan ke dalam dataset. Dari program yang kami buat, ia akan mampu untuk mendeteksi berbagai bentuk pelanggaran dengan menggunakan teknologi *deep learning*. program akan dilatih menggunakan metode pelatihan dataset atau *Datasets Train*. Digunakannya program ini, bertujuan untuk mempermudah proses identifikasi terhadap suatu pelanggaran yang dilakukan oleh

pengendara, dikarenakan sampai saat ini proses tersebut masih dilakukan secara manual oleh tenaga dan mata manusia yang sangat memungkinkan terjadinya kesalahan yang disengaja ataupun tidak dalam proses mengidentifikasi pelanggaran tersebut.

Tujuan proyek kami menggunakan drone sebagai alat pengambil gambar dan video yaitu untuk mendeteksi lebih banyak ruang yang tidak bisa ditangkap dan dilihat oleh cctv. Karena cctv masih mempunyai kekurangan dari segi mobilitas yang membuat cctv tidak mampu bergerak dari tempatnya, hanya bisa memutar kameranya saja. Tentunya banyak pengendara yang berusaha untuk menghindari deteksi dari kamera cctv tersebut. berbagai hal dilakukan pengendara untuk mencegah sistem, seperti berhenti di belakang mobil atau truk sehingga tidak terlihat oleh pantauan cctv, lalu menutupi sebagian dari plat nomor agar tidak bisa terkena tilang elektronik. Dikarenakan drone mampu bergerak secara bebas, dan bisa mendeteksi barisan belakang dari antrian kendaraan yang ada di lampu merah, sehingga mampu meminimalisir adanya pengendara yang mencoba untuk mengelabui sistem dengan cara-cara bersembunyi dan menutupi plat nomor.

Maka penggunaan drone ini diharapkan mampu menyelesaikan masalah-masalah yang dialami ketika menggunakan cctv. Drone saat ini sudah sangat canggih bahkan memiliki resolusi kamera yang tinggi, hal ini membuat proses pendekripsi terhadap pelanggaran yang dilakukan oleh program deep learning menjadi lebih akurat. Penggunaan drone ini dapat membantu lembaga penegak hukum untuk mencari orang yang mencoba untuk mengelabui sistem. Pendekripsi pelanggaran menggunakan drone juga mempunyai tantangan yang cukup banyak seperti seperti kualitas video, pergerakan drone, subjek yang ditangkap, lokasi penerbangan, dan ketinggian [3].

1.2 Informasi Pendukung Masalah

Dalam sebuah survei yang dilakukan oleh *Fact Checker* menunjukkan bahwa pelanggaran lalu lintas menyebabkan 80% (117.914) dari semua kematian kecelakaan jalan dalam satu atau sekitar 323 kematian setiap harinya. diantara kasus-kasus kriminal tersebut, hampir 36% yang tertunda pada tahap persidangan karena kurangnya bukti, argumen & penilaian, dan sekitar 61% menunggu persidangan karena kurangnya bukti substansial [4]. Dalam meningkatkan keselamatan harus didasarkan dengan peningkatan kebiasaan pengemudi, yang dapat dicapai dengan perangkat yang dapat

memantau pengemudi. Penelitian yang dilakukan menunjukkan bahwa pengemudi banyak terlibat dalam kecelakaan serius dimana mereka memiliki pelanggaran lalu lintas dengan jumlah kesalahan yang jelas [5]. Uji coba penggunaan *electronic traffic law enforcement* (ETLE) berbasis drone sudah mulai dilakukan oleh Polda Jawa Tengah. Mekanisme yang digunakan adalah dengan merekam pengendara yang dinilai melakukan pelanggaran lalu lintas, kemudian divalidasi sebelum pelanggaran tilang [6].

pelacakan target dengan drone kecil telah dilakukan dengan berbagai metode. mereka dapat dikategorikan sebagai pelacak visual, dan non-visual. pelacak visual oleh drone kecil menggunakan video, berbagai pelacak berbasis *deep learning* akan dibandingkan dengan model gerak kamera. Hasil pengujian dengan video nyata berbasis UAV menunjukkan bahwa objek kecil, sejumlah besar target, dan gerakan kamera menurunkan kinerja pelacakan bahkan menggunakan GPU kelas atas. Detektor dan pelacak objek berbasis *deep learning* memerlukan komputasi yang berat dengan data pelatihan yang masif, sehingga pemrosesan waktu nyata sering kali menjadi masalah untuk dipecahkan [7].

Dalam sebuah percobaan, drone terbang dengan kecepatan konstan 5,1 m/s pada ketinggian 150 m sambil merekam klip video sembilan target bergerak. Deteksi dan tingkat alarm palsu serta kurva karakteristik operasi penerima diperoleh, dan kecepatan drone dalam arah x dan y diperkirakan oleh vektor perpindahan. Tingkat deteksi rata-rata berkisar dari 90% hingga 97% sedangkan tingkat alarm palsu berkisar antara 0,06 hingga 0,5. Kesalahan akar kuadrat rata-rata dari kecepatan adalah 0,07 m/s ketika kerangka acuan diperbaiki, menunjukkan kekokohan metode yang diusulkan [8].

Sebagian besar sampel pelatihan berasal dari 578 video drone yang diperoleh dari layanan video populer. Kumpulan data pelatihan lengkap terdiri dari 51446 gambar yang diperkecil dari resolusi yang berbeda (mulai dari 640x480 hingga 4K) hingga resolusi 640x480, di mana 51445 gambar berisi total 52676 kotak pembatas drone dan 1 gambar negatif (tidak mengandung setiap UAV). Untuk tugas deteksi objek, sebagian besar gambar masukan adalah contoh negatif (tidak berisi kelas yang dimaksud), oleh karena itu menambahkan kumpulan data negatif tidak akan memberikan nilai tambah apapun ke model. Dalam dataset pelatihan, ada lebih banyak objek kecil daripada objek besar. Secara khusus: sekitar 40,8% objek berukuran kecil ($\text{area} < 1024$), 35,8% sedang ($1024 < \text{area} < 9216$), dan 23,4% berukuran besar ($\text{area} > 9216$) seperti yang ditentukan

oleh tantangan COCO. secara menyeluruh, tujuan dari kumpulan data set besar yang beragam adalah untuk menyajikan drone dari berbagai jenis, ukuran, skala, posisi, lingkungan, waktu pada hari itu, dll., sehingga memungkinkan berbagai representasi untuk melatih model deteksi objek [9].

UAV memiliki keunggulan kemampuan manuver yang tinggi, ukuran kecil, dan pengoperasian yang sensitif. Ini banyak digunakan dalam transportasi, militer, penelitian ilmiah, dan bidang lainnya. Perkembangan teknologi komputer dan teknologi pengenalan gambar telah membuat teknologi deteksi target bergerak semakin matang, dan penggunaan kendaraan udara tak berawak dapat membuat teknologi deteksi target bergerak ke tingkat yang lebih tinggi [10]. UAV akan terbang di atas area tersebut, mencoba mengumpulkan data sebanyak mungkin, dan akan mengevaluasi tingkat cakupan mobil, persentase kejadian abnormal (kemacetan, dan pelanggaran), dan durasi kejadian yang terdeteksi [11].

1.3 Analisis Umum

1.3.1. Aspek Ekonomi

Perancangan sistem drone menggunakan program *deep learning* ini akan mempertimbangkan aspek dari segi ekonomis. Setelah dihitung biaya yang cukup besar hanya akan dikeluarkan di awal sebagai modal untuk drone, sisanya hanya biaya *maintenance* alat tersebut, seperti pembelian baterai cadangan atau mengganti beberapa komponen yang memang sudah harus diganti.

Lalu program *deep learning* ini juga membantu meminimalisir pengeluaran untuk membayar upah operator yang sekarang bekerja sebagai tim pengecek pada saat proses identifikasi pelanggaran, dikarenakan dengan menggunakan *deep learning* proses tersebut sudah berjalan secara autonomous dan tanpa adanya banyak orang yang mengawasi dan mengoperasikannya.

1.3.2. Aspek Keberlanjutan (sustainability)

Melihat dari perkembangan teknologi pada saat ini, produk yang kami akan kerjakan memiliki masa depan yang sangat menjanjikan, karena teknologi dan program yang kami gunakan di dalam produk merupakan teknologi yang cukup mutakhir sehingga umur dari teknologi dan program tersebut tidak akan banyak berubah selama bertahun-tahun kedepan.

Sehingga kami pun akan selalu mengeluarkan versi terbaru dari program yang kami buat. Seperti pada versi pertama (1.0) sistem kami ini akan mendeteksi minimal 3 jenis pelanggaran yang dilakukan oleh pengendara. Lalu nanti kami mengeluarkan juga versi berikutnya yang akan memberikan update terhadap sistem yang kami buat, seperti meningkatkan akurasi dari proses identifikasi, memperbanyak jenis pelanggaran yang bisa dideteksi, dan kecepatan dari proses identifikasi sistem tersebut.

1.3.3. Aspek Manufakturabilitas (manufacture)

Pada proyek kami ini juga tidak sulit untuk menemukan produk pendukung seperti drone dan kelengkapannya. Secara produksinya pun sudah sangat massif, karena banyaknya masyarakat Indonesia yang antusias terhadap teknologi drone ini. Bahkan sudah banyak sekali toko elektronik yang menjual berbagai macam komponen yang ada pada drone, seperti baling-baling cadangan, baterai cadangan dan hal pendukung lainnya, sehingga alat dan perangkat pendukung pada proyek kami ini akan sangat mudah untuk ditemukan.

1.3.4. Aspek Penggunaan (usability)

Proyek yang akan dikerjakan akan memperhatikan kemudahan pengguna. Dikarenakan pengguna hanya membutuhkan skill dasar menerbangkan drone saja, selebihnya program lah yang akan melakukan analisis dan proses identifikasi pelanggaran. Dengan begitu pekerjaan yang dilakukan oleh tenaga kerja manusia akan lebih sedikit dan mudah.

1.4 Kebutuhan yang Harus Dipenuhi

Dari masalah, latar belakang dan analisis yang telah kami jelaskan, dibutuhkan sistem operasi penggunaan *deep learning* sebagai alat pemrosesan informasi yang ditangkap oleh drone berkaitan dengan pelanggaran lalu lintas. Berikut beberapa kebutuhan yang harus dipenuhi agar produk yang kami buat dapat direalisasikan. Diantaranya adalah sebagai berikut :

1. Sebuah drone dengan kamera yang memiliki resolusi lebih baik dibandingkan cctv.
2. Sebuah drone yang dapat secara langsung dihubungkan kepada perangkat komputer.
3. Sebuah sistem yang dapat mendeteksi jenis pelanggaran lalu lintas secara cepat dan akurat.

4. Sistem program yang dapat mendeteksi bentuk manusia dan kendaraan dan mampu mengkategorikan jenis pelanggaran yang dilakukan pengendara kendaraan bermotor.
5. Drone harus dapat dikendalikan secara jarak yang relatif jauh demi keamanan pengendali dan tidak mengganggu frekuensi di sekitar lokasi penggerjaan.

1.5 Solusi Sistem yang Diusulkan

1.5.1 Karakteristik Produk

Fitur Utama:

1. Perangkat ini akan menggunakan proses *deep learning* untuk menganalisa beberapa bentuk pelanggaran lalu lintas.
2. Program akan mendeteksi dan membaca plat nomor dari kendaraan yang terindikasi melakukan pelanggaran.

Fitur Dasar:

1. Penggunaan *deep Learning* akan mendeteksi berbagai bentuk pelanggaran lalu lintas dengan lebih detail dan mampu bekerja secara *autonomous*.
2. Penggunaan kamera cctv sebagai alat untuk mengambil gambar untuk dianalisa kembali oleh program *deep learning* untuk mengidentifikasi berbagai bentuk pelanggaran.
3. Penggunaan drone juga bisa mencakup area yang tertutup ketika masih menggunakan kamera cctv.

Fitur Tambahan:

1. Proses mengidentifikasi masalah pun lebih fleksibel apabila menggunakan drone dikarenakan bisa mendeteksi pada ketinggian tertentu didukung ketajaman kamera pada drone.
2. Proses penilangan bisa langsung terhubung dengan fasilitas ETLE yang membuat sistem ini mampu bekerja dengan waktu yang singkat.

Sifat solusi yang diharapkan:

1. Diharapkan produk dapat dipergunakan dengan mudah oleh pihak yang berwajib dalam melakukan proses penilangan

2. Produk juga diharapkan mampu membuat kinerja dari pihak berwajib menjadi lebih efisien dan akurat.
3. Tidak memerlukan penglihatan dan ketelitian manusia secara manual untuk mengidentifikasi berbagai bentuk pelanggaran lalu lintas.

1.5.1.1 Produk A

Sistem ini akan diimplementasikan pada drone yaitu untuk mendeteksi pelanggaran lalu lintas yang tidak terlihat oleh kamera cctv, karena kamera cctv hanya bisa berputar di satu tempat dan hanya bisa berputar 360° yang dimana pelanggaran terhadap pengendara lalu lintas tidak terjangkau kamera cctv. Maka itu kita menggunakan drone sebagai alat bantu untuk mempermudah pencarian terhadap pelanggaran yang dilakukan oleh pengendara bermotor.

Sistem yang kita tawarkan yaitu melacak pelanggaran dengan menggunakan sistem *Deep Learning* yang dimana pihak berwajib tidak usah menindak secara manual terhadap pelanggaran lalu lintas bermotor sehingga pelanggaran tersebut sudah dicatat di sebuah aplikasi yang sudah dirancang untuk pelanggar lalu lintas.

Stakeholder yang terlibat :

- Izin perusahaan
- Pihak industri yang bersangkutan

1.5.1.2 Produk B

Pada produk yang kedua kami masih menggunakan kamera cctv yang ada pada saat ini, kami akan tetap menggunakan program *deep learning* untuk proses pengidentifikasi terhadap bentuk pelanggaran lalu lintas yang dilakukan oleh pengendara kendaraan bermotor.

pada produk yang kedua ini terdapat kekurangan, dimana jangkauan dari penglihatan cctv itu sendiri pun tidak seluas dah se-fleksibel drone. Begitu juga dengan resolusi dari gambar yang dihasilkan oleh kamera cctv tidak memiliki resolusi yang cukup tinggi, sehingga proses identifikasi akan sedikit terganggu dan lambat.

Stakeholder yang terlibat :

- Izin perusahaan
- Pihak industri yang bersangkutan

1.5.1.3 Produk C

Pada produk yang ketiga kami akan menggunakan drone sebagai alat untuk mengambil gambar. Akan tetapi kami tidak akan menggunakan program *deep learning* sebagai alat untuk mengidentifikasi pelanggaran yang dilakukan oleh pengendara kendaraan bermotor. Sehingga dalam proses pengidentifikasi dan penilangan masih menggunakan penglihatan dan keahlian dari tenaga manusia secara manual.

Stakeholder yang terlibat :

- Izin perusahaan
- Pihak industri yang bersangkutan

1.5.1.4 Produk D

Pada produk yang keempat akan kembali menggunakan cara manual sepenuhnya. Menggunakan tenaga dari petugas kepolisian untuk melakukan patroli keliling pada setiap lampu merah, melakukan pengecekan menggunakan kamera dari perangkat kecil seperti kamera pada *smartphone* ataupun kamera pocket. Pada produk yang keempat ini kami tidak akan menggunakan teknologi *deep learning*, karena akan sangat membuang waktu saat melakukan pengecekan apakah terjadi pelanggaran lalu lintas atau tidak, akan jauh lebih efektif apabila petugas yang langsung menganalisa apakah pengendara tersebut melakukan pelanggaran lalu lintas atau tidak.

Stakeholder yang terlibat :

- Izin perusahaan
- Pihak industri yang bersangkutan

1.5.2 Skenario Penggunaan

1.5.2.1 Skema A

1. Produk ini akan digunakan oleh pihak berwajib dan institusi kepolisian yang bertanggung jawab terhadap penindakan pelanggaran lalu lintas
2. Produk ini akan digunakan di beberapa titik lampu merah yang terindikasi banyaknya pelanggaran yang dilakukan oleh pengendara kendaraan bermotor
3. Produk ini akan digunakan disaat cctv pada lampu merah tidak dapat bekerja secara optimal

1.5.2.2 Skema B

1. Produk ini akan digunakan oleh pihak berwajib dan institusi kepolisian yang bertanggung jawab terhadap penindakan pelanggaran lalu lintas
2. Produk ini akan digunakan di beberapa titik lampu merah yang terindikasi banyaknya pelanggaran yang dilakukan oleh pengendara kendaraan bermotor
3. Solusi ini akan digunakan pada saat terjadi kendala pada drone, seperti cuaca yang tidak mendukung dan kerusakan pada drone

1.5.2.3 Skema C

1. Produk ini akan digunakan oleh pihak berwajib dan institusi kepolisian yang bertanggung jawab terhadap penindakan pelanggaran lalu lintas
2. Produk ini akan digunakan di beberapa titik lampu merah yang terindikasi banyaknya pelanggaran yang dilakukan oleh pengendara kendaraan bermotor
3. Solusi ini akan digunakan apabila adanya kegagalan ketika menghubungkan program *deep learning* dengan drone yang digunakan oleh pihak kepolisian.

1.5.2.4 Skema D

1. Produk ini akan digunakan oleh pihak berwajib dan institusi kepolisian yang bertanggung jawab terhadap penindakan pelanggaran lalu lintas
2. Produk ini akan digunakan di beberapa titik lampu merah yang terindikasi banyaknya pelanggaran yang dilakukan oleh pengendara kendaraan bermotor
3. Solusi ini akan digunakan apabila tidak adanya biaya dan kesulitan untuk mendapatkan alat pendukung yaitu drone itu sendiri.
4. Solusi ini juga akan digunakan ketika proses pembuatan program *deep learning* tidak tercapai atau tidak selesai.

Tabel 1.1 Tabel Perbandingan Solusi

Alternatif Solusi	Aspek Ekonomi	Aspek Keberlanjutan	Aspek Manufakturabilitas	Aspek Penggunaan
Solusi 1	Kelebihan: + Hasil yang didapatkan dengan	Kelebihan: + Alat akan mempermudah apabila	Kelebihan: + Ketersediaan alat yang cukup banyak dan	Kelebihan: + Hanya dengan menguasai

Alternatif Solusi	Aspek Ekonomi	Aspek Keberlanjutan	Aspek Manufakturabilitas	Aspek Penggunaan
	<p>mengeluarkan dana anggaran sebanyak itu tentunya akan lebih maksimal.</p> <p>Kekurangan: Anggaran yang dikeluarkan cukup banyak dikarenakan adanya biaya pembelian drone dan perawatan dari drone tersebut</p>	<p>kedepannya jumlah kendaraan terus bertambah dan kondisi jalan menjadi semakin macet, karena jangkauan penglihatan yang lebih luas. Begitu pula dengan keakuratan dari alat ini akan semakin tajam dan akurat seiring berjalannya waktu dan banyaknya proses yang sudah dilakukan.</p> <p>Kekurangan: Perawatan dari alat yang digunakan cukup rumit dan menghabiskan waktu untuk tetap mengupdate program yang telah dibuat.</p>	<p>mudah didapatkan baik secara langsung ataupun dengan cara mencari alat pada situs <i>e-commerce</i>. Lalu banyaknya variasi dari alat yang digunakan membuat solusi ini menjadi lebih bisa untuk dikembangkan.</p> <p>Kekurangan: Meskipun mudah didapatkan, akan tetapi bagian dari alat-alat tersebut kadang habis, karena banyaknya pengguna drone di luar sana.</p>	<p>teknik dasar menerbangkan pesawat drone, user sudah bisa dipastikan mampu untuk menggunakan produk ini, tentunya skill tersebut juga tidak sulit untuk dipelajari.</p> <p>Kekurangan: Pengguna akan sedikit kesulitan ketika mempelajari bagaimana manuver-manuver drone yang lebih <i>advance</i>.</p>
Solusi 2	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Dikarenakan menggunakan kamera cctv, solusi ini cukup murah meskipun harus tetap melakukan 	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Hanya dengan tetap mengupdate software dari produk ini, maka produk akan tetap bisa dioperasikan, dan perawatan 	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Tentu lebih mudah mendapatkan perangkat yang digunakan pada solusi ini. Banyaknya perangkat yang tersedia pada 	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Penggunaan dari produk ini bisa dilakukan secara sangat fleksibel tanpa adanya manusia yang bekerja

Alternatif Solusi	Aspek Ekonomi	Aspek Keberlanjutan	Aspek Manufakturabilitas	Aspek Penggunaan
	<p>perawatan pada kamera cctv.</p> <p>Kekurangan: Dengan biaya yang murah, tentunya fitur dan kemampuan dari pendeksiakan akan kurang efektif dan maksimal.</p>	<p>dari alatnya pun tidak begitu sulit.</p> <p>Kekurangan: Seiring dengan bertambahnya jumlah kendaraan di jalan raya, maka penglihatan dari alat ini pun terbatas, tidak bisa bekerja secara optimal seperti ketika awal dibuat.</p>	<p>berbagai platform pun semakin mempermudah untuk mendapatkan alat-alat yang digunakan.</p> <p>Kekurangan: Semakin mudah mendapatkan kan alat-alat yang digunakan, akan semakin rumit juga untuk memilah alat mana saja yang memiliki kualitas bagus.</p>	<p>di lapangan. Pengguna hanya perlu memonitor gambar atau video hasil tangkapan dari alat tersebut dari balik layar komputer atau perangkat untuk langsung dimasukan ke dalam program dan melakukan analisis terhadap hasil dari gambar tangkapan cctv yang ada di lapangan.</p> <p>Kekurangan: Operator harus memasukan hasil dari tangkapan kamera cctv secara manual ke dalam program, dan untuk mengontrol cctv nya pun akan ada delay karena jarak yang cukup jauh antara tempat</p>

Alternatif Solusi	Aspek Ekonomi	Aspek Keberlanjutan	Aspek Manufakturabilitas	Aspek Penggunaan
				monitoring dan lapangan yang diamati.
Solusi 3	<p>Kelebihan:</p> <ul style="list-style-type: none"> + dari anggaran yang dikeluarkan akan seimbang, tidak perlu mengeluarkan biaya untuk pembuatan program pendekripsi otomatis karena alat yang digunakan hanya drone. <p>Kekurangan: Anggaran pada solusi yang ketiga tidak jauh dengan solusi pertama, karena tetap menggunakan drone yang perlu dilakukan perawatan berkala.</p>	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Produk yang digunakan bersifat praktis, mudah dibawa oleh pengguna drone, dan saat pengoprasiannya drone akan lebih fleksibel sebab pihak kepolisian lebih mudah dalam menganalisis gambar informasi yang didapat dari berbagai sudut kamera. <p>Kekurangan: Maraknya gangguan dalam penerbangan drone, seperti gangguan hewan terbang dan manusia.</p>	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Produksi drone dan perlengkapannya sudah banyak terjual karena antusiasnya warga negara Indonesia dalam mengikuti perkembangan teknologi drone. <p>Kekurangan: Hanya dapat menerbangkan drone dengan durasi sesuai baterai yang tersedia, tapi hal ini dapat diselesaikan jika drone yang digunakan mempunyai slot baterai yang dapat diganti dengan baterai terisi penuh.</p>	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Ketika sudah diterbangkan di udara, bisa menggunakan sistem <i>screenshot</i> agar pihak kepolisian dapat melihat kembali dan menganalisis hasil dari gambar yang diambil menggunakan sistem tersebut. <p>Kekurangan: Kemahiran yang dipelukan dalam mengontrol drone ketika sudah diterbangkan di udara, memerlukan tingkat pemahaman mengendalikan drone yang tinggi.</p>
Solusi 4	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Solusi inilah yang mempunyai aspek ekonomi 	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Akan sangat mudah untuk menjaga keberlangsungan dari solusi ini, 	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Tidak ada kelebihan pada aspek ini karena tidak menggunakan 	<p>Kelebihan:</p> <ul style="list-style-type: none"> + Sangat amat mudah untuk dilakukan karena termasuk

Alternatif Solusi	Aspek Ekonomi	Aspek Keberlanjutan	Aspek Manufakturabilitas	Aspek Penggunaan
	<p>yang paling efektif, dikarenakan masih menggunakan cara manual sehingga tidak ada biaya yang dikeluarkan.</p> <p>Kekurangan: Pada solusi ini tidak ada kekurangan dalam aspek ekonomi, karena memang biaya yang dikeluarkan sangatlah minim. Biaya yang dikeluarkan hanya untuk gaji petugas dan biaya Transportasi.</p>	<p>karena hanya perlu melakukan perubahan tenaga kerja secara rutin saja.</p> <p>Kekurangan: Tidak dapat dipastikan apakah akan tetap ada para tenaga kerja yang tertarik dengan pekerjaan ini, secara sistem pun solusi yang ini masih sangat tradisional dan mudah untuk dimanipulasi.</p>	<p>berbagai macam alat-alat seperti pada solusi sebelumnya.</p> <p>Kekurangan: Tidak ada kekurangan pada aspek ini karena tidak menggunakan berbagai macam alat-alat seperti pada solusi sebelumnya.</p>	<p>dalam kegiatan sehari hari yang biasa dilakukan oleh pihak petugas.</p> <p>Kekurangan: Tentunya melelahkan bagi petugas untuk terus berkeliling mengitari pemberhentian lampu merah.</p>

1.6 Kesimpulan dan Ringkasan CD-1

Permasalahan yang ada ketika melakukan penertiban lalu lintas pada jalan raya ini dapat diselesaikan dengan menggunakan solusi sistem yang penulis pilih. Sistem yang berbasis *deep learning* dan dibantu dengan teknologi drone dinilai cukup membantu dan memperbaiki sistem yang sudah ada pada saat ini untuk memudahkan pihak yang berwajib dalam menindak setiap pelanggaran yang dilakukan pengguna kendaraan bermotor.

BAB 2

DESAIN KONSEP SOLUSI

2.1 Spesifikasi Produk

Berdasarkan CD-1, kebutuhan produknya:

1. Alat mempunyai sistem *deep learning* yang dapat mendeteksi pelanggaran lalu lintas.
2. Alat dapat membedakan objek kendaraan roda 2 dan 4 menggunakan kamera yang tersedia pada alat.
3. Alat yang dibutuhkan memiliki kamera dengan resolusi lebih dari 720p dan memiliki waktu terbang yang cukup di udara dengan mempunyai kapasitas baterai minimal 3000 mAh.
4. Kamera yang mampu merekam minimum 24 fps.
5. Alat dapat memindahkan data ke perangkat lain dan bertukar informasi secara *real time*.

Karakteristik produk:

Fitur utama:

- Dapat mendeteksi pelanggaran lalu lintas seperti tidak menggunakan helm dan berhenti di atas garis zebracross yang dilakukan pengendara.
- Dapat membaca digit dari plat nomor kendaraan yang terdeteksi melakukan pelanggaran lalu lintas.
- Output dari sistem dapat dimonitor dari jarak jauh secara *real time*.

Fitur Dasar:

- Sistem dapat mendeteksi pelanggaran lalu lintas sesuai dengan poin di atas.
- Sistem mampu membaca digit dari plat nomor kendaraan.

Fitur Tambahan:

- Sistem dapat terkoneksi dengan perangkat lain dan bertukar informasi berkaitan dengan hasil pelanggaran yang dilakukan pengendara.
- Hasil dari tangkapan gambar yang dilakukan oleh sistem mampu dilihat pada *website* yang sudah dibuat bersamaan dengan sistem.

2.1.1. Spesifikasi 1

Hasil dari penangkapan gambar dari sistem akan dikirim secara real-time kedalam aplikasi berbasis web melewati *private network*. Gambar tangkapan dari drone akan langsung dikirimkan ke sistem untuk diidentifikasi lebih lanjut mengenai pelanggaran yang dilakukan. Sistem tersebut akan langsung mengidentifikasi dan menetapkan apakah pengendara tersebut melakukan pelanggaran atau tidak.

2.1.2. Spesifikasi 2

Sistem dapat mengelompokkan jenis pelanggaran lalu lintas yang dilakukan oleh pengendara berdasarkan algoritma peraturan lalu lintas yang sudah dimasukan ke dalam sistem pendekripsi. jenis pelanggaran yang akan dideteksi merupakan pelanggaran yang marak dilakukan ketika berkendara. Pentingnya reaksi yang cepat dalam pendekripsi gambar, berpengaruh kepada data informasi. YOLO merupakan salah satu model machine learning yang paling populer di ranah deteksi objek, mendekripsi objek *real-time* dan sangat cepat, lebih baik, dan lebih kuat dibandingkan dengan metode lain. Akurasinya sangat tinggi dan waktu yang diperlukan YOLO untuk memproses gambar yaitu singkat [12].

Tabel 2.1 Perbedaan waktu yang diperlukan dalam pemrosesan gambar

Metode	Kecepatan Deteksi	
RCNN	0.05 FPS	20 s/ gambar
Fast-RCNN	0.5 FPS	2 s/gambar
Faster-RCNN	7 FPS	140 ms/gambar
YOLO	45 FPS	22 ms/gambar

2.1.3. Spesifikasi 3

Sistem dapat mendekripsi dan membaca digit plat nomor dari kendaraan yang digunakan oleh pengendara yang terindikasi melakukan pelanggaran lalu lintas. Tujuan dari program mampu membaca plat nomor tersebut agar lebih mudah untuk mengkategorikan kendaraan mana saja yang telah melakukan pelanggaran lalu lintas. Sistem ini akan bekerja setelah mendapatkan foto hasil *screenshot* ketika pengendara

terindikasi melakukan pelanggaran, sehingga tidak semua kendaraan akan dibaca plat nomornya.

2.1.4. Spesifikasi 4

Sistem dapat membedakan jenis kendaraan yang digunakan dan mampu membedakan jenis pelanggaran apa saja yang dilakukan oleh pengendara kendaraan bermotor, dengan melakukan klasifikasi terhadap kendaraan apa yang digunakan oleh pengendara. Kesulitan utama dalam membuat sistem pendekripsi yang kuat adalah melatih deteksi objek berasal dari berbagai variasi gambar objek milik kelas objek yang sama. Objek serupa dari kelas yang sama mungkin ada ukuran berbeda, bentuk berbeda karena titik pandang dll. Memiliki basis model yang rinci dapat membantu mengatasi hal ini dengan membuat model abstrak dengan menggunakan bagian-bagian dari objek yang diteliti. Tantangan serupa seperti tekstur, pencahayaan, latar belakang dan lain-lain meningkatkan kesulitan mendekripsi objek dalam sebuah gambar [13].

2.1.5. Spesifikasi 5

Hasil dari analisis terjadinya pelanggaran akan ditampilkan dengan menggunakan platform aplikasi berbasis web. Pelanggaran yang telah dikonfirmasi dan ditetapkan akan ditampilkan pada web tersebut. Aplikasi berbasis web tersebut dapat diakses dan digunakan oleh seluruh pengguna kendaraan bermotor yang kendaraanya telah terdaftar (kendaraan legal). Aplikasi tersebut juga dibuat dengan sangat *user friendly*.

2.2 Verifikasi

2.2.1 Verifikasi Spesifikasi 1

Tabel 2.2 Verifikasi Spesifikasi 1

Hal	Klasifikasi jenis pelanggaran dan deteksi kendaraan
Rincian	Pelanggaran lalu lintas yang dilakukan pengendara berdasarkan peraturan lalu lintas yang berada pada sistem pendekripsi.
Metode Pengujian	Memberi tampilan pelanggaran dari berbagai sudut pandang kamera

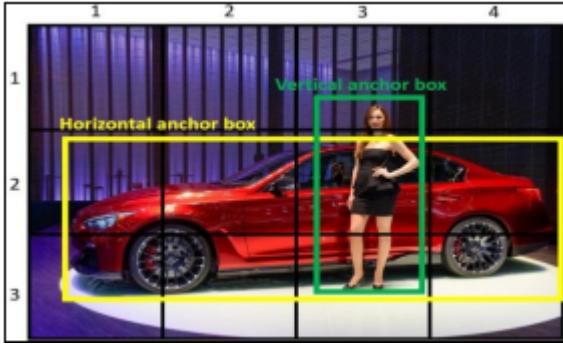
Hal	Klasifikasi jenis pelanggaran dan deteksi kendaraan
Prosedur Pengujian	<p>Sistem diberikan tampilan jenis pelanggaran lalu lintas sesuai dengan algoritma yang di input ke sistem mengenai peraturan lalu lintas. Secara rinci data yang dikerjakan dengan algoritma yaitu berupa kumpulan data jenis pelanggaran, jumlah jenis objek dan hasilnya akan dilaporkan ke data pelanggaran. Spesifikasi ini memerlukan Pengklasifikasi dengan metode <i>Haar Cascade</i> digunakan untuk mendeteksi objek. Fitur <i>anchor box</i> dapat membuat komputer sering menggunakan representasi perantara dari gambar tersebut bernama integral. Gambar integral terdiri dari unit-unit kecil yang merepresentasikan gambar yang diberikan. Gambar integral digunakan untuk deteksi objek dimana perhitungan ini membutuhkan waktu yang cepat dan hasil yang akurat. Gambar integral digunakan untuk menghitung nilai yang dihasilkan untuk area yang terdeteksi oleh fitur dari metode <i>Haar Cascade</i>. Nilai piksel yang dihitung berasal dari input gambar yang dilewati oleh fitur <i>Haar Cascade</i> pada saat itu. pemindaian untuk mendapatkan nilai fitur yang menjelaskan objek. nilai integral gambar dari fitur akan dihitung dengan menggunakan rumus berikut:</p> $s(x, y) = i(x, y) + s(x, y + 1) + s(x + 1, y) - s(x + 1, y + 1)$ <p>Pada pelanggaran di lampu lalu lintas zebra cross akan menggunakan data kendaraan yang sudah masuk pelatihan dengan menggunakan <i>Haar Cascade Classifiers</i> dengan uji video. Untuk mengukur tingkat akurasi data yang kita miliki terlatih kita akan menggunakan perhitungan sederhana seperti berikut:</p> $\frac{\text{jumlah kendaraan yang terdeteksi}}{\text{jumlah kendaraan yang berada pada area zebra cross}} \times 100\%$ <p>[14].</p>

2.2.2 Verifikasi spesifikasi 2

Tabel 2.3 Verifikasi Spesifikasi 2

Hal	Sistem pelanggaran kendaraan lalu lintas
Rincian	Membedakan jenis pelanggaran dan kendaraan yang

Hal	Sistem pelanggaran kendaraan lalu lintas
	digunakan pengendara.
Metode Pengujian	Memberi tampilan pelanggaran dari berbagai sudut pandang kamera
Prosedur Pengujian	<p>Sistem diberikan tampilan yang berlokasi di jalan raya dengan kendaraan yang cukup, untuk menguji seberapa akurat sistem tersebut dalam mendekripsi jenis pelanggaran dan kendaraan. Drone akan diluncurkan pada ketinggian, jarak, dan intensitas cahaya sebagai uji sample agar sistem dapat hasil deteksi yang <i>valid</i>. Untuk mengukur jarak antara kamera dan rintangan, hubungan matematis antara ketinggian Objek Nyata, Tinggi gambar, dan panjang fokus kamera telah digunakan.</p> $\frac{\text{Sensor height(mm)}}{\text{Focal length(mm)}} = \frac{\text{Object Dimensio(mm)}}{\text{Distance to Object (mm)}}$ <p>lalu, dimensi objek diperoleh dari</p> $OD = \frac{\text{Real Height of Object(mm)} * \text{Camera frame height (pixels)}}{\text{Image Height (pixels)}}$ <p>jadi, jika disatukan dua persamaan tersebut menghasilkan persamaan,</p> $\frac{\text{Real height of Object(mm)} * \text{Camera Frame height(pixels)} * f(\text{mm})}{\text{Image Height(pixels)} * \text{Sensor Height(mm)}}$ <p>dengan menggunakan prinsip dasar lensa ini, jarak rintangan dari kamera drone bisa lebih jauh dihitung [14]. Deteksi dilakukan dengan menggunakan OpenCV dan Numpy dengan mengkonversi gambar dari dataset sebagai pembelajaran YOLO, dan akan diproses dengan menampilkan gambar kotak persegi panjang(<i>anchor boxes</i>) bersama dengan tulisan penjelasan deteksinya. setiap sel kotak yang tersedia pada gambar merupakan tanggung jawab untuk mendekripsi objek. kotak tersebut ditunjukkan memiliki koordinat <i>x</i> dan <i>y</i> yang akan diprediksi relatif yang bisa diubah menjadi hasil prediksi deteksi. Jika pada suatu gambar memiliki 3 objek, konsep <i>anchor boxes</i> digunakan, ide dari konsep tersebut menambahkan dimensi ke label output dengan mendefinisikan terlebih dahulu beberapa <i>anchor boxes</i> dan dimensi dari <i>anchor boxes</i>. Dengan begitu, YOLO dapat menempatkan satu objek ke setiap <i>anchor boxes</i>.</p>

Hal	Sistem pelanggaran kendaraan lalu lintas
	 <p><i>Gambar 1 Contoh Anchor Boxes</i></p> <p>Dimensi dari <i>anchor boxes</i> akan di kalkulasi dari pelatihan dataset [15].</p>

2.2.3 Verifikasi spesifikasi 3

Tabel 2.4 Verifikasi Spesifikasi 3

Hal	Waktu beroperasi sistem
Rincian	Sistem dapat mendeteksi dan membaca digit plat nomor dari kendaraan yang digunakan oleh pengendara yang terindikasi melakukan pelanggaran lalu lintas.
Metode Pengujian	Memberi sampel foto plat nomor untuk di deteksi dan dibaca
Prosedur Pengujian	Sistem akan dihubungkan ke kamera, baik webcam maupun drone. dikarenakan masih pengujian, maka sistem akan diberikan beberapa sampel foto untuk dicoba apakah program berjalan dengan baik atau tidak. Setelah program dianggap mampu dan berhasil mendeteksi adanya plat nomor, maka program akan lanjut ke proses selanjutnya, membaca digit dari plat nomor tersebut. Lalu dilihat hasilnya apakah program tersebut sudah mampu membaca digit dari plat nomor kendaraan tersebut dengan tingkat keakuratan yang tinggi.

2.2.4 Verifikasi Spesifikasi 4

Tabel 2.5 Verifikasi Spesifikasi 3

Hal	Waktu beroperasi sistem
Rincian	sistem yang dapat memonitor tampilan, memberikan data, bertukar informasi secara <i>real time</i> dari drone
Metode Pengujian	Memberi jarak drone dengan laptop yang menyediakan website tersebut.
Prosedur Pengujian	<i>Website</i> dihubungkan dengan alat drone, dan melakukan sampel pengujian berupa jarak antara laptop dengan drone dan menganalisis objek tetapi membutuhkan waktu lama untuk mendeteksi objek kemungkinan besar tidak akan memenuhi secara <i>real time</i> dengan akurat. Oleh karena itu dimana deteksi dilakukan sebelum hasil yang ditentukan. Tetapi dengan menggunakan model struktur YOLO yang dirancang untuk bertujuan mendeteksi target dengan karakteristik <i>end-to-end</i> serta kecepatan kalkulasinya lebih cepat daripada kebanyakan jaringan dan memiliki kinerja realtime yang baik. Selain itu jaringan dapat mempertahankan tingkat akurasi yang tinggi dengan alasan berdasarkan dua keunggulan yang dianggap memenuhi persyaratan akurasi dan waktu nyata.

2.2.5 Verifikasi spesifikasi 5

Tabel 2.6 Verifikasi Spesifikasi 5

Hal	Waktu beroperasi sistem
Rincian	<i>Website</i> yang dapat menampilkan hasil data dari sistem pendekripsi berupa waktu, dan bukti pelanggaran.
Metode Pengujian	Memberi tampilan informasi pelanggaran dari berbagai sudut pandang kamera sesuai keberadaan drone dengan mengobservasi secara langsung.
Prosedur Pengujian	<i>Website</i> alat ini akan dilakukan <i>Alpha</i> dan <i>Beta Testing</i> . Tim akan melakukan kedua metode tersebut demi mencari identifikasi kemungkinan <i>error</i> dan mencari kualitas dari <i>Website</i> tersebut. Selama menggunakan metode pengujian <i>Alpha</i> , hanya menguji fungsionalitas

dan kegunaannya, sedangkan selama metode pengujian *Beta* akan menguji kegunaan, fungsionalitas, keamanan, dan keandalan diuji [16].

2.3 Kesimpulan dan Ringkasan CD-2

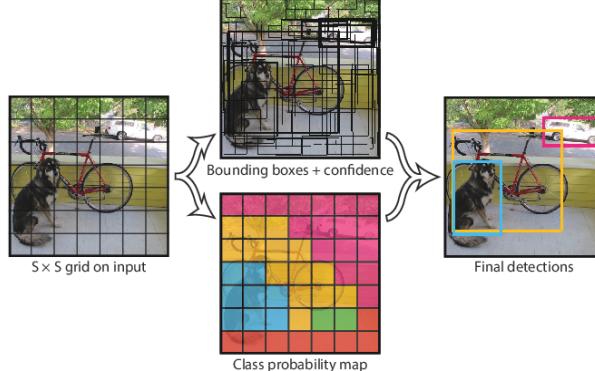
Berdasarkan poin-poin spesifikasi yang sudah dipaparkan di atas, dapat disimpulkan bahwa kami akan memilih setidaknya 4 dari 5 spesifikasi yang kami targetkan. Semua tergantung dengan bagaimana kami menguji sistem tersebut dan seberapa tinggi tingkat kesuksesan dari pengujian tersebut. Sistem juga akan kembali disesuaikan dengan keadaan yang akan terjadi kedepannya, karena jika ada terjadinya kendala maka kemungkinan akan ada beberapa spesifikasi yang akan dihilangkan, dengan begitu spesifikasi yang difokuskan akan bekerja secara maksimal seperti yang diinginkan.

BAB 3

DESAIN RANCANGAN SOLUSI

3.1 Konsep Sistem

3.1.1. You Only Look Once (YOLO)



Gambar 3.1 ilustrasi oleh Redmon et al

YOLO, yang merupakan singkatan dari *You Only Look Once*, adalah suatu metode pengenalan objek yang berbasis pada *Convolutional Neural Network* (CNN) bagaimana diilustrasikan oleh Redmon et al. (2016). YOLO memiliki keunikan dalam kesederhanaannya: seperti yang ditunjukkan pada Gambar . Dengan hanya satu jaringan konvolusional, YOLO mampu memprediksi sejumlah *bounding boxes* dan probabilitas kelas dari tiap kotak tersebut secara simultan. YOLO menggunakan gambar secara keseluruhan dalam proses pelatihannya dan secara langsung meningkatkan kinerja dalam mendekripsi objek. Dengan demikian, model terintegrasi ini menawarkan sejumlah kelebihan dibandingkan metode deteksi objek yang lebih tradisional. Keunggulan YOLO terletak pada kecepatannya. Versi dasar dari perangkat yang tinggi dapat menghasilkan 45 frame per detik tanpa memerlukan pengolahan batch pada perangkat keras spesifik seperti GPU Titan X. Sebuah versi yang lebih cepat bahkan bisa mencapai lebih dari 150 frame per detik. Ini memungkinkan kita untuk memproses *video streaming* secara *real-time* dengan latensi yang kurang dari 25 milidetik. Yang lebih penting, YOLO mampu mencapai presisi rata-rata yang lebih dari dua kali lipat dibandingkan dengan sistem real-time lainnya[17].

$$conf(class) = Pr(Class) \times IOU_{Pred}^{Truth}$$

Gambar 3.2 rumus confidence

Prinsip dasar YOLO adalah mempartisi gambar menjadi grid berdimensi $s \times s$ untuk proses deteksi objek. Di setiap grid, bounding box mengantisipasi keberadaan objek dan menghitung nilai *confidence*, yakni ukuran sejauh mana *bounding box* berisi objek yang sesuai dengan perencanaan dan proyeksi yang tepat. Nilai ini dapat diuraikan dengan persamaan dalam gambar. *Confidence score* ini sangat krusial dalam menjamin deteksi objek yang tepat. Lebih lanjut, pendekatan ini memfasilitasi algoritma untuk mempertimbangkan semua objek yang mungkin ada dalam gambar, alih-alih hanya berkonsentrasi pada area spesifik. Dengan metode ini, YOLO mampu mencapai deteksi objek secara *real-time* dengan tingkat presisi yang impresif. Metode ini juga

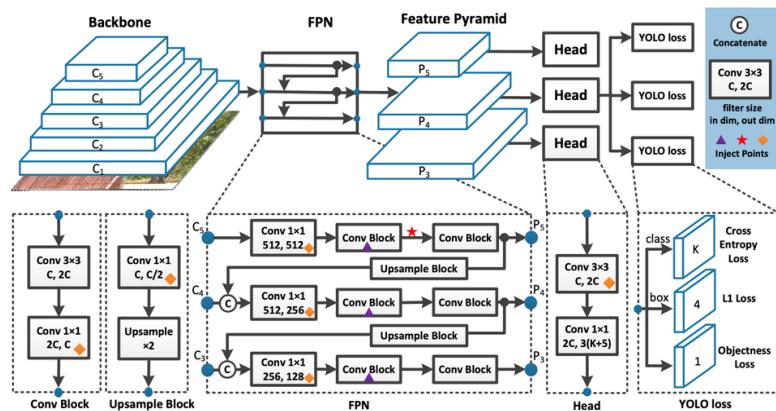
tanggap terhadap variasi dalam skala dan orientasi objek, yang menjadikannya pilihan yang efektif untuk beragam penerapan deteksi objek [18].

$$IOU_{Pred}^{Truth} = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Gambar 3.3 rumus *Intersection of Union*

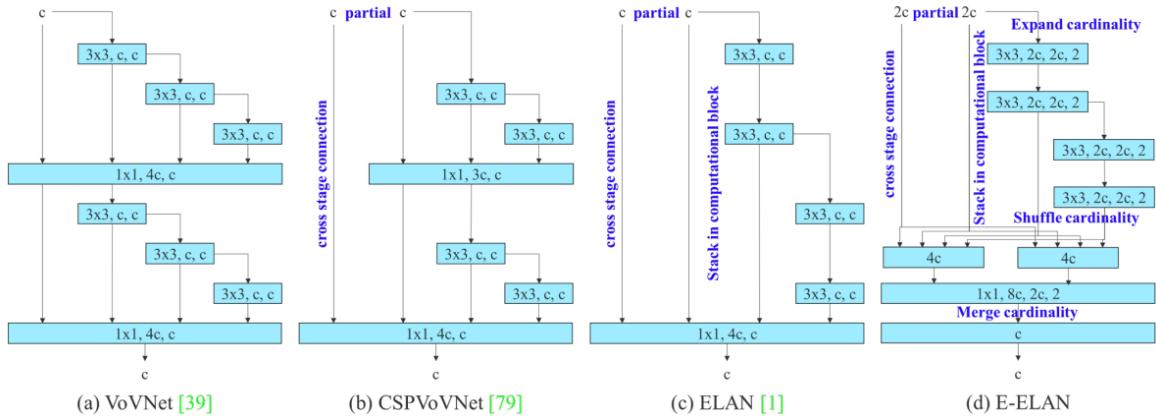
YOLO memperhitungkan nilai yang disebut *Intersection of Union* (IOU) untuk meningkatkan akurasi deteksi objek. IOU adalah rasio tumpang tindih antara kotak prediksi dan kotak ground truth. Semakin besar nilai IOU, semakin tinggi tingkat akurasi deteksi objek, seperti yang ditampilkan pada gambar persamaan .

Setiap *bounding box* dalam metode YOLO mengandung lima variabel: x, y, w, h, dan c. Variabel x dan y mencirikan koordinat pusat dari *bounding box* objek yang terdeteksi. Lebar (w) dan tinggi (h) dari *bounding box* digambarkan oleh variabel w dan h. Variabel terakhir, c, mewakili *confidence* dari *bounding box*, yang memberikan indikasi sejauh mana sistem yakin bahwa *bounding box* tersebut mengandung objek yang dicari. Jadi, konsep IOU dan struktur *bounding box* ini bekerja secara bersamaan untuk meningkatkan efektivitas dan keandalan deteksi objek oleh YOLO [18].



Gambar 3.4 arsitektur YOLOv7

Pada YOLOv7 menggunakan teknik *bag of freebies* yang dapat dilatih, teknik tersebut merupakan modul dan metode pengoptimalan yang bertujuan untuk meningkatkan akurasi deteksi objek, memperkuat biaya untuk pelatihan, sekaligus menjaga biaya interferensi. YOLOv7 juga menggunakan Penetapan Label Terpanas Kasar ke Halus baru alih-alih Penetapan Label Dinamis konvensional, yang meningkatkan pelatihan model dengan beberapa *layer output*. Efisiensi lapisan konvolusional jaringan YOLO sangat penting untuk kecepatan inferensi yang efisien. WongKinYiu memulai jalur efisiensi lapisan maksimal dengan *Cross Stage Partial Networks*. Dengan mengingat jumlah memori yang diperlukan untuk menyimpan lapisan dalam memori bersama dengan jarak yang dibutuhkan gradien untuk menyebar balik melalui lapisan. Semakin pendek gradiennya, semakin kuat jaringan mereka dapat belajar [19].



Gambar 3.5 arsitektur YOLOv7

Pada arsitektur YOLOv7 yang digunakan terbagi menjadi Backbone, Neck, dan Head. Pada Backbone digunakan model ELAN dan E-ELAN, yang digunakan sebagai jaringan saraf konvolusional yang mengumpulkan piksel gambar untuk membentuk fitur pada berbagai tingkat granularitas. *Backbone* biasanya dilatih sebelumnya pada dataset klasifikasi. Dengan menerapkan lebih banyak layer pada arsitektur dapat menghasilkan level tingkatan fitur yang berbeda dengan semantik yang lebih tinggi. Dan untuk *Neck* menggunakan model CSPSPP+(ELAN, E-ELAN)PAN yang digunakan sebagai lapisan ekstra yang digunakan untuk mengekstrak *feature maps* dari berbagai stages pada *backbone* sebelum dilanjutkan prosesnya ke *head*. Untuk *head* menggunakan model YOLOR yang berperan sebagai bagian dari jaringan yang bertanggung jawab membuat prediksi *bounding box* dan klasifikasi [20].

3.1.2. Parameter Kinerja YOLO

3.1.2.1. Precision

Fraksi deteksi positif benar di antara semua deteksi positif. Ini mengukur kualitas dari deteksi, yaitu, berapa banyak objek yang diprediksi yang benar-benar tepat [21]. *Precision* memiliki rumus seperti berikut:

$$Precision = \frac{TP}{TP+FP} \quad [21]$$

Penjelasan :

TP = *True Positive*

FP = *Fake Positive*

3.1.2.2. Recall

Fraksi deteksi positif benar di antara semua objek *ground truth*. Ini mengukur kelengkapan dari deteksi, yaitu, berapa banyak objek aktual yang terdeteksi [22]. *Recall* memiliki rumus seperti berikut:

$$Recall = \frac{TP}{TP+FN} \quad [22]$$

TP = *True Positive*

FN = *Fake Negative*

3.1.2.3. *F1-Score*

Rata-rata harmonik dari presisi dan *recall*. Ini memberikan keseimbangan antara presisi dan *recall*, dan merupakan metrik yang berguna ketika kelas tidak seimbang [23]. *F1-score* memiliki rumus seperti berikut:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad [23]$$

3.1.2.4. *Mean Average Precision*

Metrik yang digunakan untuk mengevaluasi model deteksi objek seperti YOLO. Ini adalah rata-rata dari AP (*Average Precision*) dari setiap kelas, di mana AP adalah area di bawah kurva presisi-recall. Ini mengukur kinerja keseluruhan dari model di semua kelas [24]. *mAP* memiliki rumus seperti berikut:

$$mAP = \sum_{i=1}^N \frac{AP_i}{N} \times 100\% \quad [24]$$

Penjelasan:

Σ (AP) adalah jumlah total dari *Average Precision* untuk semua kelas.

N adalah jumlah kelas.

3.1.3. Library Frontend dan Backend

A. Frontend

- React

React JS adalah *library* JavaScript yang dibuat oleh Facebook untuk membantu dalam pembuatan user interface pada aplikasi web dan mobile. Diluncurkan pada 2013, React JS adalah *open-source* dan tersedia secara gratis untuk semua pengguna. Salah satu fitur utama dari React JS adalah konsep berbasis komponen, di mana setiap komponen memiliki tampilan dan logika sendiri dan dapat digunakan ulang di berbagai tempat dalam aplikasi. Selain itu, React JS menggunakan Virtual DOM untuk melakukan update pada tampilan aplikasi secara efektif. Virtual DOM memperbolehkan React JS untuk memperbarui hanya bagian yang telah berubah dalam aplikasi, menghindari *refresh* halaman secara keseluruhan. Keuntungan lain dari React JS termasuk sifatnya yang deklaratif, efisiensi, fleksibilitas, kemudahan dalam pembelajaran, dan dukungan dari komunitas yang sangat aktif [25].

- Tailwind CSS

Tailwind CSS merupakan framework CSS yang memprioritaskan utility, memfasilitasi pembangunan desain antarmuka yang spesifik dengan kecepatan tinggi. Tidak seperti *framework* CSS lain seperti Bootstrap, Tailwind CSS tidak memberikan komponen yang sudah dirancang sebelumnya, melainkan menawarkan *class-class utility* pada level rendah yang memungkinkan pembuatan desain khusus secara lengkap, tanpa harus meninggalkan file HTML Anda. Tailwind menyertakan variasi responsif

pada setiap utilitasnya, yang memudahkan pembangunan interface yang responsif tanpa perlu menulis CSS khusus. Framework ini adalah ciptaan Adam Wathan, Jonathan Reinink, David Hemphill, dan Steve Schoger, dan pertama kali diluncurkan pada tahun 2017. Beberapa fitur utama Tailwind CSS antara lain konsep utility-first, interaktivitas, desain responsif, mode gelap, kemampuan penggunaan ulang, gaya khusus, serta fungsi dan direktif [26].

- React Router DOM

React Router DOM adalah *library* yang diperuntukkan bagi pengembangan aplikasi web menggunakan React. Fungsinya adalah untuk memfasilitasi pembuatan rute dan navigasi antar halaman dalam aplikasi web secara efisien dan cepat. React Router DOM menawarkan dua jenis router, yaitu HashRouter dan BrowserRouter, yang masing-masing memiliki kegunaan tergantung pada jenis situs web yang ingin dibuat. HashRouter biasanya digunakan dalam pembuatan situs web statis yang tidak memiliki server untuk merender data dinamis, sementara BrowserRouter digunakan dalam pembuatan situs web yang memiliki server untuk merender data dinamis [27].

B. Backend

- argparse

Argparse merupakan library Python yang berfungsi untuk mengolah argumen dalam command-line interface (CLI). Dengan library ini, kita bisa dengan mudah mendefinisikan argumen yang diperlukan oleh program dan Argparse akan mengelola argumen-argumen tersebut secara otomatis. Argparse menghadirkan berbagai fitur seperti penentuan argumen yang *mandatory*, argumen *optional*, tipe data argumen, serta nilai *default* argumen, di antara fitur lainnya [28].

- io

Library io dalam Python adalah library *built-in* yang dirancang untuk menangani operasi *input/output* (I/O) dalam Python. Ini mencakup berbagai kelas dan fungsi yang memungkinkan pengguna membaca dan menulis data dari berbagai sumber, termasuk file, string, dan aliran data. Beberapa kelas yang tersedia dalam *library* io antara lain BytesIO, StringIO, BufferedReader, dan BufferedWriter. BytesIO digunakan untuk membaca dan menulis data dalam format *byte*, sementara StringIO berfungsi untuk membaca dan menulis data sebagai string. Sementara itu, BufferedReader dan BufferedWriter digunakan untuk operasi baca/tulis pada *data buffer*, yang memungkinkan pengguna membaca dan menulis data dengan efisiensi tinggi. Dengan

demikian, *library* io pada Python sangat berperan penting dalam operasi I/O, seperti baca/tulis file dan baca/tulis data dari aliran data, dan lainnya [29].

- PIL

PIL (*Python Imaging Library*) merupakan sebuah *library* dalam Python yang dirancang untuk proses dan manipulasi gambar. PIL memberikan kemampuan bagi Python untuk berinteraksi dengan berbagai gambar. Namun, perkembangan PIL dihentikan pada tahun 2011 dan hanya kompatibel dengan Python 2. Sebagai alternatif, muncul library Pillow yang merupakan versi lebih baru dari PIL dan kompatibel dengan Python 3. Pillow menyuguhkan berbagai alat dan fungsi yang dapat digunakan untuk manipulasi gambar, seperti *resize*, memotong, memutar, merubah format, dan sebagainya. *Library* ini juga mendukung berbagai jenis format file gambar seperti BMP, PNG, JPEG, dan TIFF [30].

- *datetime*

Library *datetime* dalam Python merupakan library built-in yang dirancang untuk mengatur dan memanipulasi data tanggal dan waktu. Library ini mencakup berbagai kelas dan fungsi yang memungkinkan pembuatan, manipulasi, dan format dari tanggal dan waktu. Beberapa kelas yang tersedia dalam library *datetime* meliputi: *date*, *time*, *datetime*, dan *timedelta*. Kelas *date* digunakan untuk menyajikan data tanggal, kelas *time* digunakan untuk menyajikan data waktu, dan kelas *datetime* digunakan untuk menyajikan data tanggal dan waktu secara bersamaan. Sementara itu, kelas *timedelta* digunakan untuk mewakili perbedaan waktu antara dua objek *datetime* atau *time*. Oleh karena itu, *library datetime* dalam Python sangat bermanfaat untuk melakukan berbagai operasi yang terkait dengan tanggal dan waktu, seperti menghitung perbedaan waktu, memformat tanggal dan waktu, dan sebagainya [31].

- torch

Library torch dalam Python adalah library *open-source* yang digunakan dalam pengembangan dan pelatihan jaringan saraf (neural network) dalam ranah *machine learning*. PyTorch adalah library yang dioptimalkan berbasis Python dan Torch, dan secara luas digunakan dalam aplikasi yang membutuhkan penggunaan GPU dan CPU. PyTorch memanfaatkan grafik komputasi dinamis dan sepenuhnya berbasis Python, sehingga menjadi pilihan banyak praktisi dibandingkan dengan *framework deep learning* lain seperti TensorFlow dan Keras. Dengan API dan kinerja yang impresif, PyTorch sangat sesuai untuk digunakan oleh ilmuwan data dan pengembang perangkat

lunak. Beberapa fitur andalan PyTorch meliputi perhitungan tensor dengan akselerasi GPU, sistem diferensiasi otomatis berbasis *tape*, dan dukungan untuk *deep learning*. PyTorch telah digunakan dalam berbagai aplikasi dan proyek, seperti Tesla Autopilot, PyTorch Lightning, Hugging Face's Transformers, dan Catalyst [32].

- cv2

Library cv2 dalam Python adalah sebuah *library* yang diperuntukkan bagi pengolahan gambar dan video. *Library* ini adalah bagian dari OpenCV (*Open Source Computer Vision Library*), sebuah *library open-source* yang dirancang untuk pengolahan citra digital secara *real-time*. cv2 dalam Python menyediakan sejumlah fungsi dan algoritma untuk memproses gambar dan video, termasuk membaca dan menulis gambar, merubah ukuran gambar, mengubah warna gambar, mendeteksi wajah, dan sebagainya. cv2 juga mendukung berbagai format file gambar dan video seperti BMP, PNG, JPEG, dan AVI. Oleh karena itu, *library* cv2 dalam Python sangat bermanfaat untuk berbagai operasi yang terkait dengan pengolahan gambar dan video, seperti pengolahan gambar untuk keperluan *computer vision* dan *machine learning* [33].

- numpy

Library NumPy (*Numerical Python*) dalam Python adalah sebuah *library* yang digunakan untuk operasi array dan menawarkan fungsi yang beroperasi dalam ruang aljabar linier, transformasi Fourier, dan matriks. Dibuat pada tahun 2005 oleh Travis Oliphant, NumPy adalah proyek *open source* dan dapat digunakan secara gratis. NumPy menyimpan data dalam bentuk array, baik array 1-dimensi maupun multidimensi. Array 1-dimensi adalah kumpulan data dengan jenis variabel dan tipe data yang sama yang dapat diakses menggunakan satu indeks saja. Dengan fungsi aljabar linier, transformasi Fourier, dan komputasi matriks, NumPy menjadi alat penting dalam pengembangan aplikasi di mana kecepatan dan efisiensi sumber daya adalah prioritas utama. Tujuan dari NumPy adalah untuk menyediakan objek array yang 50 kali lebih cepat dibandingkan dengan list Python tradisional. *Library* NumPy dalam Python sangat penting untuk melakukan berbagai operasi yang berhubungan dengan komputasi numerik dan ilmiah, seperti pengolahan data, *machine learning*, dan *computer vision* [34].

- flask

Flask adalah sebuah *library* dan *framework* dalam Python yang dirancang untuk membangun aplikasi web. Sebagai salah satu *framework* web yang populer dan mudah

digunakan dalam Python, Flask memungkinkan pengembang untuk membangun aplikasi web secara cepat dan efisien. Dengan memanfaatkan Flask, pengembang dapat menciptakan aplikasi web yang terorganisir, mengendalikan perilaku aplikasi dengan mudah, dan memanfaatkan berbagai fitur yang disediakan oleh *framework* ini. Meski Flask sangat sesuai untuk proyek-proyek kecil dan sederhana, tetapi juga bisa digunakan dalam proyek yang lebih besar dan kompleks [35].

- os

Library os dalam Python adalah *library built-in* yang digunakan untuk menjalankan operasi terkait sistem operasi, seperti akses file, direktori, dan pengaturan sistem. *Library* ini menawarkan berbagai fungsi dan metode untuk mengakses dan memanipulasi sistem operasi, termasuk membaca dan menulis file, mengubah direktori kerja, dan mengatur variabel lingkungan, di antara operasi lainnya. Beberapa fungsi yang tersedia dalam *library* os meliputi os.getcwd() untuk mendapatkan direktori kerja saat ini, os.listdir() untuk mendapatkan daftar file dan direktori dalam direktori tertentu, os.mkdir() untuk membuat direktori baru, dan lainnya. Oleh karena itu, *library* os dalam Python sangat bermanfaat untuk melakukan berbagai operasi yang berhubungan dengan sistem operasi, seperti mengakses dan memanipulasi file dan direktori [36].

- re

Library re dalam Python adalah *library* yang dirancang untuk melakukan operasi ekspresi reguler atau regex. *Library* ini menawarkan berbagai fungsi dan metode yang memungkinkan pencocokan dan manipulasi string berdasarkan pola tertentu. Beberapa fungsi yang tersedia dalam *library* re meliputi re.search() untuk mencari pola spesifik dalam string, re.sub() untuk menggantikan pola tertentu dalam string, dan lainnya. Dengan demikian, *library* re dalam Python sangat berguna untuk operasi yang berkaitan dengan manipulasi string, seperti pencarian dan penggantian string berdasarkan pola tertentu [37].

- time

Library time dalam Python adalah *library built-in* yang dirancang untuk melaksanakan operasi yang berhubungan dengan waktu, seperti menghitung waktu eksekusi suatu program, menetapkan jeda waktu, dan merubah waktu ke berbagai format. *Library* ini menyuguhkan berbagai fungsi dan metode untuk mengakses dan memanipulasi waktu, termasuk time.time() untuk mendapatkan waktu saat ini dalam detik sejak epoch, dan time.sleep() untuk membuat program beristirahat sejenak, dan

lainnya. Library time juga dapat digunakan untuk mengkalkulasi durasi antara dua titik waktu, mengubah format waktu ke string, dan melaksanakan operasi yang berkaitan dengan tanggal dan waktu. Oleh karena itu, *library time* sangat berguna untuk operasi yang berkaitan dengan waktu dalam pengembangan aplikasi, penilaian performa, dan sinkronisasi waktu [38].

- *json*

Library json dalam Python adalah *library* yang digunakan untuk melakukan operasi yang berkaitan dengan JSON (*JavaScript Object Notation*) dalam Python. JSON adalah format data yang sering digunakan dalam pertukaran data antar aplikasi, khususnya aplikasi web. *Library json* di Python menawarkan berbagai fungsi dan metode untuk mengkonversi data Python ke format JSON, dan sebaliknya. Beberapa fungsi yang disediakan oleh *library json* meliputi `json.dumps()` untuk mengubah data Python ke format JSON, dan `json.loads()` untuk mengubah data JSON ke format Python, dan lainnya. Oleh karena itu, *library json* dalam Python sangat berguna untuk operasi yang berkaitan dengan pertukaran data antar aplikasi, terutama dalam konteks aplikasi web [39].

- *math*

Library math dalam Python adalah *library* built-in yang dirancang untuk melaksanakan operasi matematika di Python. *Library* ini menyuguhkan berbagai fungsi dan metode untuk melakukan operasi matematika, seperti menghitung nilai absolut, menghitung logaritma, menghitung akar kuadrat, dan lainnya. Beberapa fungsi yang disediakan oleh *library math* meliputi `math.sqrt()` untuk menghitung akar kuadrat, `math.log()` untuk menghitung logaritma, dan sebagainya. Oleh karena itu, *library math* dalam Python sangat bermanfaat untuk operasi matematika, seperti dalam pengolahan data, *machine learning*, dan *computer vision* [40].

- *pytesseract*

Library pytesseract dalam Python adalah *library* yang digunakan untuk melaksanakan operasi pengenalan karakter optik (OCR) di Python. *Library* ini memungkinkan pengguna untuk mengenali dan membaca teks yang ada dalam gambar. *Pytesseract* adalah sebuah *wrapper* dari *Tesseract* OCR, sebuah mesin OCR *open-source* yang dikembangkan oleh Google. *Library pytesseract* di Python menawarkan berbagai fungsi dan metode yang digunakan untuk mengenali dan membaca teks dalam gambar, seperti `pytesseract.image_to_string()` yang berfungsi

untuk mengubah gambar menjadi teks. Oleh karena itu, *library pytesseract* dalam Python sangat berguna untuk operasi yang berkaitan dengan OCR, seperti pengenalan teks dalam dokumen atau gambar [41].

- *watchdog*

Library watchdog dalam Python adalah *library* yang dirancang untuk memantau perubahan pada file dan direktori sistem. *Library* ini menawarkan berbagai fungsi dan metode untuk memantau perubahan pada file sistem dan direktori, seperti penambahan *observer* untuk melacak perubahan pada direktori tertentu, menambahkan *event handler* untuk menangani perubahan pada file sistem dan direktori, dan lainnya. Beberapa fungsi yang disediakan oleh *library watchdog* meliputi `watchdog.observers.Observer()` untuk membuat *observer*, dan `watchdog.events.FileSystemEventHandler()` untuk menangani *event* pada *file* sistem dan direktori, dan lainnya. Oleh karena itu, *library watchdog* dalam Python sangat berguna untuk operasi yang berkaitan dengan pemantauan *file* sistem dan direktori, seperti melacak perubahan pada file dan direktori untuk tujuan seperti backup dan sinkronisasi, di antara lainnya [42].

- *threading*

Library threading dalam Python adalah *library* yang digunakan untuk melakukan operasi *multithreading*, yang memungkinkan beberapa thread berjalan secara simultan dalam satu program. Sebuah *thread* adalah entitas eksekusi yang dapat beroperasi secara independen dalam sebuah program. *Library threading* di Python menyediakan fungsi dan metode yang memungkinkan pengguna untuk menciptakan, mengendalikan, dan mengkoordinasikan thread dalam program mereka. Beberapa fungsi yang disediakan oleh *library threading* meliputi `threading.Thread()` untuk membuat thread baru, `threading.Lock()` untuk mengatur akses ke sumber daya bersama, dan sebagainya. Penggunaan *library threading* dapat meningkatkan efisiensi dan performa program dengan menjalankan tugas-tugas independen secara simultan. Oleh karena itu, *library threading* dalam Python sangat berguna untuk operasi yang berkaitan dengan konkurensi dan paralelisme, seperti optimalisasi penggunaan CPU, penanganan tugas yang memerlukan waktu lama, dan lainnya [43].

- *queue*

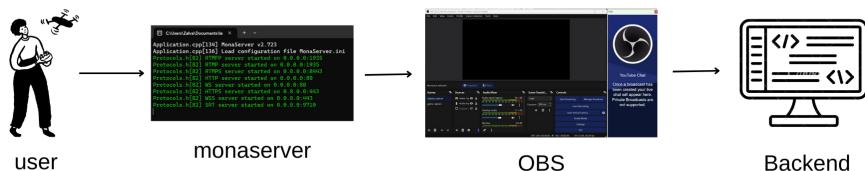
Library queue dalam Python adalah sebuah *library* yang digunakan untuk menerapkan struktur data antrian dalam sebuah program. Antrian adalah struktur data yang mengikuti prinsip FIFO (*First-In, First-Out*), yang berarti elemen yang pertama

kali masuk akan menjadi yang pertama keluar. *Library queue* di Python menyediakan kelas *Queue* yang memungkinkan pengguna untuk membuat antrian baru, menambahkan elemen ke dalam antrian, menghapus elemen dari antrian, dan menjalankan operasi lain yang berkaitan dengan antrian. Beberapa metode yang disediakan oleh library queue meliputi *Queue()* untuk membuat antrian baru, *put(item)* untuk menambahkan elemen ke dalam antrian, dan *get()* untuk menghapus dan mengembalikan elemen dari antrian, di antara lainnya. Oleh karena itu, *library queue* dalam Python sangat berguna untuk operasi yang berkaitan dengan manajemen antrian, seperti pemrosesan tugas berdasarkan urutan datangnya atau penjadwalan [44].

- flask_cors

Library Flask-Cors dalam Python adalah sebuah *library* yang dirancang untuk mengelola *Cross-Origin Resource Sharing* (CORS) pada aplikasi web yang dibuat dengan Flask. CORS adalah suatu mekanisme keamanan web yang membatasi akses ke sumber daya dari domain yang berbeda. *Library Flask-Cors* di Python menyediakan fungsi dan metode yang memungkinkan pengguna untuk mengelola CORS dalam aplikasi web Flask, seperti menambahkan *decorator* untuk mengatur CORS pada rute tertentu, dan mengatur *header* CORS, dan lainnya. Beberapa fungsi yang disediakan oleh *library Flask-Cors* meliputi *flask_cors.CORS()* untuk mengatur CORS dalam aplikasi Flask, dan *flask_cors.cross_origin()* untuk menambahkan *decorator* pada rute tertentu, dan lainnya. Oleh karena itu, *library Flask-Cors* dalam Python sangat berguna untuk operasi yang berkaitan dengan pengaturan CORS dalam aplikasi web Flask, seperti membolehkan akses dari domain yang berbeda atau mengatur *header* CORS [45].

3.1.4. Koneksi Drone dengan Perangkat keras



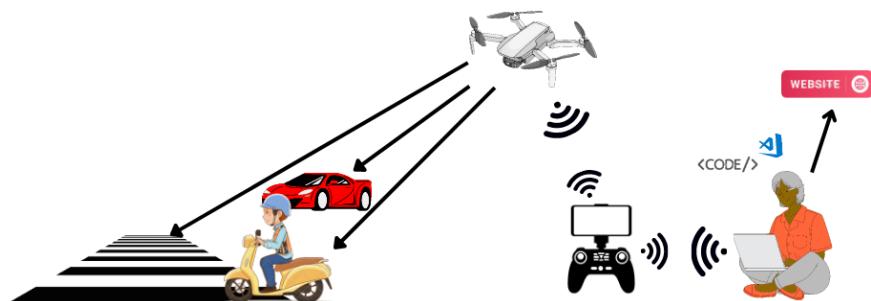
Gambar 3.6 ilustrasi monaserver

MonaServer adalah server multimedia yang dirancang untuk memberikan solusi real-time dengan mudah dan fleksibel. Ia mendukung berbagai protokol seperti

RTMFP, RTMP, RTMPE, WebSocket, dan HTTP. OBS (Open Broadcaster Software) adalah perangkat lunak sumber terbuka dan gratis untuk merekam video dan streaming langsung. OBS memiliki dukungan yang luas untuk berbagai plugin dan skrip, dan dapat digunakan untuk siaran langsung, rekaman layar, dan lainnya. OBS Virtual Camera adalah salah satu plugin tersebut, yang memungkinkan output OBS dapat digunakan sebagai webcam virtual dalam aplikasi lain.

Rancangan penggunaan MonaServer dan OBS dimulai dengan mengatur MonaServer pada **Gambar 3.6** sebagai pusat koneksi RTMP. MonaServer akan berfungsi sebagai server RTMP yang akan menerima koneksi drone ke stream dari OBS dan menyiarannya ke aplikasi web. Anda akan perlu mengonfigurasi pengaturan MonaServer agar cocok dengan kebutuhan, seperti penyetelan alamat IP dan port. OBS akan digunakan untuk menangkap video dari tayangan RTMP drone, mengolahnya, dan mengirimkannya ke aplikasi web. Ini berarti bahwa apa pun yang ditampilkan di OBS, termasuk overlay, grafis, dan lainnya, akan ditampilkan dalam aplikasi web.

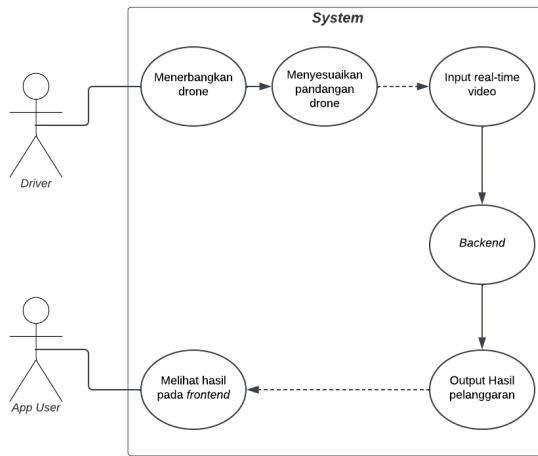
3.1.5. Arsitektur utama sistem



Gambar 3.7 ilustrasi Fungsi sistem

Sistem akan menggunakan input untuk melakukan pengambilan data berupa live video yang akan dikirim oleh perangkat komunikasi lalu akan diproses. Proses akan dilakukan oleh program *Deep Learning* untuk mengklasifikasikan jenis kendaraan dan mengkategorikannya, apakah kendaraan yang digunakan berupa mobil atau motor. Selanjutnya akan dilakukan klasifikasi jenis pelanggarannya, dari data tersebut akan dikelompokkan kembali dan akan dibuat transkrip laporan sebagai keluaran dari proses / *output*. Hasil keluaran / *output* yang akan langsung dikirim kembali ke aplikasi yang digunakan oleh pengguna / user.

- Interaksi Pengguna



Gambar 3.8 Use Case Diagram

Pada *Use Case Diagram* menggambarkan bagaimana user melakukan beberapa interaksi terhadap sistem, seperti menerbangkan drone, menginisiasi sistem, memilih target yang akan diidentifikasi, dan dapat melihat hasilnya yang sudah sesuai dengan target yang dipilih.

3.1.6. Analisis

3.1.4.1 Kriteria

Tabel 3.1 Tabel Kriteria

Kriteria aksesibilitas	Seberapa mudah sistem yang akan dipilih untuk diakses oleh pengguna dan pengguna yang menerbangkan drone harus memiliki sertifikasi penerbangan drone.
Kriteria pengoperasian sistem	Sistem yang akan dioperasikan oleh pengguna mengetahui seberapa mudah pengoperasian sebuah sistem dalam melakukan sebuah solusi dari suatu permasalahan.
Kriteria estimasi waktu	Suatu proses pengambilan data dengan waktu real time sehingga proses informasi yang terkait dengan jenis pelanggaran kendaraan tersampaikan dengan cepat ke data penyimpanan.
Kriteria interaksi pengguna	Seberapa baik pengguna dalam menggunakan sistem serta keahlian dalam menggunakan drone sehingga pengguna dapat memberikan solusi terbaik dalam permasalahan.

3.1.4.2 Analisis konsep

Tabel 3.2 Analisis Konsep

Kriteria	Kelebihan	Kekurangan
Kriteria aksesibilitas	pengguna akan mendapatkan jaminan dengan sistem yang dapat diandalkan dikarenakan dengan adanya teknologi YOLO pada sistem serta drone yang dapat menggunakan akses secara individu.	Tidak semua pengguna memiliki pemahaman terhadap sistem yang disediakan dan penggunaan drone yang tidak memiliki sertifikat drone serta pengambilan yang tidak efektif dalam pengambilan data.
Kriteria pengoperasian sistem	pengguna dapat mengoperasikan sistem yang sudah disediakan beserta fitur yang dapat digunakan untuk mengambil data yang akan disimpan kedalam aplikasi.	pengguna harus bisa mengoperasikan drone yang nantinya akan dipakai untuk pengambilan gambar dari ketinggian yang berbeda - beda.
Kriteria estimasi waktu	pengguna dapat meminimalisir estimasi waktu dalam pengukuran sedang diobservasi, sehingga memiliki efektivitas waktu yang cukup signifikan	Pengguna harus memperhatikan situasi yang sedang terjadi dilapangan baik itu cuaca, arah angin, maupun kekuatan hembusan angin, yang akan sangat berpengaruh terhadap efektivitas waktu.
Kriteria interaksi pengguna	pengguna dilatih untuk terbiasa dengan sistem dan cara menggunakan drone untuk mengambil data pelanggaran.	sistem yang digunakan untuk pengguna butuh beberapa waktu untuk terbiasa menggunakan sistem dan menggunakan drone dengan ketinggian dalam pengambilan data

3.1.7. Sistem yang akan Dikembangkan

Sistem ini memiliki kebutuhan utama pada bagian program *deep learning*, terutama mengenai akurasi dalam proses pemindaian kendaraan dan jenis pelanggaran yang dilakukan. Memberikan gambaran informasi secara *realtime* juga menjadi salah satu kebutuhan utama dari sistem ini. Kebutuhan ini juga perlu mendapat dukungan dari kriteria aksesibilitas sehingga sistem dapat mengolah data dengan akurat.

pada penelitian ini kami akan mendeteksi beberapa jenis pelanggaran diantaranya Penumpang kendaraan sepeda motor tidak menggunakan helm, kendaraan bermotor menerobos lampu merah, dan kendaraan bermotor berhenti di area *zebracross*. Mengacu pada ketiga jenis pelanggaran tersebut kami telah melakukan *research* seputar undang-undang yang berkaitan dengan jenis pelanggaran tersebut, berikut rincian dari peraturan perundang-undangan yang berkaitan:

Tabel 3.3 UU yang berkaitan dengan pengembangan sistem

Pelanggaran	UU
(1) Orang yang menggunakan kendaraan sepeda motor saat berkendara tidak menggunakan helm	<p>Pasal 106 ayat (8) UU No. 22/2009 “Setiap orang yang mengemudikan Sepeda Motor dan Penumpang Sepeda Motor wajib mengenakan helm yang memenuhi standar nasional Indonesia.”</p>
	<p>Pasal 291 ayat (1) UU No. 22/2009 “Setiap orang yang mengemudikan Sepeda Motor tidak mengenakan helm standar nasional Indonesia sebagaimana dimaksud dalam Pasal 106 ayat (8) dipidana dengan pidana kurungan paling lama 1 (satu) bulan atau denda paling banyak Rp250.000,00 (dua ratus lima puluh ribu rupiah).”</p>
	<p>Pasal 291 ayat (2) UU No. 22/2009 “Setiap orang yang mengemudikan Sepeda Motor yang membiarkan penumpangnya tidak mengenakan helm sebagaimana dimaksud dalam Pasal 106 ayat (8) dipidana dengan pidana kurungan paling lama 1 (satu) bulan atau denda paling banyak Rp250.000,00 (dua ratus lima puluh ribu rupiah).”</p>
(2) Orang yang menggunakan kendaraan bermotor dan berhenti diatas garis <i>zebracross</i>	<p>Pasal 106 ayat (3) UU No. 22/2009 “Setiap orang yang mengemudikan Kendaraan Bermotor di Jalan wajib mematuhi ketentuan tentang persyaratan teknis dan laik jalan.”</p>
	<p>Pasal 106 ayat (4) UU No. 22/2009 “Setiap orang yang mengemudikan Kendaraan Bermotor di Jalan wajib mematuhi ketentuan:</p> <ul style="list-style-type: none"> a. rambu perintah atau rambu larangan; b. Marka Jalan;

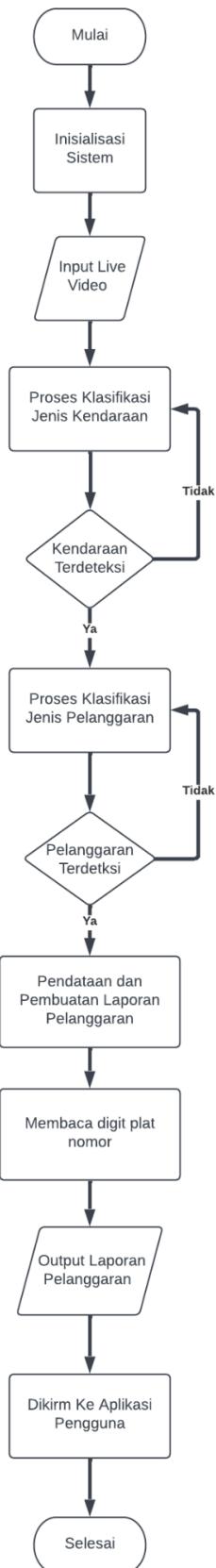
Pelanggaran	UU
	<p>c. Alat Pemberi Isyarat Lalu Lintas;</p> <p>d. gerakan Lalu Lintas;</p> <p>e. berhenti dan Parkir;</p> <p>f. peringatan dengan bunyi dan sinar;</p> <p>g. kecepatan maksimal atau minimal; dan/atau</p> <p>h. tata cara penggandengan dan penempelan dengan kendaraan lain.</p> <p>Pasal 287 ayat (1) UU No. 22/2009 “Setiap orang yang mengemudikan Kendaraan Bermotor di Jalan yang melanggar aturan perintah atau larangan yang dinyatakan dengan Rambu Lalu Lintas sebagaimana dimaksud dalam Pasal 106 ayat (4) huruf a atau Marka Jalan sebagaimana dimaksud dalam Pasal 106 ayat (4) huruf b dipidana dengan pidana kurungan paling lama 2 (dua) bulan atau denda paling banyak Rp500.000,00 (lima ratus ribu rupiah).”</p>

3.2 Desain Sistem



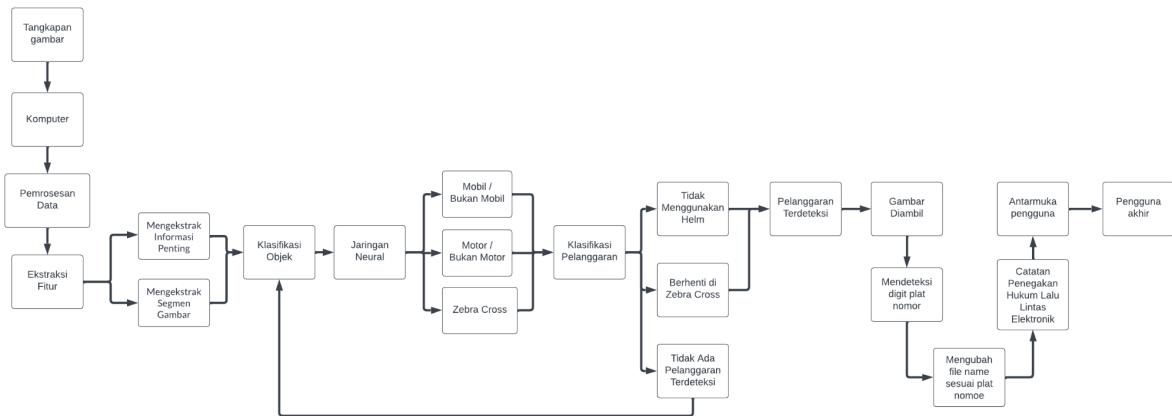
Gambar 3.9 Diagram blok sistem

Sistem terbagi menjadi beberapa sub sistem. Masing - masing dari sub sistem akan saling terhubung sehingga mempengaruhi kerja dari sub sistem yang lainnya. Input digunakan untuk melakukan pengambilan data berupa *live video* yang akan dikirim oleh perangkat komunikasi lalu akan diproses. Proses akan dilakukan oleh program *Deep Learning* untuk mengklasifikasikan jenis kendaraan dan mengkategorikannya, apakah kendaraan yang digunakan berupa mobil atau motor. Selanjutnya akan dilakukan klasifikasi jenis pelanggarannya, dari data tersebut akan dikelompokan kembali dan akan dibuat transkrip laporan sebagai keluaran dari proses / *output*. Hasil keluaran / *output* yang akan langsung dikirim kembali ke aplikasi yang digunakan oleh pengguna / *user*.



Gambar 3.10 Flowchart solusi sistem

Sistem akan bekerja sesuai dengan alur pada diagram diatas. Ketika drone mengirimkan data berupa *live record*, maka program *deep learning* akan melakukan proses terhadap tangkapan gambar tersebut. Dari hasil proses tersebut program akan mengeluarkan *output* berupa bukti *screenshot* dan surat laporan pelanggaran yang telah dilakukan. Hasil keluaran akan dikirimkan langsung ke aplikasi dari pengguna.

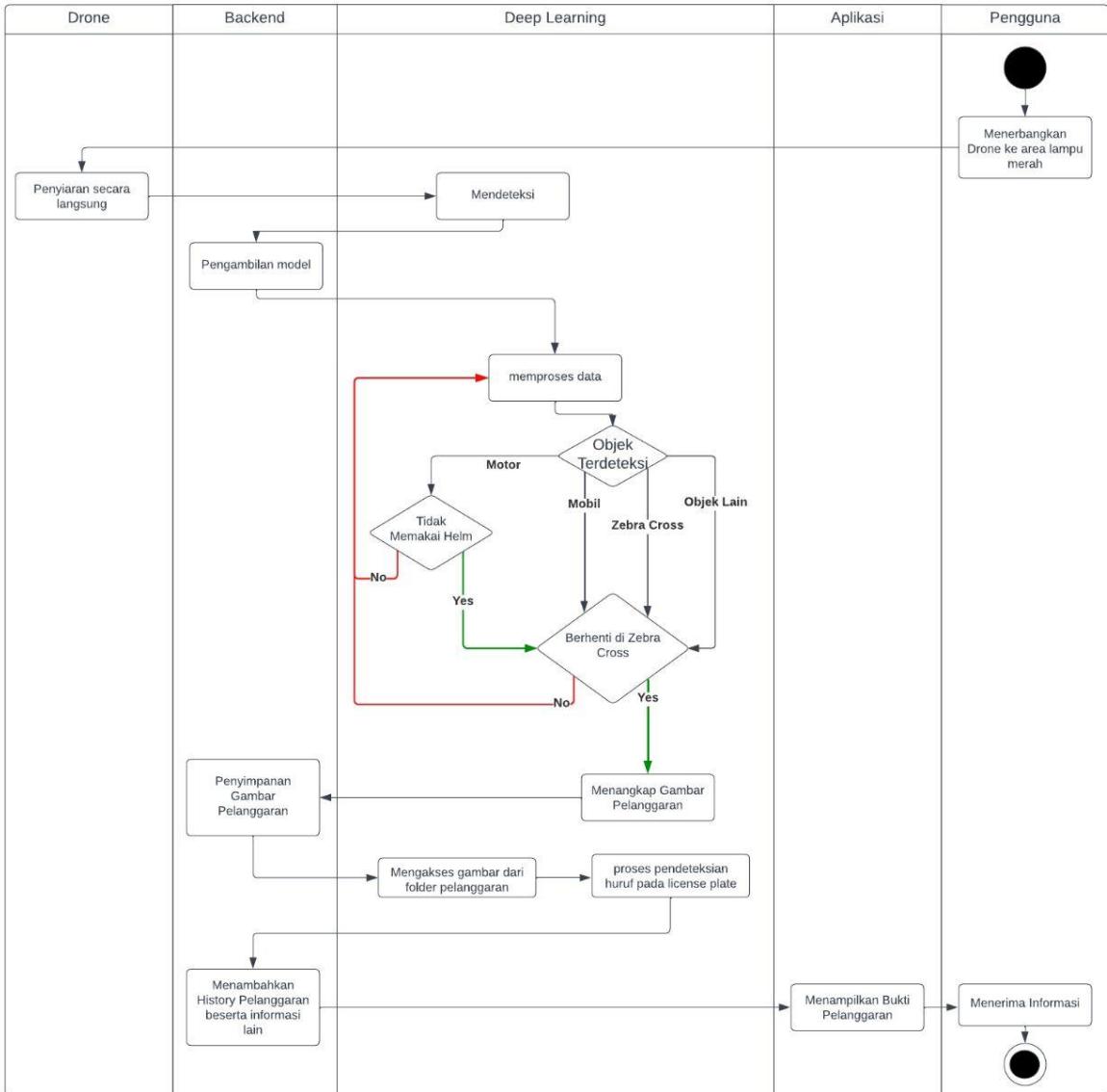


Gambar 3.11 Diagram blok keseluruhan sistem

Kerja sistem secara keseluruhan terdapat pada diagram diatas. Sistem dimulai ketika drone mengambil dan mengirimkan *live video* ke dalam komputer. Komputer akan secara langsung memasukan hasil tangkapan ke dalam program *deep learning*. Dataset yang sudah ada akan dibuat sebagai gambaran dasar dari proses identifikasi kendaraan dan pelanggaran lalu lintas yang dilakukan. Setelah program mulai memahami dan menunjukkan hasil yang akurat terhadap deteksi yang dilakukan, maka program akan melakukan serangkaian proses ekstraksi. Dalam proses tersebut program akan melakukan *Feature Extraction*, yang dimana secara hampir bersamaan program akan melakukan *Important Information*, *Important colors* dan *Image Segment Extraction*. *Feature Extraction* merupakan komponen inti dari saluran visi komputer, fitur ini berguna untuk membuat program dapat dengan jelas mendefinisikan objek dalam gambar[1].

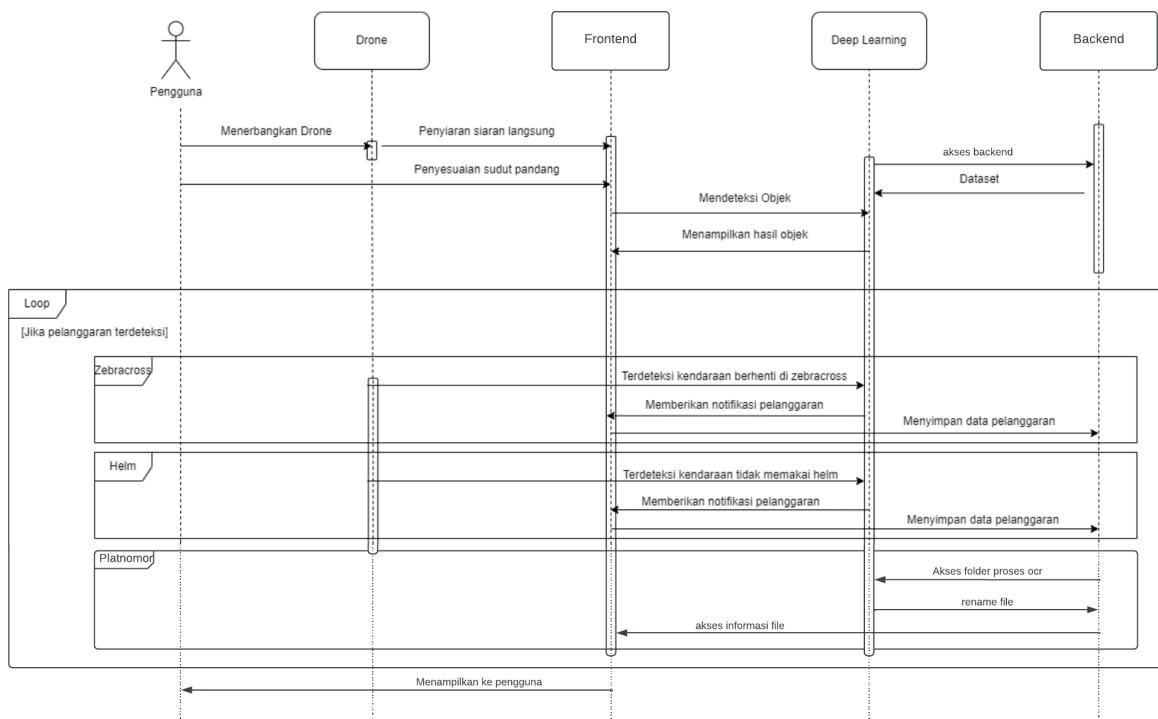
Setelah program melakukan proses ekstraksi, program akan langsung mengklasifikasi objek menggunakan *Neural Networks* untuk mendeteksi kendaraan apa yang digunakan oleh pengendara. Setelah mengetahui kendaraan apa yang digunakan, program akan mengklasifikasi pelanggaran yang dilakukan oleh pengendara sesuai dengan kendaraan yang digunakan apabila tidak terbukti adanya pelanggaran maka program akan lanjut melakukan scan ke kendaraan yang lain dengan kembali ke proses klasifikasi objek. Namun, apabila program mendeteksi adanya pelanggaran yang dilakukan, maka program akan secara langsung mengambil *screenshot* dan mengirimkannya ke dalam database, yang nantinya akan langsung dibuatkan surat laporan pelanggaran di *Electronic Traffic Law Enforcement Record* (ETLE).

Data output yang dikeluarkan dari program tersebut berupa surat laporan pelanggaran lalu lintas yang didalamnya sudah dijelaskan terkait pasal yang dilanggar dan hasil *screenshot* yang sudah diambil yang akan langsung dikirimkan ke aplikasi pengguna dan akan memunculkan notif pada aplikasi tersebut.



Gambar 3.12 Activity Diagram

diagram ini menunjukkan aktivitas dari masing masing fungsi setiap komponen, yang menggambarkan cara kerja (*Workflow*) dari suatu sistem. proses mendeteksi pelanggaran lalu lintas dimulai dari pengguna melakukan penerbangan secara manual, drone yang diterbangkan sudah dilakukan konfigurasi penyiaran langsung atau *live broadcast* ke komputer yang kemudian komputer akan melakukan pendekstian menggunakan *deep learning* YOLO(*You Only Look Once*). Isi dari *deep learning* tersebut terdapat algoritma pendekstian sederhana dalam bentuk jenis kendaraan dan marka jalan, setelah itu dilanjutkan dengan pendekstian pelanggaran dari masing masing kendaraan. Hasil pendekstian akan berbentuk tangkapan foto pelanggaran yang dilakukan pengendara, foto tersebut akan dikirim ke database agar aplikasi web dapat menampilkan hasil pelanggaran. pada intinya pengguna hanya melakukan penerbangan drone dan menerima informasi yang diberikan oleh aplikasi web dikarenakan semua pekerjaan pendekstian sudah bekerja dengan sendirinya .



Gambar 3.13 Sequence Diagram

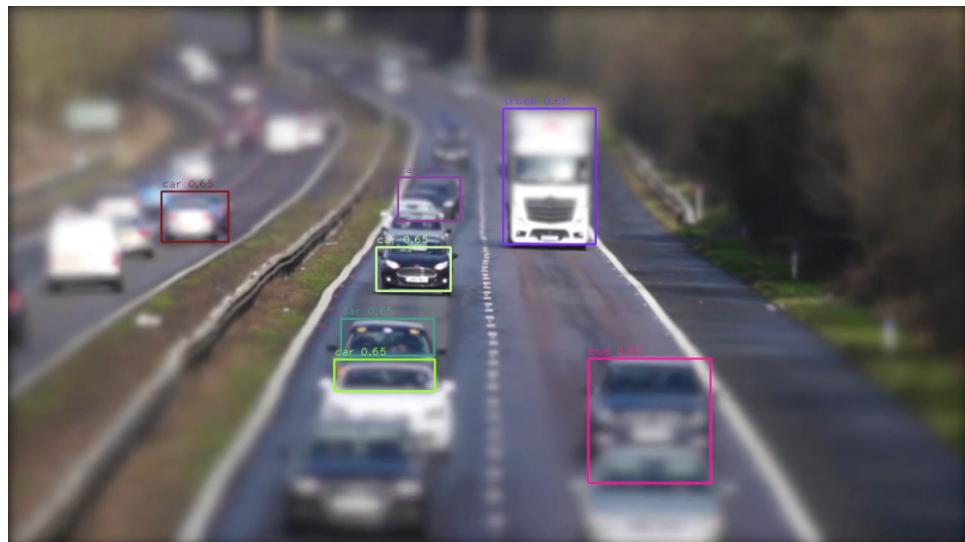
Sequence Diagram yang menggambarkan urutan pada sistem yang diawali dengan user untuk menerbangkan drone yang bertujuan mendeteksi pelanggaran serta menangkap hasil pelanggaran yang berupa pelanggaran berhenti di zebra cross dan tidak memakai helm, hasil data pelanggaran akan ditampilkan pada *website*.

3.3 Pengujian Komponen (Kalibrasi)

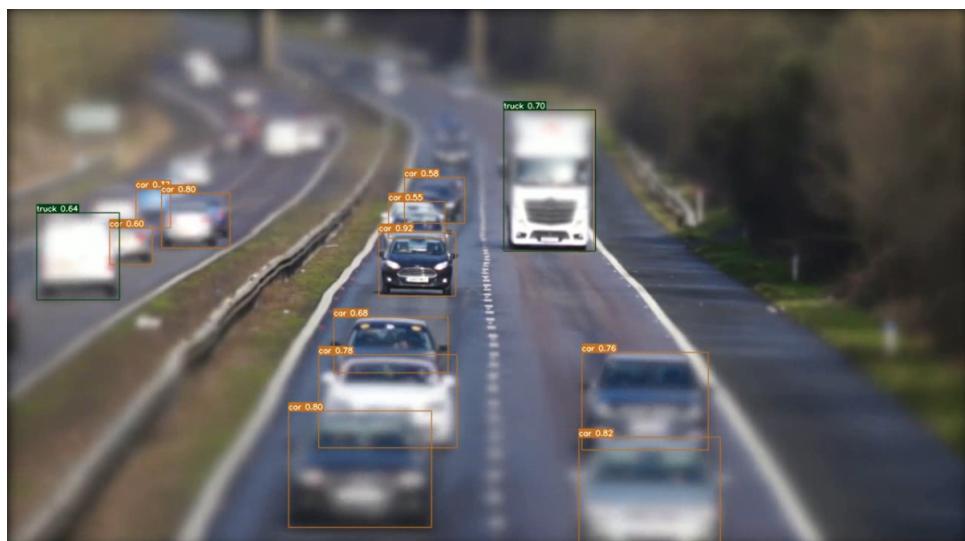
3.3.1. Algoritma

Tabel 3.4 Pengujian Algoritma

Komponen	YOLOv7 dan YOLOv3
Rincian	Mengukur tingkat <i>confidence score</i> dan keagresifan yang timbul dalam mendeteksi objek kendaraan.
Metode Pengukuran	Pengujian kedua model ini akan dicoba menggunakan video yang sama untuk mengukur tingkat <i>confidence score</i> dan keagresifan deteksi objek kendaraan.
Prosedur Pengujian	Menggunakan video yang sama dan model sesuai dengan yang kami gunakan untuk pendekripsi. Video akan diproses dan menghasilkan video yang baru.



Gambar 3.14 Car Detection Test



Gambar 3.15 Car Identification

Penggunaan YOLO dari versi yang berbeda sebagai pendekripsi objek diuji dengan penggunaan video yang sama menghasilkan *export file* dengan FPS yang berbeda. hal ini terjadi karena YOLOv3 *inference time* berbeda dengan YOLOv7. *inference time* disimpulkan sebagai jumlah waktu yang dibutuhkan model *machine learning* untuk memproses data baru dan membuat prediksi. selain itu kedua model ini juga menguji sampel lain yaitu, *mAP* atau *mean Average Precision* adalah rata rata Presisi mengukur seberapa akurat prediksi dalam model tersebut. yaitu persentase prediksi model yang dihasilkan benar [2].

Tabel 3.5 Perbandingan Pengujian Algoritma

YOLOv7		YOLOv3	
mAP (%)	inference time (ms)	mAP (%)	inference time (ms)

YOLOv7		YOLOv3	
51	6	40	22
53	9	43	29
55	12	45	51

Jelas dari angka tersebut bahwa dari YOLOv7, tidak ada persaingan dalam hal kecepatan dan akurasi. Meskipun ini bukan lagi model pendekripsi objek yang paling akurat, YOLOv3 masih merupakan pilihan yang baik saat Anda membutuhkan pendekripsi real-time dengan tetap mempertahankan akurasi yang baik.

3.3.2. Perangkat Drone

Tabel 3.6 Pilihan Drone

Pilihan Drone		
Tipe	DJI Mavic Pro Platinum – Longer And Quieter Flights	Dji Mavic Mini 1
Gambar		
Harga	Rp.17.000.000	Rp. 8.000.000

3.3.3. Perangkat Komputer

Tabel 3.7 Pilihan perangkat komputer

Spesifikasi Komputer		
Tipe	Personal Computer	Laptop
Gambar		
RAM	Up to 32gb	Up to 32gb
DDR	Up to DDR5	Up to DDR5
VGA	VRAM Up to 16gb	VRAM Up to 12gb
Processor	Minimal 3.50Ghz	Minimal 3.50Ghz
Battery	On Power	Minimal bertahan 8 jam

Spesifikasi Komputer

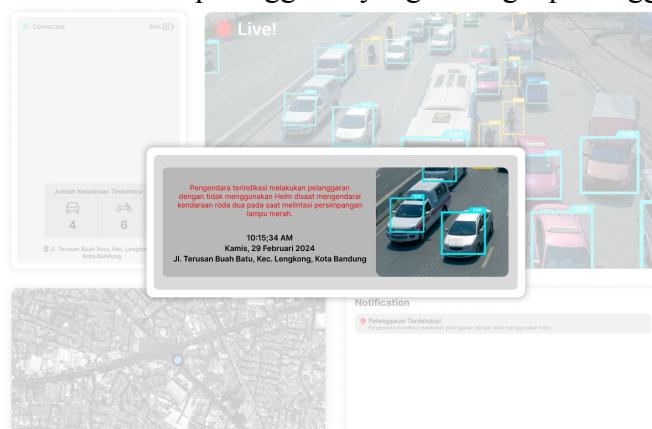
Harga	Rp.50.000.000	Rp.36.000.000
-------	---------------	---------------

3.3.4. Desain Aplikasi Web



Gambar 3.16 Tampilan UI sederhana pada website

Produk ini menerapkan desain sederhana agar pengguna mudah dalam menggunakan aplikasi deteksi pelanggaran ini. aplikasi web dirancang menggunakan Figma sebagai *Prototype*. *user flow* atau yang bisa diartikan urutan pengguna saat menggunakan produk ini tidak rumit, pengguna hanya menggunakan 1 fitur yang dapat ditekan untuk melihat hasil dari pelanggaran yang tertangkap menggunakan drone.



Gambar 3.17 Tampilan UI indikasi pelanggaran

Selain itu, aplikasi ini juga memberikan bermacam informasi seperti jumlah kendaraan yang terdeteksi, lokasi penerbangan drone, sistem drone, dan juga tampilan drone bersamaan dengan bounding box.

3.4 Jadwal Pengerjaan

Tabel 3.6 Gantt chart

Aktivitas	Progres	2022				2023					
		Sep	Okt	Nov	Des	Jan	Feb	Mar	Apr	Mei	Jun
Capstone Design CD-1											
Penentuan topik CD-1	100%										
Penyusunan CD-1	100%										
Pengumpulan literatur dan referensi	100%										
Rumusan masalah	100%										
Penentuan solusi	100%										
Capstone Design CD-2											
Penyusunan CD-2	100%										
Perumusan spesifikasi	100%										
Penentuan verifikasi	100%										
Penentuan konsep sistem	100%										
Capstone Design CD-3											
Pengujian YoloV7	100%										
Perancangan kebutuhan sistem	100%										
Perumusan karakteristik sistem	100%										
Pengumpulan dataset	100%										
Training dataset	100%										
Pembuatan algoritma	100%										
Penulisan CD-3	100%										
Capstone Design CD-4 dan CD-5											
Pembuatan design website	100%										
Penyelesaian training dataset	100%										
Test terbang drone	100%										
Percobaan keseluruhan sistem (Hardware, Algoritma, Software)	100%										
Penulisan CD-4	100%										
Penulisan CD-5	100%										
Pendaftaran sidang TA											
Penyusunan dokumen TA	100%										
Revisi draft dokumen TA	100%										
Pendaftaran sidang TA	100%										
Latihan pra sidang TA	100%										

3.5 Kesimpulan dan Ringkasan CD-3

Setelah berbagai pembahasan di atas, kami menyimpulkan bahwa sistem yang kami pilih akan berdasarkan data yang sudah tercantum pada pembahasan diatas. Diawali dengan pemilihan sistem YOLOv7 dibandingkan dengan YOLOv3, dikarenakan nilai yang dihasilkan dari YOLOv7 jauh lebih baik. Berikut juga dijelaskan apa saja faktor yang digunakan untuk mengukur manakah yang terbaik, seperti *Precision*, *Recall*, *F1-Score*, dan *map* (*mean Average Precision*). Dari pembahasan di atas dijelaskan pula mengenai diagram blok dari keseluruhan sistem sebagai gambaran akan seperti apa sistem akan bekerja, bersamaan dengan *flowchart*, *activity*, *sequence* dan *use case diagram*.

Pada pembahasan di atas kami juga memberikan keterangan mengenai komponen komponen yang akan digunakan selama pembuatan dan penggunaan sistem, seperti jenis drone, algoritma dan jenis *hardware* yang digunakan.

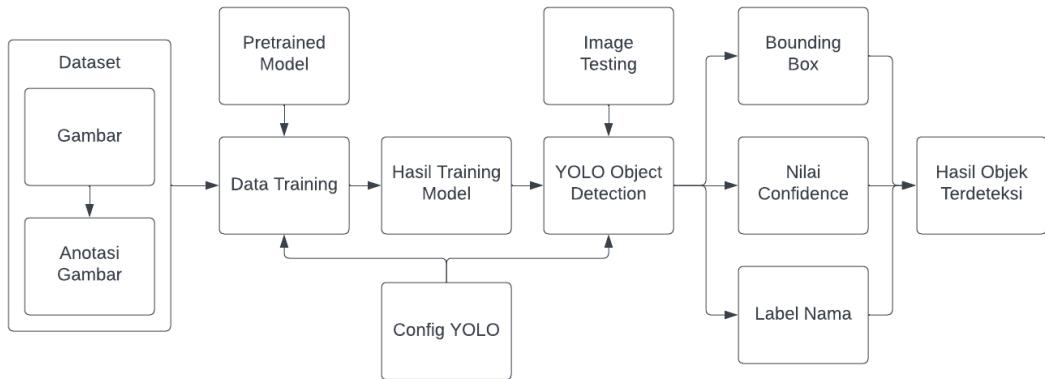
BAB 4

IMPLEMENTASI

4.1 Implementasi Sistem

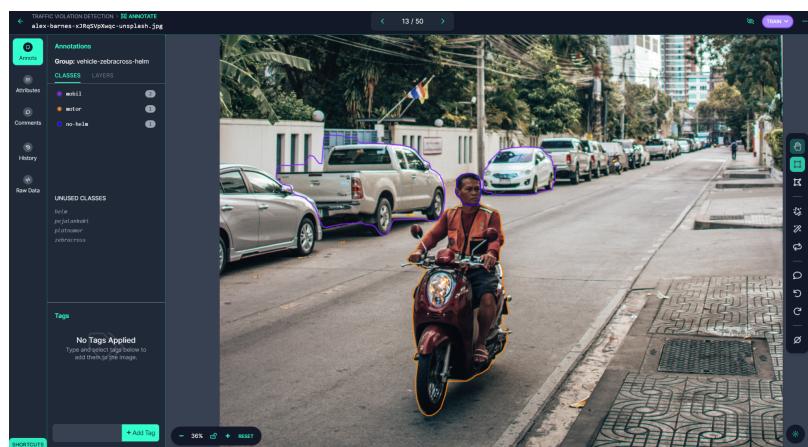
4.1.1 Sub-sistem 1 (Dataset)

4.1.1.1. Cara Kerja Sub-Sistem



Gambar 4.1 Diagram block

Roboflow adalah aplikasi web yang digunakan dalam bagian dataset Gambar 4.1, berfungsi untuk mengolah dan mengatur dataset gambar. Aplikasi ini digunakan oleh berbagai praktisi machine learning untuk mengotomatiskan proses anotasi, *preprocessing*, dan augmentasi data. Dalam konteks deteksi objek menggunakan YOLO (*You Only Look Once*), dataset gambar yang sudah dikumpulkan perlu dianotasi untuk menandai di mana objek tertentu berada di dalam gambar. Hal ini sangat penting karena akan membantu model belajar mengenali objek yang diinginkan.



Gambar 4.2 proses anotasi gambar yang akan di *training*

Dalam aplikasi Roboflow, anotasi dilakukan dengan cara menandai objek pada gambar dengan *bounding box* dan memberi label pada setiap *bounding box* tersebut. Kemudian, hasil anotasi akan disimpan dalam file .txt, yang berisi koordinat dari setiap *bounding box* dan label objek. File .txt ini yang akan digunakan dalam proses *training* model YOLO. Dengan menggunakan Roboflow, proses anotasi data menjadi lebih

mudah dan efisien. Selain itu, aplikasi ini juga memungkinkan pengguna untuk mengaplikasikan teknik augmentasi data, yang dapat memperbanyak variasi data dan membantu meningkatkan performa model.

Dataset yang sudah dianotasi akan memiliki file .txt yang berisi koordinat anotasi tiap gambar menggunakan web aplikasi bernama roboflow, *dataset* tersebut akan dilanjutkan ke tahap *training* yang akan menggunakan *pretrained model* dan *file config* dari YOLO. setelah proses *training* selesai, YOLO akan menghasilkan *file* model hasil *training* dengan format .pt yang digunakan sebagai model kustom baru. jika YOLO menggunakan model tersebut, objek pendekripsi sesuai dataset akan terdeteksi jika ada pada *display* yang memberikan *bounding box*, *label*, dan nilai *confidence* dari objek tersebut.

4.1.1.2. Implementasi

Untuk melatih model YOLOv7, ada beberapa langkah yang harus dilakukan, termasuk pemasangan modul, *pretrained object detection*, *training on custom data*, dan inferensi menggunakan *custom weights*. Langkah pertama adalah menginstal semua modul yang diperlukan, yang dapat dilakukan dengan mengkloning *repository* YOLOv7 dan menginstal *library* yang diperlukan. Langkah selanjutnya adalah mengunduh *weight* yang telah dilatih sebelumnya dan kumpulan data khusus. Setelah itu, model dapat dilatih pada kumpulan *custom data* menggunakan *weight* yang telah dilatih sebelumnya. Terakhir, model dapat dievaluasi menggunakan *custom weight*. (<https://blog.roboflow.com/yolov7-breakdown/>)

4.1.1.3. Pengujian

Pengujian pada bagian sub-sistem ini dilakukan dengan cara membandingkan *Performance* tiap *pretrained model* yolo yang digunakan untuk *training custom model*.

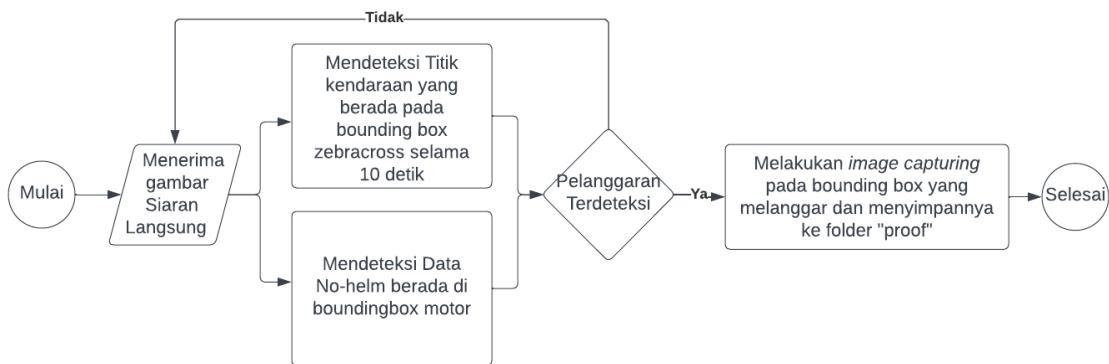
Tabel 4.1 pengujian model yolo

Model	APtest	AP50test	AP75test	batch 1 fps	batch 32 average time
YOLOv7	51.4%	69.7%	55.9%	161 fps	2.8 ms
YOLOv7-X	53.1%	71.2%	57.8%	114 fps	4.3 ms
YOLOv7-W6	54.9%	72.6%	60.1%	84 fps	7.6 ms
YOLOv7-E6	56.0%	73.5%	61.2%	56 fps	12.3 ms
YOLOv7-D6	56.6%	74.0%	61.8%	44 fps	15.0 ms
YOLOv7-E6E	56.8%	74.4%	62.1%	36 fps	18.7 ms

4.1.2 Mendeteksi Pelanggaran

4.1.2.1. Cara Kerja Sub-Sistem

Berdasarkan gambar tersebut sudah terlihat hasil dari program yang sudah berjalan sesuai dengan yang diharapkan.



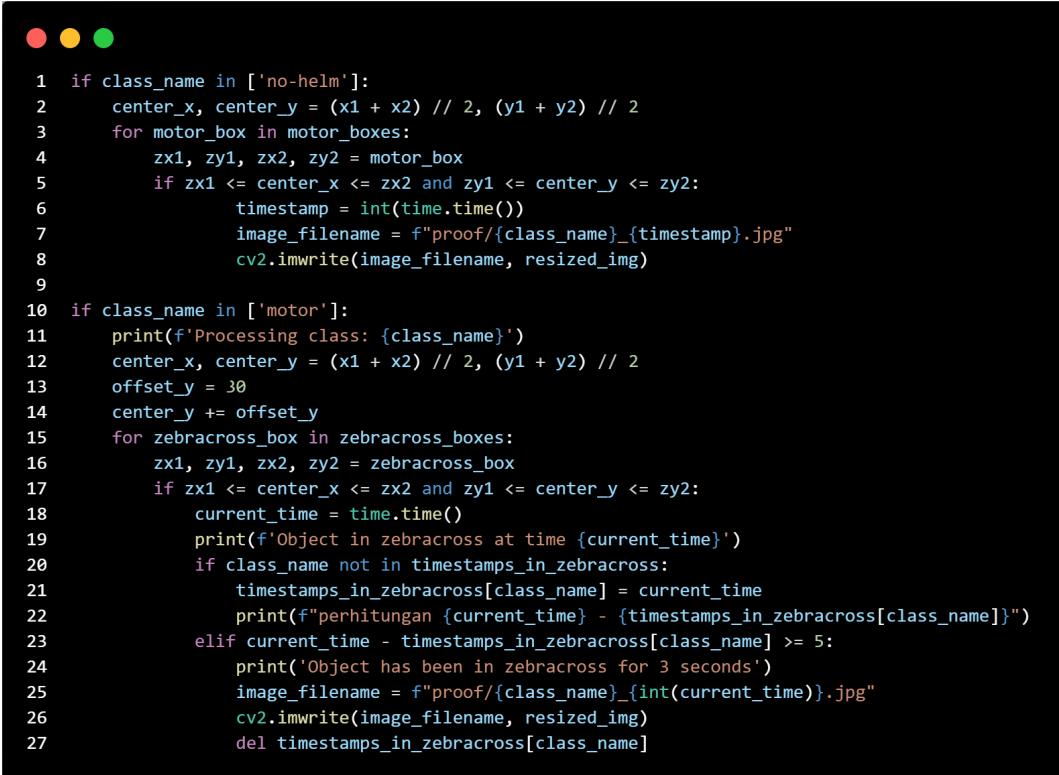
Gambar 4.3 Diagram alur pendekripsi

Gambar 4.3 merupakan flowchart dari tahap tahap yang akan dilakukan pada sistem. Ketika sistem sudah dimulai maka akan mendeteksi apakah adanya pelanggaran yang terjadi. hasil dari pendekripsi ini adalah berbentuk foto pelanggaran tersebut.

4.1.2.2. Implementasi

Pada tahap pengimplementasian sistem pelanggaran ini akan dilakukan proses ekstrak koordinat *bounding box* pada *class no-helm*, motor, mobil, dan zebrecross. data video yang telah dikumpulkan akan digunakan sebagai pengujian, hal ini dilakukan untuk melakukan *testing source code* yang digunakan.

Berikut *Source Code* :



```
1 if class_name in ['no-helm']:
2     center_x, center_y = (x1 + x2) // 2, (y1 + y2) // 2
3     for motor_box in motor_boxes:
4         zx1, zy1, zx2, zy2 = motor_box
5         if zx1 <= center_x <= zx2 and zy1 <= center_y <= zy2:
6             timestamp = int(time.time())
7             image_filename = f"proof/{class_name}_{timestamp}.jpg"
8             cv2.imwrite(image_filename, resized_img)
9
10 if class_name in ['motor']:
11     print(f'Processing class: {class_name}')
12     center_x, center_y = (x1 + x2) // 2, (y1 + y2) // 2
13     offset_y = 30
14     center_y += offset_y
15     for zebrecross_box in zebrecross_boxes:
16         zx1, zy1, zx2, zy2 = zebrecross_box
17         if zx1 <= center_x <= zx2 and zy1 <= center_y <= zy2:
18             current_time = time.time()
19             print(f'Object in zebrecross at time {current_time}')
20             if class_name not in timestamps_in_zebrecross:
21                 timestamps_in_zebrecross[class_name] = current_time
22                 print(f'perhitungan {current_time} - {timestamps_in_zebrecross[class_name]}')
23             elif current_time - timestamps_in_zebrecross[class_name] >= 5:
24                 print('Object has been in zebrecross for 3 seconds')
25                 image_filename = f"proof/{class_name}_{int(current_time)}.jpg"
26                 cv2.imwrite(image_filename, resized_img)
27                 del timestamps_in_zebrecross[class_name]
```

Gambar 4.4 *source code* untuk mendeteksi pelanggaran

Tujuan potongan *code* pada *source code* adalah untuk mendapatkan informasi tentang pengendara yang tidak memakai helm dan berhenti di atas garis zebra cross selama 3 detik. Setelah pengolahan informasi selesai, hasil dari percobaan tersebut dalam bentuk analisis hasil pengujian.

4.1.2.3. Pengujian

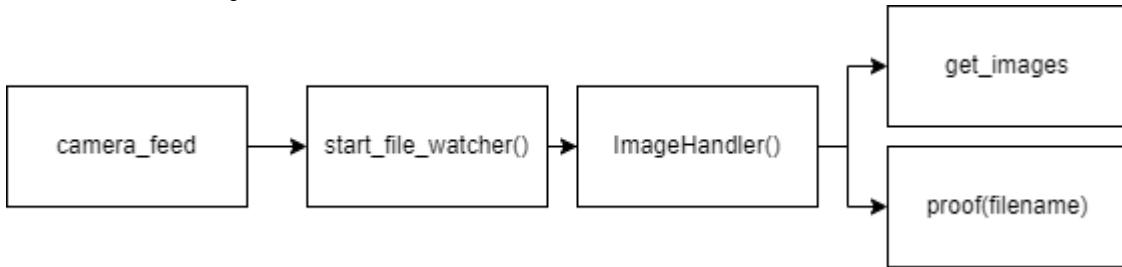
Pada pengujian ini dilakukan langsung menggunakan *source code* terpisah pada aplikasi utama. hal ini diawali dari pengumpulan video pelanggaran lalu lintas. Pengumpulan video tersebut akan diuji satu per satu pada *software code editor* dan memantau hasil *image capture* yang dihasilkan dari kode tersebut.

4.1.2.4. Langkah Pengujian

- Menangkap video *live streaming* dari drone
- Mendeteksi apabila titik pada *bounding box* kendaraan berhenti selama 5 detik di dalam *bounding box* zebra cross, dan pengendara tidak menggunakan helm.
- Sistem akan melakukan *image capturing* video itu lalu mengirimkan hasilnya ke sistem agar dilanjutkan ke proses selanjutnya.

4.1.3 Webapp

4.1.3.1. Cara Kerja Sub-Sistem



Gambar 4.5 alur kerja sub-sistem

Gambar tersebut menampilkan bagian kode dari aplikasi berbasis web yang dirancang dengan memanfaatkan teknologi *machine learning* dan kamera untuk memantau kondisi lalu lintas. Aplikasi ini dibangun dengan menggunakan React JS dan Python, dua bahasa pemrograman yang populer dan kuat. React JS, sebuah library JavaScript yang dikembangkan oleh Facebook, digunakan untuk membangun antarmuka pengguna dari aplikasi ini. Dengan React JS, aplikasi ini dapat memberikan pengalaman pengguna yang dinamis dan responsif, memungkinkan informasi lalu lintas untuk diperbarui secara *real-time* pada tampilan aplikasi. Di sisi lain, Python digunakan sebagai back-end dari aplikasi ini, khususnya dalam penerapan algoritma *machine learning*. Python adalah pilihan yang sangat tepat untuk ini karena memiliki sejumlah library, seperti TensorFlow dan PyTorch, yang memudahkan implementasi dan peningkatan model *machine learning*.

4.1.3.2. Implementasi

Pada bagian `camera_feed` sistem mengambil setiap frame dari aliran video, melakukan deteksi objek, dan memberikan penanda pada frame berdasarkan hasil deteksi. Setelah deteksi, aplikasi akan melakukan berbagai aksi berdasarkan objek yang ditemukan. Fungsi ini mencari beberapa kategori objek tertentu seperti 'platnomor', 'no-helm', 'motor', 'mobil', dan 'zebracross'. pada bagian *route* terdapat pendekripsi pelanggaran secara langsung, jika objek 'no-helm' ditemukan dalam objek 'motor', fungsi ini akan menyimpan gambar tersebut sebagai bukti pelanggaran. bersamaan dengan objek 'motor' terdeteksi berada dalam objek 'zebracross' lebih dari 5 detik, gambar tersebut juga akan disimpan.

Kode ini juga menginisiasi pemantauan menggunakan fungsi '`start_file_watcher()`' pada direktori `image_folder` dan akan menjalankan tindakan '`ImageHandler()`' saat terjadi penambahan foto pada file di direktori tersebut. dalam fungsi '`start_file_watcher()`' membuat instance dari kelas *Observer* yang akan memulai proses pemantauan. Selain itu, fungsi '`ImageHandler()`' yang bertindak sebagai pengelola peristiwa pada sistem file, yang berisi fungsi '`process_and_rename_number_plate()`' untuk melakukan pendekripsi huruf atau digit pada platnomor. Saat instansiasi `ImageHandler` dibuat, sistem ini juga menginisialisasi antrian dan thread baru. Antrian ini dirancang untuk menampung jalur dari setiap gambar baru yang dibuat, sementara thread yang dibuat akan bertugas untuk memproses gambar di antrian tersebut. Secara keseluruhan, aplikasi ini dirancang untuk memantau setiap pembuatan file gambar baru dalam suatu direktori, menambahkan jalur file

tersebut ke dalam antrian, dan memprosesnya secara asinkron dalam *thread* yang terpisah.

Sedangkan, disediakan dua jalur API tambahan ditetapkan di dalam *main code* yang akan mengirimkan informasi ke *frontend*. dua *route* tersebut yaitu /proof/:filename dan /get_images. Fungsi proof() adalah untuk mengambil dan mengirimkan kembali file yang tersimpan di direktori "proof" berdasarkan nama file yang menjadi parameter. Sedangkan fungsi get_images() bertugas untuk menghasilkan daftar gambar yang disimpan dalam server. Fungsi ini mencari semua file dengan ekstensi .jpg dalam direktori yang ditentukan (image_folder), merangkai URL lengkap untuk setiap gambar berdasarkan alamat server, dan mengirim kembali semua data ini sebagai respons dalam format JSON.

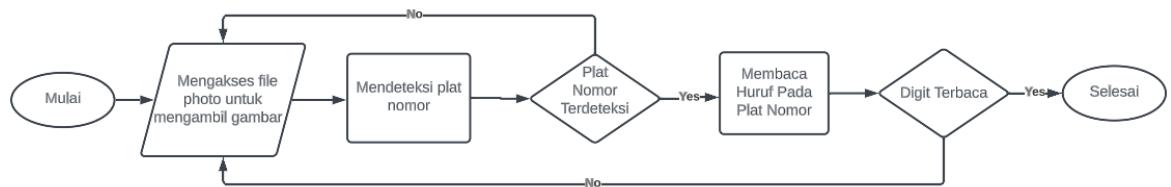
4.1.3.3. Pengujian

Pengujian dilakukan dengan mendeteksi letak titik pada *bounding box* kendaraan, kendaraan yang berada di atas zebra cross lebih dari parameter waktu yang ditentukan maka akan diproses menggunakan yolov7 untuk dianalisa apakah ada pelanggaran yang terdeteksi.

4.1.4 Mendeteksi Plat Nomor pada Kendaraan Bermotor

4.1.4.1. Cara Kerja Sub-Sistem

Berdasarkan gambar tersebut sudah terlihat hasil dari program yang sudah berjalan sesuai dengan yang diharapkan.



Gambar 4.6 Diagram Alur Sistem

Gambar 4.6 merupakan *flowchart* dari tahap-tahap yang akan dilakukan pada sistem. Ketika sistem sudah dimulai maka akan mencoba mendeteksi objek yang sebelumnya sudah ada di dataset, hasil akhir akan menganalisa hasil dari objek tersebut.

4.1.4.2. Implementasi

Implementasi yang dilakukan pada sub-sistem ini diawali dengan mengumpulkan beberapa foto sampel yang akan diberikan *bounding box* untuk dipotong oleh program, setelah itu digit akan dibaca oleh *pytesseract* untuk dijadikan *result*, dan *result* tersebut akan menjadi nama file dari *output* yang dihasilkan.



Gambar 4.7 Foto awal mobil dengan plat nomor



Gambar 4.8 Foto mobil dengan plat nomor yang sudah dipotong



Gambar 4.9 Foto mobil dengan plat nomor yang sudah di *invert color*

4.1.4.3. Pengujian

Pengujian pada sistem pendekripsiplat nomor ini dilakukan dengan cara memasukkan foto input contoh yang digunakan untuk mendekripsi plat nomor. lalu sistem akan mendekripsi dan membaca plat nomor yang ada pada foto yang telah di input.

4.2 Analisis Pengerjaan Implementasi Sistem

4.2.1. Sub-sistem 1 (Dataset)

Dataset foto kebutuhan yang kami kumpulkan berjumlah 1000 foto, dari foto tersebut mencakup 7 jenis *class* yang diambil menggunakan berbagai skenario di jalan raya. anotasi data yang dihasilkan terbagi seimbang dengan jumlah 150 sampai 200 anotasi dalam total 1000 foto.

Rincian data adalah sebagai berikut:

- zebracross: Gambar ini mencakup berbagai jenis penyeberangan zebra di berbagai kondisi pencahayaan dan sudut pandang.

- helm: Ini mencakup berbagai jenis helm yang dikenakan oleh pengendara sepeda motor dalam berbagai kondisi dan pengaturan.
- no-helm: Kumpulan ini mencakup pengendara sepeda motor yang tidak mengenakan helm, dalam berbagai kondisi dan pengaturan.
- mobil: Gambar ini mencakup berbagai jenis dan model mobil, dalam berbagai kondisi pencahayaan dan sudut pandang.
- motor: Gambar ini mencakup berbagai jenis dan model sepeda motor yang dikendarai, dalam berbagai kondisi pencahayaan dan sudut pandang.
- platnomor: Kumpulan ini mencakup berbagai jenis plat nomor dari berbagai jenis kendaraan dan dalam berbagai kondisi pencahayaan dan sudut pandang.
- pejalan kaki: Ini mencakup berbagai jenis individu yang berjalan kaki dalam berbagai kondisi dan pengaturan.

Hasil dari peraturan yang dibuat untuk penggunaan dataset sangat berpengaruh dalam masalah klasifikasi motor dengan pejalan kaki.

4.2.2. Mendeteksi Pelanggaran

Mengenai kasus pelanggaran zebracross, sistem yang telah dirancang mampu mendeteksi pengendara yang melakukan pelanggaran dengan memanfaatkan *bounding* titik di posisi tengah. Jika pengendara berhenti lebih dari 5 detik di area zebracross, ini dapat ditafsirkan sebagai pelanggaran. Sama halnya dengan kasus pelanggaran pengendara motor yang tidak memakai helm. Sistem ini mampu mendeteksi jika *bounding box* dari kelas *no-helm* berada di dalam salah satu *bounding box* kelas motor. Dengan demikian, sistem dapat mengidentifikasi pengendara motor mana yang tidak memakai helm dan dianggap melakukan pelanggaran. Keunggulan sistem ini adalah kemampuannya untuk mendeteksi dan menangkap pelanggaran secara *real-time*, yang berarti bahwa penegakan hukum dapat dilakukan lebih cepat dan lebih efisien.

4.2.3. Webapp

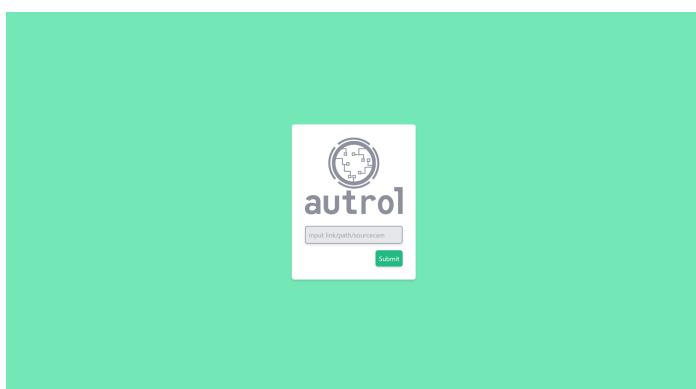
Berbagai komponen web aplikasi dirancang dan dikembangkan. Ini melibatkan proses merencanakan dan merancang antarmuka pengguna dan arsitektur sistem secara keseluruhan. Desain antarmuka pengguna harus intuitif dan mudah digunakan. Di sisi lain, arsitektur sistem perlu dirancang sedemikian rupa untuk memastikan bahwa web aplikasi dapat berfungsi dengan baik dan dapat diandalkan. dengan

demikian, hasil mulai dari input jenis video sampai output modal notifikasi berjalan dengan lancar, belum ada interupsi dari *error*.

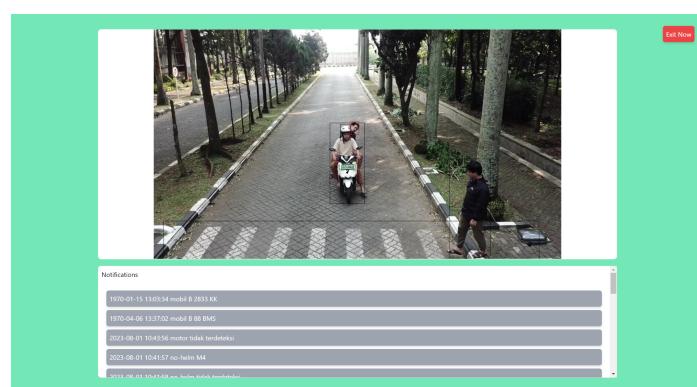
4.2.4. Mendeteksi dan Membaca Plat Nomor

Ketika mendeteksi plat nomor, sistem akan menggunakan fungsi dari YOLOv7 dengan menggunakan *dataset* yang sudah di *training* pada proses sub-sistem sebelumnya. Setelah plat nomor telah dideteksi, maka plat tersebut akan dibaca menggunakan teknologi OCR, digit dari plat nomor mampu dibaca dengan tingkat akurasi yang cukup tinggi. Untuk program OCR yang digunakan disini adalah *pytesseract*. Menggunakan *pytesseract* perlu untuk melalui berbagai macam progress, dengan memotong khusus pada daerah plat nomor, lalu *meng-invert* warna dari plat nomor tersebut untuk mendapatkan kontur dari digit plat nomor tersebut. Setelah semua proses tersebut dilakukan, maka didapatkan lah digit dari plat nomor tersebut.

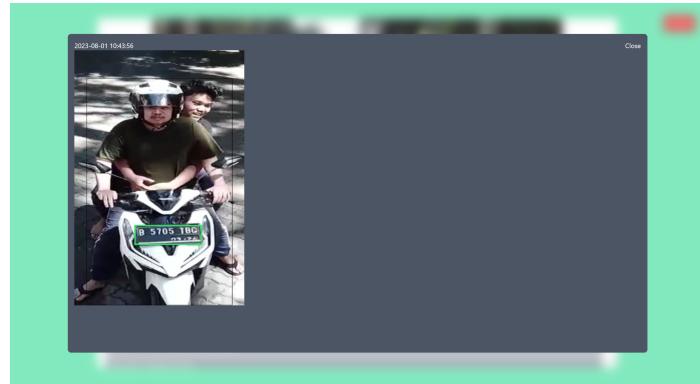
4.3 Hasil Akhir Sistem



Gambar 4.10 Foto halaman *landing page* dari website



Gambar 4.11 Foto halaman *main page* dari website



Gambar 4.12 Foto halaman *landing page* dari *website*

Pada gambar, Sistem pendekripsi pelanggaran lalu lintas yang menggabungkan teknologi drone, aplikasi web berbasis React JS dan Python, dan algoritma *machine learning*, telah berhasil diimplementasikan dengan hasil yang memuaskan. Fungsionalitas utama sistem ini, termasuk pemantauan lalu lintas melalui drone, analisis gambar menggunakan *machine learning*, dan penyampaian informasi real-time kepada pihak berwenang, telah berjalan dengan efisien dan efektif. Meski menghadapi tantangan dalam kondisi cuaca atau pencahayaan yang kurang ideal, sistem ini tetap menunjukkan kinerja yang stabil dan akurasi deteksi pelanggaran lalu lintas yang tinggi. Antarmuka pengguna aplikasi web, yang responsif dan intuitif, memungkinkan pemantauan kondisi lalu lintas secara efektif, sementara integrasi dengan *database* dan sistem pelaporan lainnya telah berhasil mengotomatiskan proses pengiriman notifikasi dan denda. Seluruh sistem telah dikembangkan dan diuji sesuai dengan standar teknologi dan keamanan yang berlaku, menegaskan bahwa implementasi ini bukan hanya sukses, tetapi juga aman dan sesuai dengan regulasi yang berlaku. Meskipun telah berhasil, sistem ini masih membuka peluang untuk peningkatan dan perbaikan di masa mendatang.

4.4 Kesimpulan dan Ringkasan CD-4

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa dalam merancang sistem kami telah melakukan berbagai macam implementasi dan pengujian pada setiap sub-sistem, dari implementasi dan pengujian tersebut kami mampu mendapatkan hasil seperti dataset yang siap untuk diuji dalam mencari model YOLO yang terbaik. Begitu juga dengan implementasi pada program pendekripsi pelanggaran, webapp, dan OCR.

BAB 5

PENGUJIAN SISTEM

5.1 Skema Pengujian Sistem

5.1.1 Pengujian Model

Proses pengujian sistem parameter YOLO (*You Only Look Once*) akan dilakukan dengan memanfaatkan berbagai parameter dan skenario. Dimulai dengan penentuan rasio data *training* dan validasi, kita akan mencoba rasio 58%:42%, 68%:32%, 78%:22%, 88%:12%, dan 98%:2%. Untuk setiap rasio, bagian yang lebih besar dari data akan digunakan untuk melatih model dan bagian yang lebih kecil untuk validasi. Setelah pembagian data, kita akan menguji model menggunakan berbagai *learning rate* yaitu 0.001, 0.0001, 0.00001, dan 0.000001. *Learning rate* ini akan membantu kita mengendalikan seberapa cepat atau lambat model kita belajar dari data. Selanjutnya, kita akan menguji model dengan berbagai ukuran *batch* yaitu 2, 4, dan 8. Ukuran *batch* ini menentukan jumlah sampel yang diproses sebelum model memperbarui parameter internalnya. Terakhir, kita akan menguji model dengan jumlah *epochs* yang berbeda yaitu 100, 200, dan 300. Jumlah *epochs* ini menentukan berapa kali model akan melihat seluruh set data latihan. Semua pengujian ini akan dilakukan dalam lingkungan komputasi yang stabil dan idealnya pada komputer dengan spesifikasi yang memadai untuk menjalankan model *deep learning*. Sepanjang proses pengujian ini, kita akan memantau berbagai metrik performa seperti presisi, *recall*, dan *F1-score*, dan berusaha mencapai kombinasi parameter yang memberikan hasil terbaik berdasarkan metrik tersebut.

5.1.2 Pengujian Sistem

Untuk menguji sistem, terlebih dahulu kita perlu menciptakan lingkungan yang aman dan terkendali. Hal ini dapat berupa ruangan tertutup atau area terbuka yang cukup luas bagi drone untuk bergerak pada ketinggian yang berbeda, bebas dari hambatan dan interferensi yang tidak diinginkan. Selanjutnya, merancang rangkaian skenario pengujian untuk mencakup semua variabel yang ingin kita uji, seperti ketinggian 3 meter, 5 meter, dan 7 meter, dengan sudut kamera 45, 60, dan 90 derajat, serta berbagai kecepatan drone.

Pada tahap pelaksanaan, skenario pengujian tersebut harus dijalankan, yang melibatkan pengoperasian drone pada ketinggian dan kecepatan yang ditentukan,

sambil merotasi kamera drone pada sudut yang ditentukan. Selama proses ini, perhatian harus diberikan pada drone dan sistem secara keseluruhan untuk memastikan tidak ada masalah yang muncul.

Selama pengujian, sistem harus merekam data yang relevan seperti kualitas gambar dari kamera, *respons* drone terhadap perintah, dan lainnya. Setelah pengujian selesai, data ini harus dianalisis untuk melihat bagaimana performa sistem di bawah berbagai kondisi.

Berdasarkan hasil analisis ini, kemungkinan akan ditemukan beberapa area yang perlu disempurnakan pada sistem. Perubahan akan dilakukan dan pengujian akan diulangi. Proses ini berulang-ulang sampai sistem mencapai level performa yang diinginkan. Ini adalah proses iteratif dan kontinu yang memastikan sistem berfungsi dengan baik di berbagai kondisi operasional.

5.2 Proses Pengujian

5.2.1 Pengujian 1: Pengujian Model

Pada pengujian metode yang pertama ini mengenai bagaimana metode deteksi objek menggunakan YOLO memberikan hasil tingkat kepercayaan yang tinggi terhadap objek yang dilatih pada model tersebut.

5.2.1.1. *Ratio Training Data*

Pada pengujian model yang pertama, kami menggunakan parameter yang sudah ditentukan. Diantaranya menggunakan:

- *Learning rate = 0,00001*
- *Batch size = 8*
- *Epoch = 100*

Berikut adalah hasil pengujian rasio data.

a. Rasio Data 58% : 42%

Training dataset menggunakan rasio 58% *data train* dan 42% untuk *data valid* dan *test* dari *dataset*.

Tabel 5.1 nilai rata-rata pada rasio data 58% : 42%

Parameter	Nilai Rata-Rata 0.573
<i>Annotation</i>	73
<i>Precision</i>	0.613
<i>Recall</i>	0.616
<i>F1-Score</i>	0.614
<i>mAP@[0.5]</i>	0.577

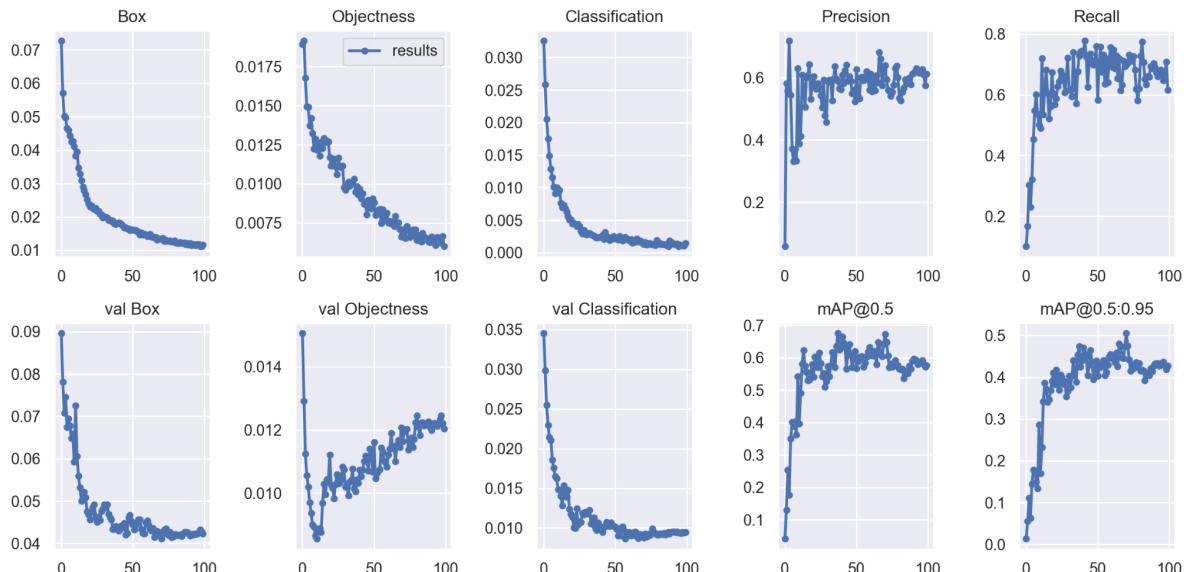
Parameter	Nilai Rata-Rata 0.573
$mAP @[0.5:0.95]$	0.427

```

Epoch    gpu_mem      box      obj      cls      total      labels      img_size
98/99    7.87G  0.01117  0.006652  0.001083  0.0189      20      640: 100%|████| 73/73 [00:19<00:00, 3.75
          Class      Images      Labels      P      R      mAP@.5  mAP@.5:.95: 100%|████| 14/14 [00:08<
          all       210        442      0.575      0.709      0.572      0.419
99/99    7.87G  0.01164  0.006014  0.001493  0.01915     34      640: 100%|████| 73/73 [00:19<00:00, 3.76
          Class      Images      Labels      P      R      mAP@.5  mAP@.5:.95: 100%|████| 14/14 [00:09<
          all       210        442      0.613      0.616      0.577      0.427
          helm      210         12      0.257      0.667      0.244      0.191
          mobil      210        244      0.943      0.744      0.84      0.729
          motor      210        17      0.256      0.587      0.329      0.297
          no-helm    210         4      0.528      0.75      0.653      0.456
          pejalankaki 210        65      0.69      0.24      0.448      0.299
          platnomor   210        56      0.777      0.857      0.888      0.623
          zebracross  210        44      0.838      0.471      0.638      0.392
100 epochs completed in 0.815 hours.

```

Gambar 5.1 nilai rata-rata pada rasio data 58% : 42%



Gambar 5.2 grafik nilai rata-rata pada rasio data 58% : 42%

b. Rasio Data 68% : 32%

Training dataset menggunakan rasio 68% *data train* dan 32% untuk data *valid* dan *test* dari dataset.

Tabel 5.2 nilai rata-rata pada rasio data 68% : 32%

Parameter	Nilai Rata-Rata 0.5926
<i>Annotation</i>	85
<i>Precision</i>	0.565
<i>Recall</i>	0.694

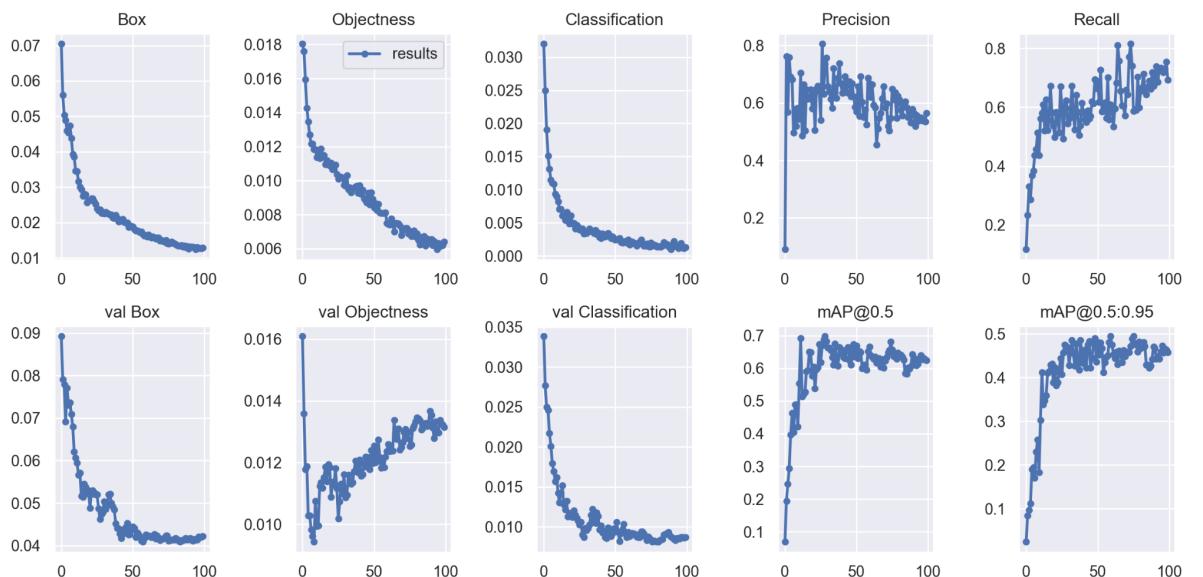
Parameter	Nilai Rata-Rata 0.5926
<i>F1-Score</i>	0.622
<i>mAP@[0.5]</i>	0.624
<i>mAP @[0.5:0.95]</i>	0.458

```

Epoch    gpu_mem      box      obj      cls      total      labels      img_size
99/99    8.27G  0.01293  0.006414  0.0013   0.02065      49      640: 100%||| 85/85 [00:23<00:00, 3.6
Class     Images     Labels      P       R
all       161        360      0.565   0.694      0.624      0.458
helm      161         12      0.267   0.667      0.293      0.209
mobil     161        239      0.967   0.726      0.846      0.749
motor     161         17      0.297   0.588      0.437      0.358
no-helm   161          4      0.479   0.1      0.77      0.421
pejalankaki 161        37      0.785   0.396      0.587      0.428
platonmor  161        48      0.785   0.812      0.856      0.583
zebracross 161         3      0.376   0.667      0.583      0.458
100 epochs completed in 0.954 hours.

```

Gambar 5.3 nilai rata-rata pada rasio data 68% : 32%



Gambar 5.4 grafik nilai rata-rata pada rasio data 68% : 32%

c. Rasio Data 78% : 22%

Training dataset menggunakan rasio 78% *data train* dan 22% untuk data *valid* dan *test* dari *dataset*.

Tabel 5.3 nilai rata-rata pada rasio data 78% : 22%

Parameter	Nilai Rata-Rata 0.6032
<i>Annotation</i>	98
<i>Precision</i>	0.651

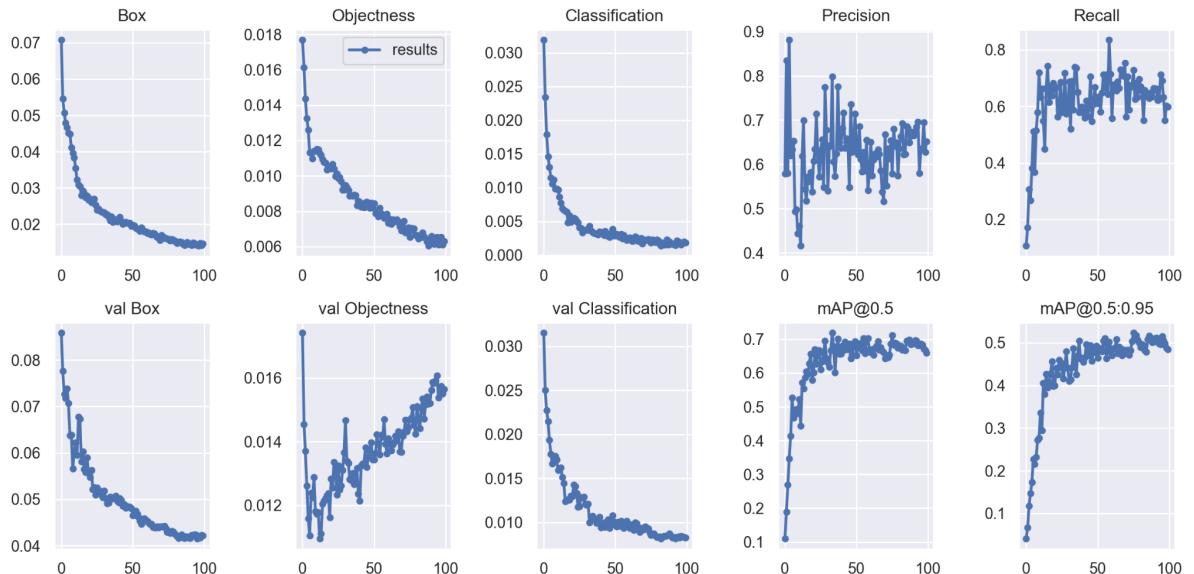
<i>Recall</i>	0.598
<i>F1-Score</i>	0.623
<i>mAP@[0.5]</i>	0.66
<i>mAP @[0.5:0.95]</i>	0.484

```

Epoch    gpu_mem      box      obj      cls      total      labels      img_size
99/99    7.93G  0.01459  0.006319  0.001848  0.02275     15      640: 100%||| 98/98 [00:26<00:00,  3.6
Class   Images      Labels      P          R          mAP@.5  mAP@.5:.95: 100%||| 7/7 [00:04<0
      all       111       296      0.651      0.598      0.66      0.484
      helm      111        12      0.314      0.583      0.34      0.218
      mobil      111       185      0.974      0.601      0.843      0.733
      motor      111        17      0.384      0.587      0.499      0.396
      no-helm    111         4      0.884      0.75      0.888      0.53
      pejalankaki 111        29      0.855      0.407      0.644      0.48
      platonmor   111       47      0.798      0.756      0.872      0.619
      zebracross   111        2      0.349      0.5      0.532      0.415
100 epochs completed in 0.876 hours.

```

Gambar 5.5 nilai rata-rata pada rasio data 78% : 22%



Gambar 5.6 grafik nilai rata-rata pada rasio data 78% : 22%

d. Rasio Data 88% : 12%

Training dataset menggunakan rasio 88% *data train* dan 12% untuk data *valid* dan *test* dari *dataset*.

Tabel 5.4 nilai rata-rata pada rasio data 88% : 12%

Parameter	Nilai Rata-Rata 0.7394
<i>Annotation</i>	111
<i>Precision</i>	0.784
<i>Recall</i>	0.746

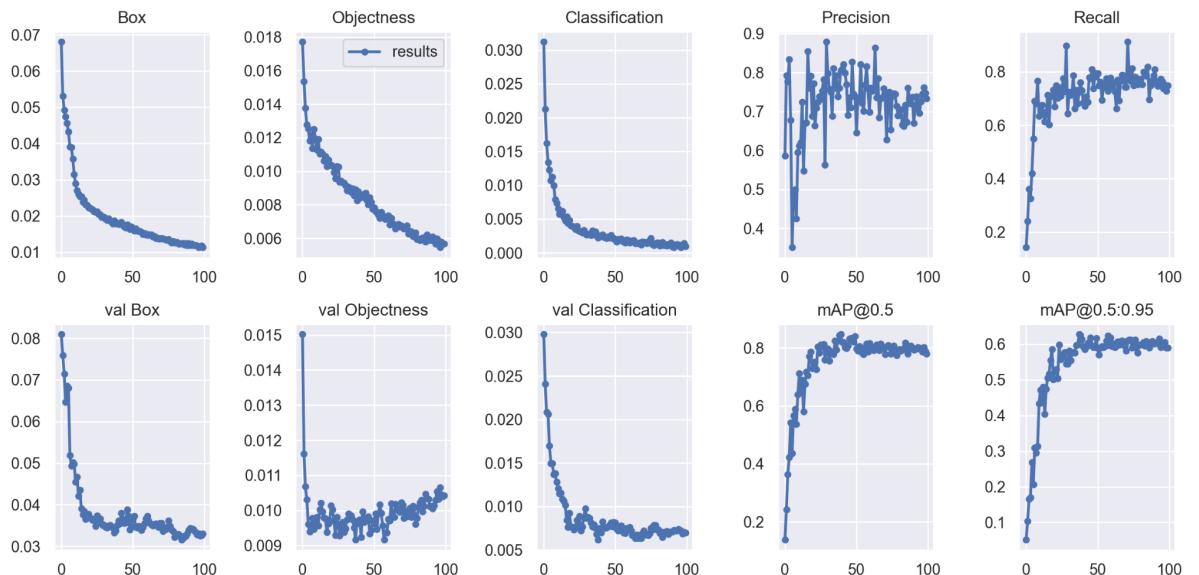
Parameter	Nilai Rata-Rata 0.7394
<i>F1-Score</i>	0.764
<i>mAP@[0.5]</i>	0.8
<i>mAP @[0.5:0.95]</i>	0.603

```

Epoch      gpu mem      box      obj      cls      total      labels      img_size
99/99      8.12G  0.01362  0.00627  0.001414  0.02131      1          640: 100%||| 111/111 [00:29<00:00, 3
Class      Images      Labels      P          R
all        60          141       0.784     0.746      0.8          mAP@.5
helm       60           10       0.885     0.8       0.874      0.563
mobil      60           74       0.97      0.919     0.982      0.904
motor       60           12       0.553     0.833     0.518      0.424
no-helm    60            4       0.948     0.75      0.945      0.564
pejalankaki 60          24       0.833     0.625     0.75      0.525
platnomor   60          15           1       0.793     0.908      0.733
zebracross  60            2       0.295     0.5      0.62      0.508
100 epochs completed in 0.913 hours.

```

Gambar 5.7 nilai rata-rata pada rasio data 88% : 12%



Gambar 5.8 grafik nilai rata-rata pada rasio data 88% : 12%

e. Rasio Data 98% : 2%

Training dataset menggunakan rasio 98% *data train* dan 2% untuk data *valid* dan *test* dari *dataset*.

Tabel 5.5 nilai rata-rata pada rasio data 98% : 2%

Parameter	Nilai Rata-Rata 0.7086
<i>Annotation</i>	123
<i>Precision</i>	0.657
<i>Recall</i>	0.765

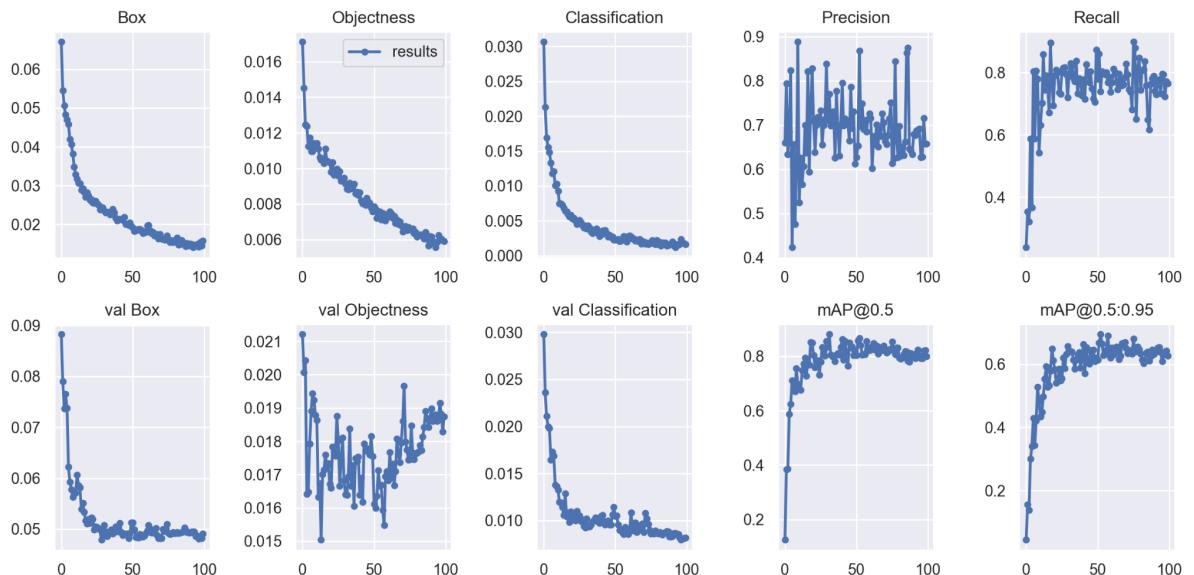
<i>F1-Score</i>	0.697
<i>mAP@[0.5]</i>	0.799
<i>mAP @[0.5:0.95]</i>	0.625

```

Epoch      gpu_mem      box      obj      cls      total      labels      img_size
99/99      6.81G  0.01573  0.005898  0.001639  0.02327      55      640: 100% ||| 123/123 [00:33<00:00,  3
Class      Images      Labels      P          R          mAP@.5  mAP@.5:.95: 100%||| 1/1 [00:00<0
all        10          37          0.657     0.765     0.799     0.625
helm       10          10          0.833     0.5       0.86      0.652
mobil      10          4           0.656     1          0.829     0.758
motor      10          7           0.493     0.857     0.615     0.538
no-helm    10          4           0.949     0.75      0.845     0.491
pejalankaki 10          8           0.804     0.75      0.832     0.611
platnomor  10          2           0.618     1          0.995     0.821
zebracross  10          2           0.247     0.5       0.62      0.588
100 epochs completed in 0.958 hours.

```

Gambar 5.9 nilai rata-rata pada rasio data 98% : 2%



Gambar 5.10 grafik nilai rata-rata pada rasio data 98% : 2%

5.2.1.2. Learning Rate Training Data

Pada pengujian model tahap kedua, kami menambahkan parameter yang terbaik dari pengujian sebelumnya. Parameter yang digunakan adalah rasio data 88% : 12%, dengan menggunakan parameter dibawah ini:

- *Learning rate = (0.001), (0,0001), (0,00001), (0,000001)*
- *Batch size = 8*
- *Epoch = 100*

Berikut adalah hasil pengujian *learning rate* setelah menyesuaikan parameter hasil terbaik dari rasio data.

a. Learning Rate 0.001

Tabel 5.6 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,001

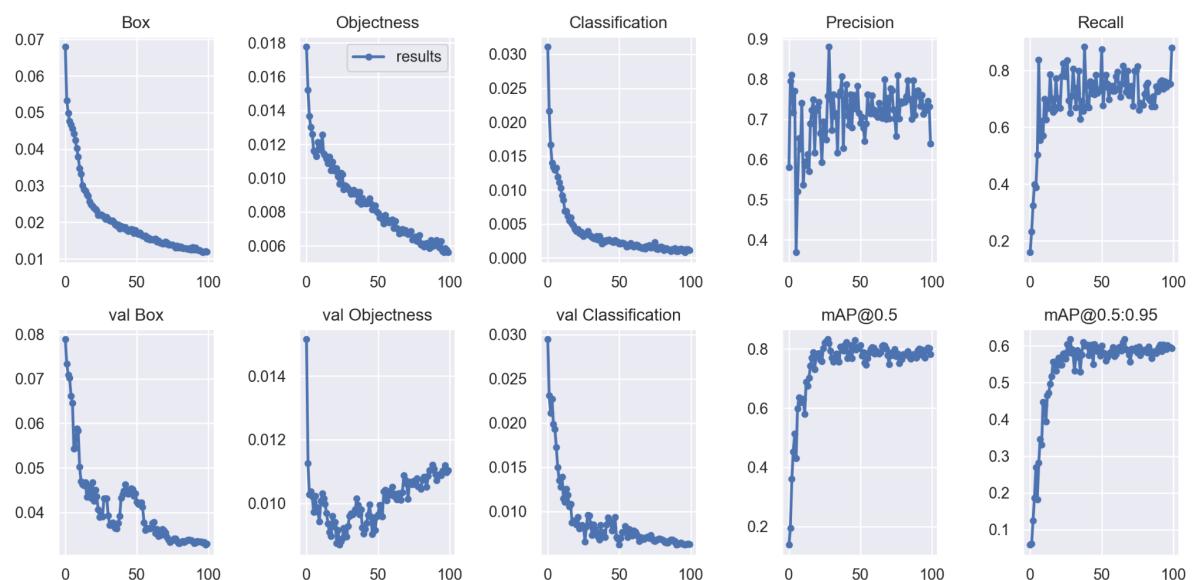
Parameter	Nilai Rata-Rata 0.727
<i>Annotation</i>	111
<i>Precision</i>	0.64
<i>Recall</i>	0.88
<i>F1-Score</i>	0.741
<i>mAP@[0.5]</i>	0.782
<i>mAP @[0.5:0.95]</i>	0.593

```

Epoch 99/99   gpu_mem 8.15G   box    obj    cls   total   labels   img_size
99/99        0.01194 0.005629 0.00112 0.01869      1       640: 100% | 111/111 [00:29<00:00, 3.72it/s]
Class Images Labels          P         R   mAP@.5 mAP@.5: .95: 100% | 4/4 [00:02<00:00, 1.90it/s]
all     60      141    0.64  0.88  0.782  0.593
helm    60      10     0.529 0.897  0.672  0.494
mobil   60      74     0.966 0.959  0.97   0.968
motor   60      12     0.476 0.988  0.571  0.473
no-helm 60      4     0.963 1      0.995  0.529
pejalankaki 60     24     0.695 0.667  0.786  0.495
platnomor 60     15     0.785 0.933  0.84   0.711
zebracross 60      2     0.286 0.793  0.62   0.545
100 epochs completed in 0.905 hours.

```

Gambar 5.11 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,001



Gambar 5.12 grafik nilai rata-rata pada pengujian *learning rate* dengan nilai 0,001

b. *Learning Rate 0.0001*

Pada pengujian pertama, kami mencoba untuk meningkatkan nilai *learning rate*, yang mempunyai nilai *default* 0,00001.

Tabel 5.7 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,0001

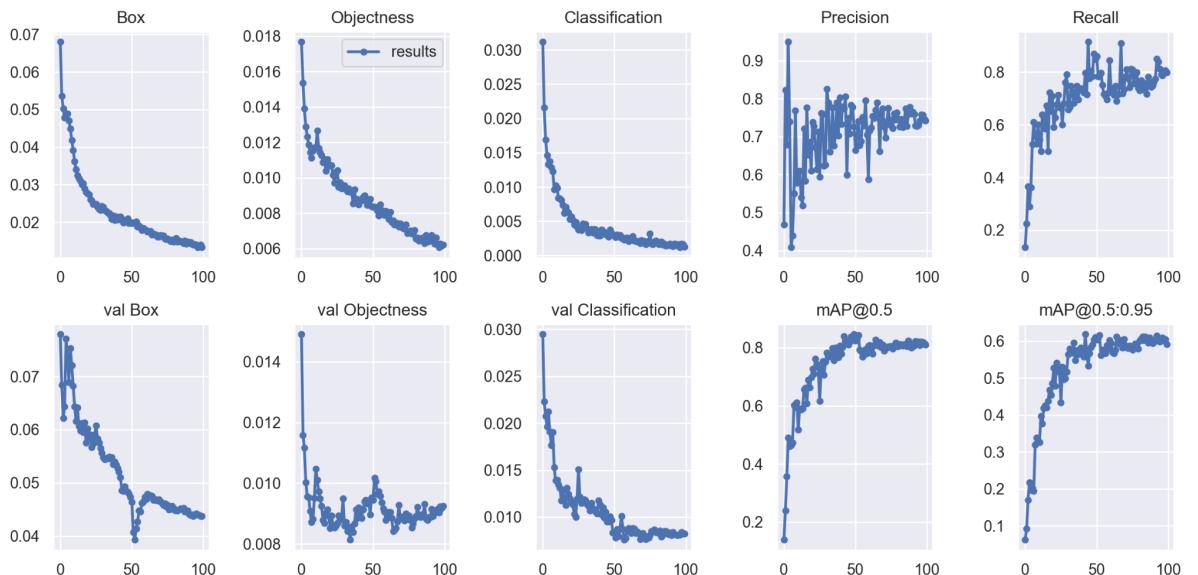
Parameter	Nilai Rata-Rata 0.742
<i>Annotation</i>	111
<i>Precision</i>	0.742
<i>Recall</i>	0.798
<i>F1-Score</i>	0.768
<i>mAP@[0.5]</i>	0.81
<i>mAP @[0.5:0.95]</i>	0.592

```

Epoch    gpu_mem      box      obj      cls      total      labels      img_size
99/99    7.66G  0.01337  0.006232  0.00131  0.02091      1      640: 100%| 111/111 [00:30<00:00,  3
          Class   Images   Labels      P        R      mAP@.5  mAP@.5:95: 100%| 4/4 [00:01<0
          all     60       141     0.742     0.798     0.81    0.592
          helm    60        10     0.9       0.9       0.892    0.574
          mobil   60       74     0.982     0.946     0.982    0.913
          motor   60       12     0.569     0.833     0.564    0.441
          no-helm 60        4     1         0.892     0.995    0.564
          pejalankaki 60       24     0.809     0.583     0.705    0.471
          platonmor 60       15     0.661     0.933     0.934    0.693
          zebracross 60        2     0.27      0.5       0.595    0.486
100 epochs completed in 0.897 hours.

```

Gambar 5.13 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,0001



Gambar 5.14 grafik nilai rata-rata pada pengujian *learning rate* dengan nilai 0,0001

c. *Learning Rate 0.00001*

Pada pengujian kedua kami mencoba untuk menggunakan nilai *default* dari sistem *training YOLOv7*, yaitu 0,00001

Tabel 5.8 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,00001

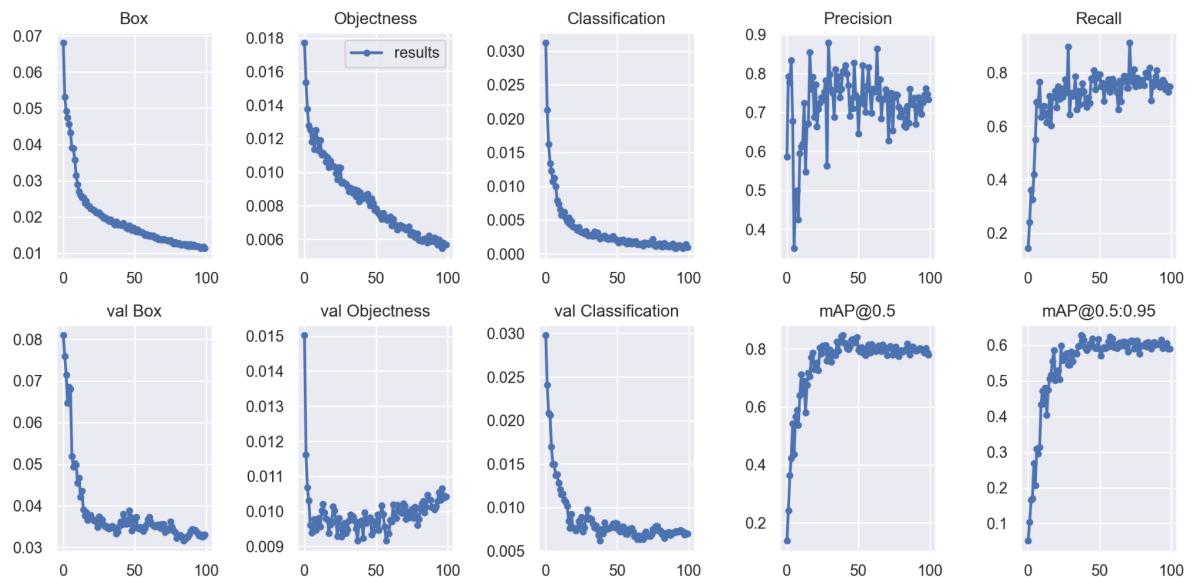
Parameter	Nilai Rata-Rata 0.7394
<i>Annotation</i>	111
<i>Precision</i>	0.784
<i>Recall</i>	0.746
<i>F1-Score</i>	0.764
<i>mAP@[0.5]</i>	0.8
<i>mAP @[0.5:0.95]</i>	0.603

```

Epoch    gpu_mem      box      obj      cls      total      labels      img_size
99/99    8.12G  0.01362  0.00627  0.001414  0.02131      1      640: 100%||| 111/111 [00:29<00:00,  3
          Class   Images   Labels      P        R      mAP@.5  mAP@.5:95: 100%||| 4/4 [00:02<0
          all     60       141      0.784    0.746     0.8      0.8      0.603
          helm    60       10       0.885    0.8      0.8      0.874     0.563
          mobil   60       74       0.97     0.919     0.982     0.982     0.904
          motor   60       12       0.553    0.833     0.518     0.518     0.424
          no-helm 60       4        0.948    0.75      0.945     0.945     0.564
          pejalankaki 60       24       0.833    0.625     0.75      0.75      0.525
          platonmor 60       15       1        0.793     0.988     0.988     0.733
          zebracross 60       2        0.295    0.5      0.62      0.62      0.508
100 epochs completed in 0.913 hours.

```

Gambar 5.15 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,00001



Gambar 5.16 grafik nilai rata-rata pada pengujian *learning rate* dengan nilai 0,00001

d. *Learning Rate 0.000001*

Pada pengujian pertama, kami mencoba untuk menurunkan nilai *learning rate*, dari yang mempunyai nilai *default* 0,00001 menjadi 0,000001.

Tabel 5.9 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,00001

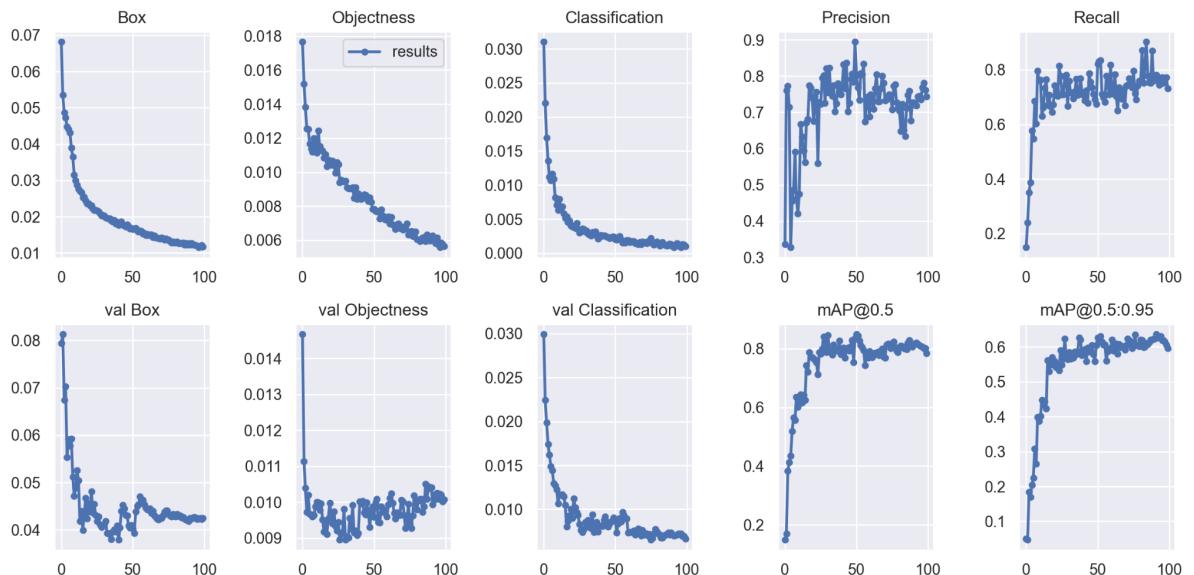
Parameter	Nilai Rata-Rata 0.7182
<i>Annotation</i>	111
<i>Precision</i>	0.743
<i>Recall</i>	0.732
<i>F1-Score</i>	0.737
<i>mAP@[0.5]</i>	0.784
<i>mAP @[0.5:0.95]</i>	0.595

```

Epoch    gpu_mem      box      obj      cls      total      labels      img_size
99/99    7.92G  0.01175  0.005657  0.001013  0.01842      1      640: 100%|██████████| 111/111 [00:29<00:00,  3
Class   Images       Labels
      all      60      141      0.743      0.732      0.784      mAP@.5: 0.595
      helm     60       10      0.714      0.501      0.668      mAP@.5..95: 100%|██████████| 4/4 [00:02<0
      mobil     60       74      0.971      0.898      0.961      0.906
      motor     60       12      0.572      0.833      0.634      0.499
      no-helm   60        4      1.000      0.995      0.995      0.57
      pejalankaki 60       24      0.766      0.458      0.691      0.498
      platonmor   60       15      0.873      0.933      0.922      0.724
      zebrecross   60        2      0.306      0.500      0.62       0.496
100 epochs completed in 0.905 hours.

```

Gambar 5.17 nilai rata-rata pada pengujian *learning rate* dengan nilai 0,00001



Gambar 5.18 grafik nilai rata-rata pada pengujian *learning rate* dengan nilai 0,00001

5.2.1.3. Batch-size Training Data

Pada pengujian model tahap ketiga, kami kembali memasukan parameter yang sudah dihitung dan dianalisis dari proses pengujian sebelumnya, dengan mengubah nilai *learning rate* yang terbaik, yaitu:

- *Learning rate = 0,0001*
- *Batch size = 2, 4, 8*

- **Epoch = 100**

Berikut adalah hasil pengujian *batch-size* setelah menyesuaikan parameter hasil terbaik dari rasio data dan *learning rate*.

a. **Batch-size 2**

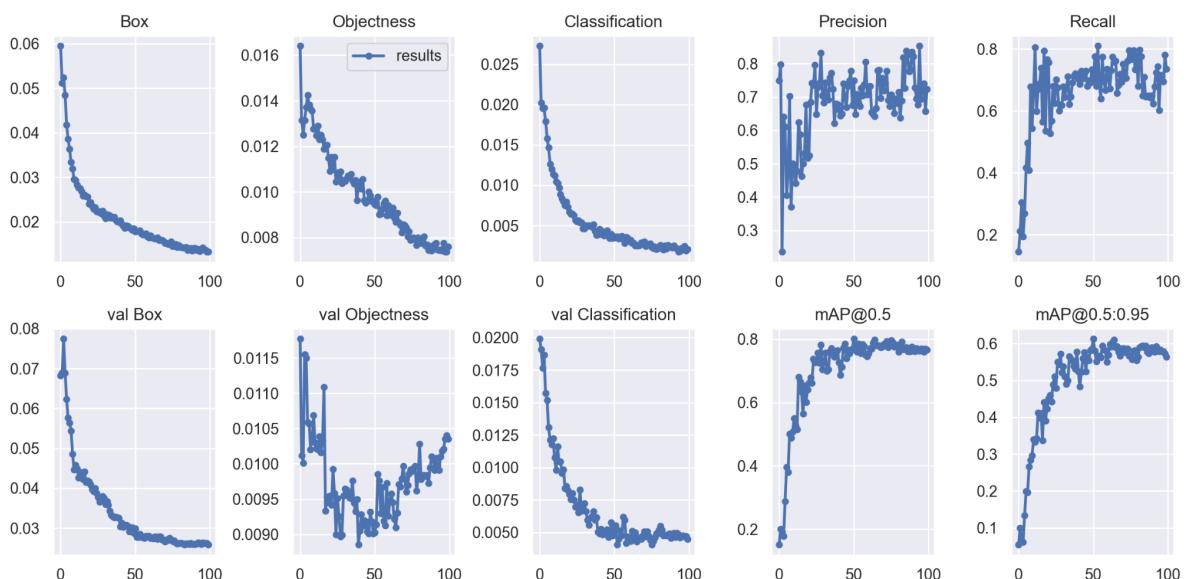
Pengujian *batch size* yang pertama, kami menggunakan nilai 2.

Tabel 5.10 nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 2

Parameter	Nilai Rata-Rata 0.704
<i>Annotation</i>	441
<i>Precision</i>	0.724
<i>Recall</i>	0.736
<i>F1-Score</i>	0.729
<i>mAP@[0.5]</i>	0.767
<i>mAP @[0.5:0.95]</i>	0.564

Epoch	gpu mem	box	obj	cls	total	labels	img_size
99/99	2.26G	0.01331	0.007609	0.002058	0.02298	3	640: 100% 441/441 [00:36<00:00, 12.2
	Class	Images	Labels	P	R		mAP@.5 mAP@.5-.95: 100% 15/15 [00:00<0
	all	60	141	0.724	0.736		0.767 0.564
	helm	60	10	0.75	0.6		0.665 0.442
	mobil	60	74	0.985	0.89		0.964 0.903
	motor	60	12	0.619	0.75		0.608 0.496
	no-helm	60	4	0.963	1		0.995 0.537
	pejalankaki	60	24	0.755	0.583		0.683 0.447
	platnomor	60	15	0.756	0.828		0.911 0.711
	zebracross	60	2	0.241	0.5		0.543 0.41
100 epochs completed in 1.064 hours.							

Gambar 5.19 nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 2



Gambar 5.20 grafik nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 2

b. **Batch-size 4**

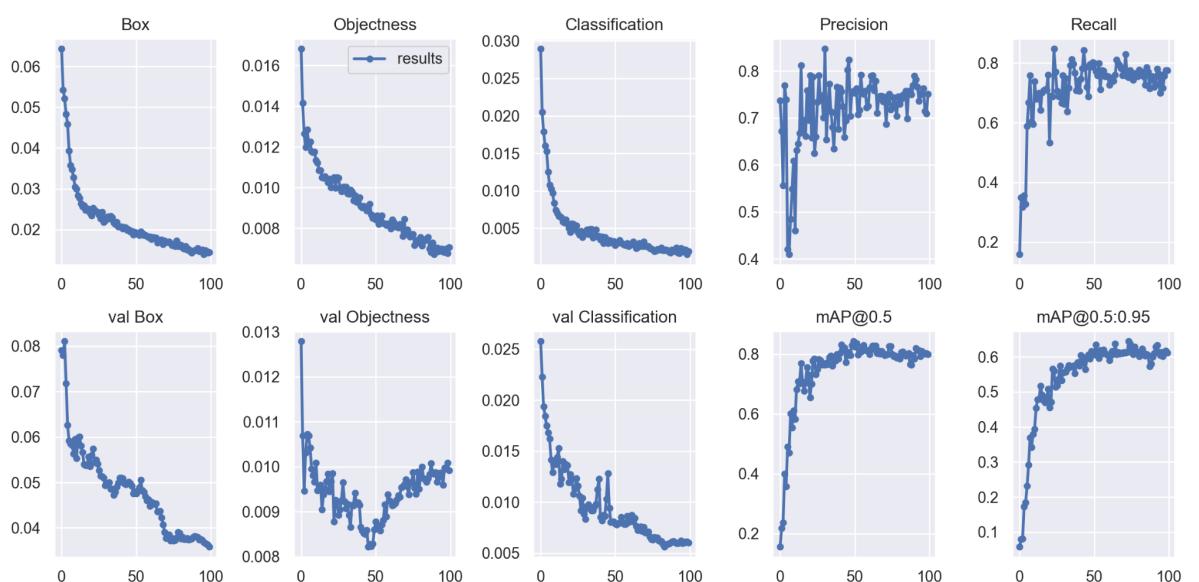
Pengujian *batch size* yang pertama, kami menggunakan nilai 4.

Tabel 5.11 nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 4

Parameter	Nilai Rata-Rata 0.739
<i>Annotation</i>	111
<i>Precision</i>	0.751
<i>Recall</i>	0.775
<i>F1-Score</i>	0.762
<i>mAP@[0.5]</i>	0.8
<i>mAP @[0.5:0.95]</i>	0.611

Epoch	gpu_mem	box	obj	cls	total	labels	img_size
99/99	4.41G	0.01439	0.007066	0.001923	0.02338	8	640: 100% 221/221 [00:31<00:00, 7.0
Class	Images	Labels		P	R	mAP@.5	mAP@.5:.95: 100% 8/8 [00:00<00:
all	60	141	0.751	0.775	0.8	0.611	
helm	60	10	0.877	0.715	0.852	0.557	
mobil	60	74	1	0.905	0.982	0.924	
motor	60	12	0.546	0.833	0.559	0.452	
no-helm	60	4	0.972	1	0.995	0.646	
pejalankaki	60	24	0.765	0.542	0.668	0.499	
platonmor	60	15	0.932	0.933	0.946	0.704	
zebracross	60	2	0.165	0.5	0.595	0.496	
100 epochs completed in 0.916 hours.							

Gambar 5.21 nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 4



Gambar 5.22 grafik nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 4

c. Batch-size 8

Pengujian *batch size* yang pertama, kami menggunakan nilai 8.

Tabel 5.12 nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 8

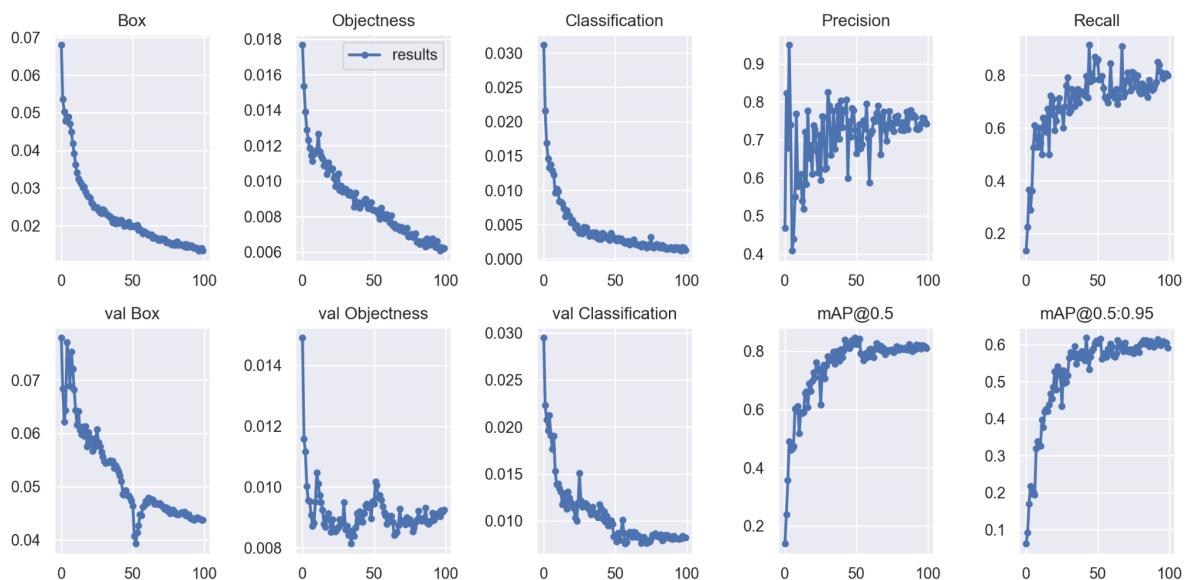
Parameter	Nilai Rata-Rata 0.742
<i>Annotation</i>	111
<i>Precision</i>	0.742
<i>Recall</i>	0.798
<i>F1-Score</i>	0.768
<i>mAP@[0.5]</i>	0.81
<i>mAP @[0.5:0.95]</i>	0.592

```

Epoch      gpu mem    box      obj      cls      total      labels      img_size
99/99      7.66G  0.01337  0.006232  0.00131   0.02091      1          640: 100% ||| 111/111 [00:30<00:00, 3
Class      Images     Labels
all        60         141       0.742      0.798      0.81       0.592
helm       60         10        0.9       0.9       0.892      0.574
mobil      60         74        0.982     0.946      0.982      0.913
motor      60         12        0.569     0.833      0.564      0.441
no-helm    60         4         1        0.892      0.995      0.564
pejalankaki 60         24        0.809     0.583      0.705      0.471
platonmor  60         15        0.661     0.933      0.934      0.693
zebracross  60         2         0.27      0.5       0.595      0.486
100 epochs completed in 0.897 hours.

```

Gambar 5.23 nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 8



Gambar 5.24 grafik nilai rata-rata pada pengujian *batch size* dengan nilai *batch size* 8

5.2.1.4. Epoch Training Data

Pada pengujian model tahap ketiga, kami kembali memasukan parameter yang sudah dihitung dan dianalisis dari proses pengujian sebelumnya, dengan mengubah nilai *learning rate* yang terbaik, yaitu:

- a. *Learning rate = 0,0001*
- b. *Batch size = 8*
- c. *Epoch = 100, 200, 300*

Berikut adalah hasil pengujian *epoch* setelah menyesuaikan parameter hasil terbaik dari rasio data dan *learning rate*.

a. 100 Epoch

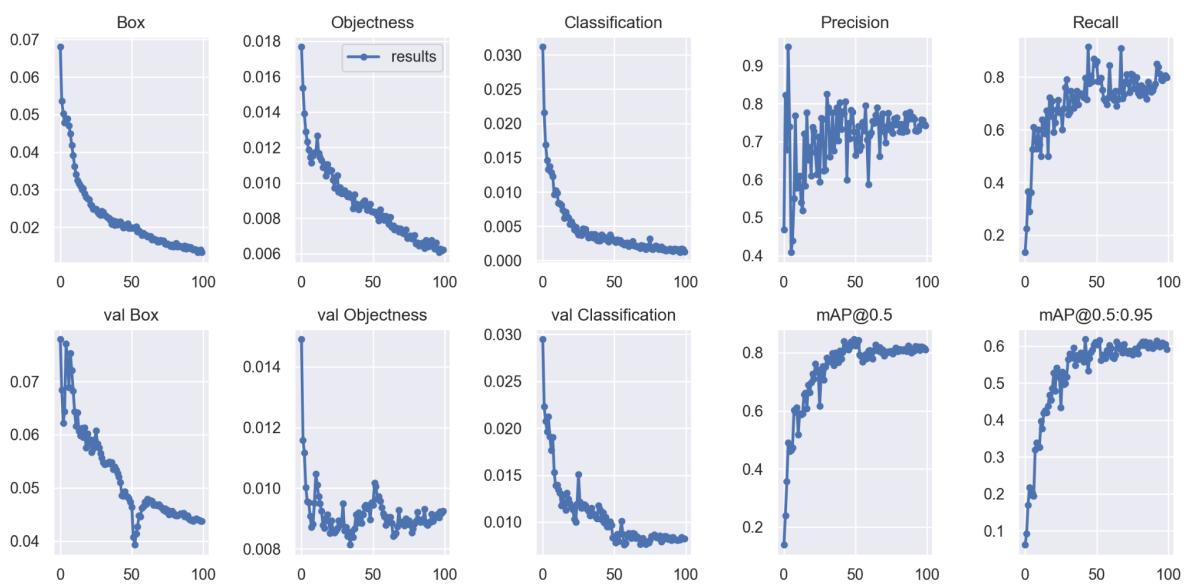
Pada pengujian pertama kami menggunakan parameter *epoch* sebesar 100.

Tabel 5.13 nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 100

Parameter	Nilai Rata-Rata 0.742
<i>Annotation</i>	111
<i>Precision</i>	0.742
<i>Recall</i>	0.798
<i>F1-Score</i>	0.768
<i>mAP@[0.5]</i>	0.81
<i>mAP @[0.5:0.95]</i>	0.592

Epoch	gpu_mem	box	obj	cls	total	labels	img_size	
99/99	7.66G	0.01337	0.006232	0.00131	0.02091	1	640: 100% 111/111 [00:30<00:00, 3	
	Class	Images	Labels	P	R		mAP@.5: .95: 100% 4/4 [00:01<0	
	all	60	141	0.742	0.798		0.81	0.592
	helm	60	10	0.9	0.9		0.892	0.574
	mobil	60	74	0.982	0.946		0.982	0.913
	motor	60	12	0.569	0.833		0.564	0.441
	no-helm	60	4	1	0.892		0.995	0.564
	pejalankaki	60	24	0.809	0.583		0.705	0.471
	platnomor	60	15	0.661	0.933		0.934	0.693
	zebracross	60	2	0.27	0.5		0.595	0.486
100 epochs completed in 0.897 hours.								

Gambar 5.25 nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 100



Gambar 5.26 grafik nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 100

b. 200 Epoch

Pada pengujian pertama kami menggunakan parameter *epoch* sebesar 200.

Tabel 5.14 nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 200

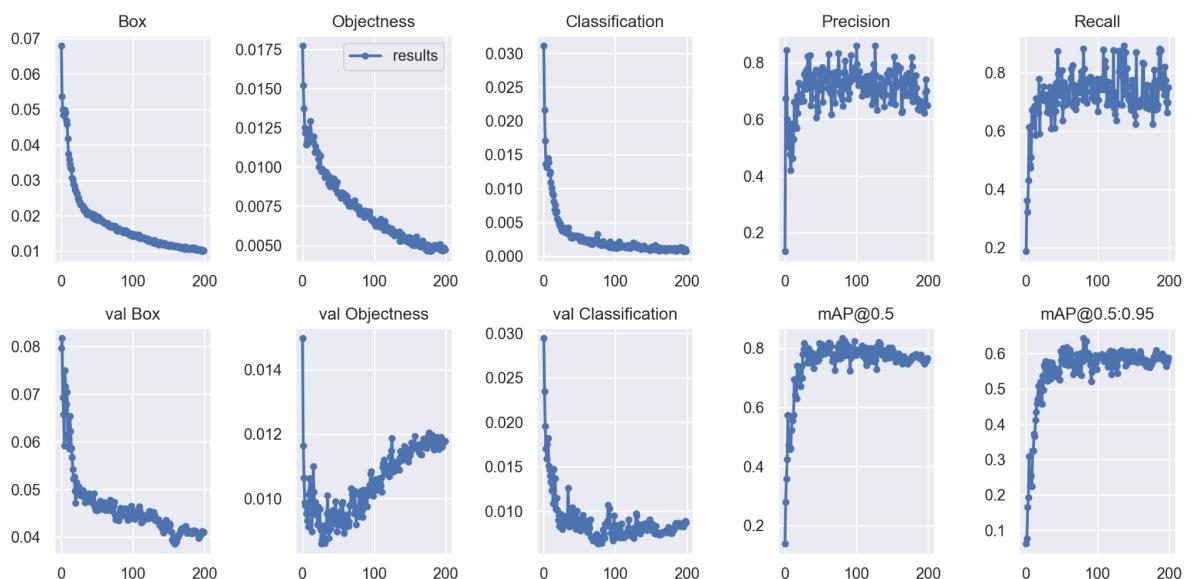
Parameter	Nilai Rata-Rata 0.689
<i>Annotation</i>	111
<i>Precision</i>	0.648
<i>Recall</i>	0.75
<i>F1-Score</i>	0.695
<i>mAP@[0.5]</i>	0.766
<i>mAP @[0.5:0.95]</i>	0.588

```

Epoch      gpu_mem    box      obj      cls      total      labels      img_size
199/199    8.16G  0.01016  0.004727  0.0007169  0.0156      1           640: 100%|██████████| 111/111 [00:30<00:00,  3.69it/s]
Class     Images   Labels      P       R      mAP@.5: 95: 100%|██████████| 4/4 [00:03<00:00,  1.32it/s]
all       60       141      0.648    0.75     0.766    0.588
helm      60        10      0.796    0.8      0.829    0.578
mobil     60        74      0.947    0.958    0.963    0.899
motor     60        12      0.464    0.723    0.546    0.453
no-helm   60         4      0.683    0.75     0.888    0.526
pejalankaki 60        24      0.736    0.583    0.597    0.467
platnomor 60        15      0.737    0.933    0.903    0.675
zebracross 60         2      0.174     0.5     0.638    0.517
200 epochs completed in 1.882 hours.

```

Gambar 5.27 nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 200



Gambar 5.28 grafik nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 200

c. 300 Epoch

Pada pengujian ketiga kami menggunakan parameter *epoch* sebesar 300.

Tabel 5.15 nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 300

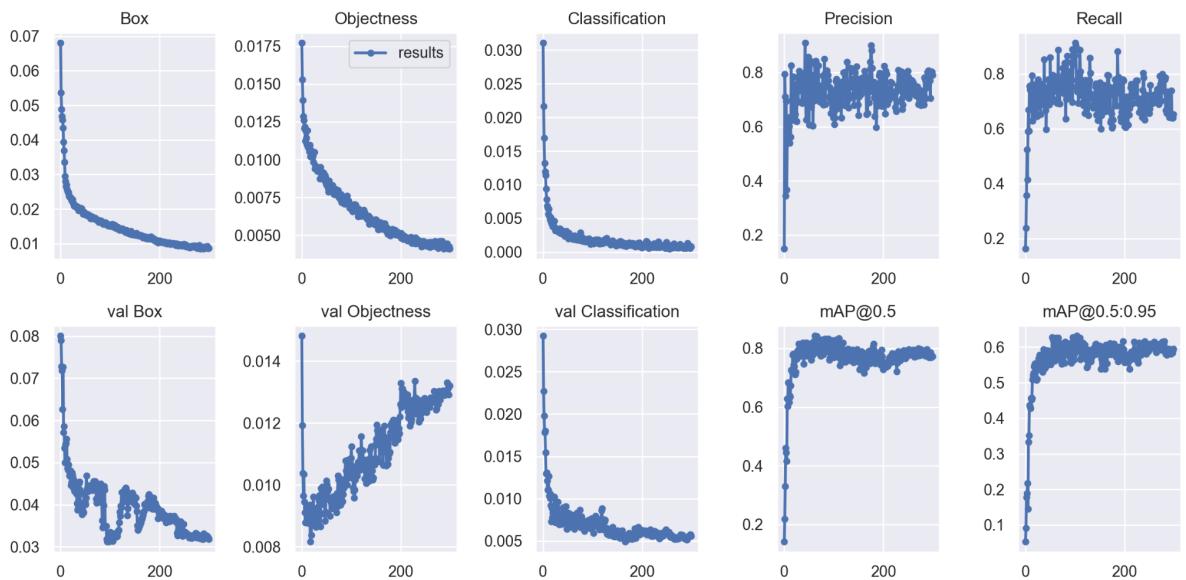
Parameter	Nilai Rata-Rata 0.705
<i>Annotation</i>	111
<i>Precision</i>	0.791
<i>Recall</i>	0.654
<i>F1-Score</i>	0.716
<i>mAP@[0.5]</i>	0.772
<i>mAP @[0.5:0.95]</i>	0.594

```

Epoch      299/299   gpu_mem    7.8G   box       0.008598   obj       0.004107   cls       0.0007601   total     0.01346   labels    5   img_size  640: 100% |██████████| 111/111 [00:29<00:00,  3.73it/s]
Class      Images    Labels
all        60        141      0.791      0.654      0.772      0.594
helm       60        10       0.999      0.7       0.841      0.556
mobil      60        74       0.974      0.824      0.95       0.896
motor      60        12       0.495      0.583      0.537      0.482
no-helm    60        4        0.98       0.75       0.871      0.512
pejalankaki 60        24       0.795      0.484      0.636      0.462
platnomor  60        15       0.981      0.733      0.971      0.707
zebracross  60        2        0.316      0.5       0.595      0.545
300 epochs completed in 2.721 hours.

```

Gambar 5.29 nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 300



Gambar 5.30 nilai rata-rata pada pengujian *epoch* dengan nilai *epoch* 300

5.2.2 Pengujian 2: Pengujian Model OCR

Pada pengujian model OCR ini kami mencoba dua jenis program yang berbeda, yang pertama menggunakan EasyOCR dan Pytesseract. Kedua pengujian dilakukan menggunakan foto yang sama dan dengan metode pengujian yang sama sehingga hasil yang didapat menjadi lebih bisa untuk dibandingkan

5.2.2.1. OCR menggunakan EasyOCR

Pada pengujian yang pertama kami menggunakan *easyocr* untuk mendeteksi digit pada plat nomor yang terdeteksi, berikut *full code* dari program tersebut:



```
● ● ●
1 import cv2
2 import easyocr
3 import matplotlib.pyplot as plt
4 import numpy as np
5 from PIL import Image
6
7
8 def extract_plate_number(image_path):
9     # Load the image
10    image = cv2.imread(image_path)
11    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
12
13    # Initialize the EasyOCR reader
14    reader = easyocr.Reader(['en'])
15
16    # Perform OCR on the image
17    results = reader.readtext(image_rgb)
18
19    # Extract the license plate number
20    plate_number = ""
21    bounding_box = None
22
23    for (bbox, text, _) in results:
24        if text.isalnum():
25            plate_number += text
26            if bounding_box is None:
27                bounding_box = np.array(bbox).astype(int)
28            else:
29                bounding_box = np.vstack((bounding_box, np.array(bbox).astype(int)))
30
31    # Crop the image based on the bounding box
32    if bounding_box is not None:
33        x, y, w, h = cv2.boundingRect(bounding_box)
34        cropped_image = image[y:y+h, x:x+w]
35    else:
36        cropped_image = image
37
38    # Segment the characters in the cropped image
39    gray = cv2.cvtColor(cropped_image, cv2.COLOR_BGR2GRAY)
40    _, binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
41    contours, _ = cv2.findContours(binary, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
42
43    # Initialize the list of segmented images
44    segmented_images = [None] * 10
45
46    # Extract the segmented images
47    for i, contour in enumerate(contours):
48        x, y, w, h = cv2.boundingRect(contour)
49        if i < len(segmented_images):
50            segmented_images[i] = cropped_image[y:y+h, x:x+w]
51
52    # Display the results
53    plt.figure(figsize=(10, 10))
54    plt.subplot(2, 6, 1)
55    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
56    plt.title('Original Image')
57    plt.axis('off')
58
59    plt.subplot(2, 6, 2)
60    plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
61    plt.title('Cropped Image')
62    plt.axis('off')
63
64    for i, segmented_image in enumerate(segmented_images):
65        plt.subplot(2, 6, i + 3)
66        if segmented_image is not None:
67            plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
68            plt.title(f'Segmented Image {i + 1}')
69        else:
70            plt.axis('off')
71
72    plt.subplot(2, 6, 12)
73    plt.text(0.5, 0.5, f'Plate Number: {plate_number}', fontsize=12, ha='center')
74    plt.axis('off')
75
76    plt.tight_layout()
77    plt.show()
78
79
80 # Example usage
81 image_path = 'test55.jpg'
82 extract_plate_number(image_path)
83
```

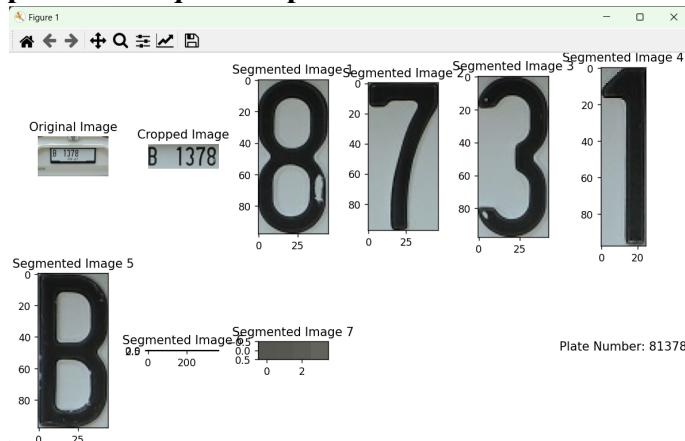
Gambar 5.31 Full code easyocr

Sample foto pertama:



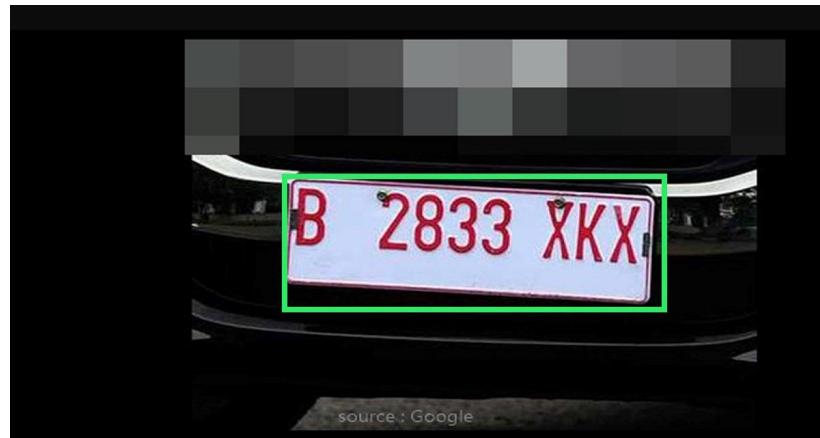
Gambar 5.32 sample foto pertama

Hasil yang didapat dari sample foto pertama:



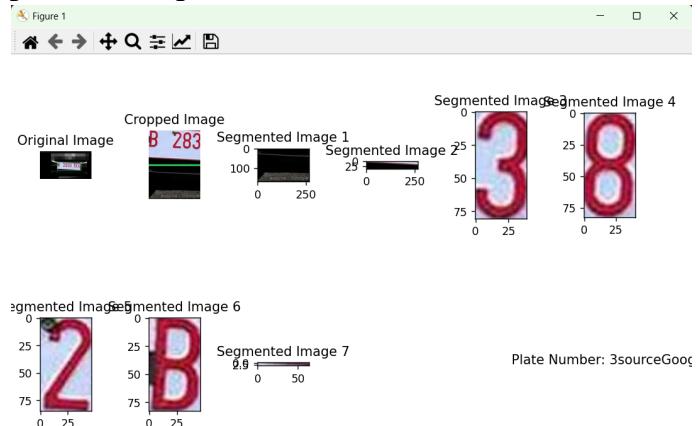
Gambar 5.33 hasil foto kedua

Sample foto kedua:



Gambar 5.34 sample foto kedua

Hasil yang didapat dari *sample* foto kedua:



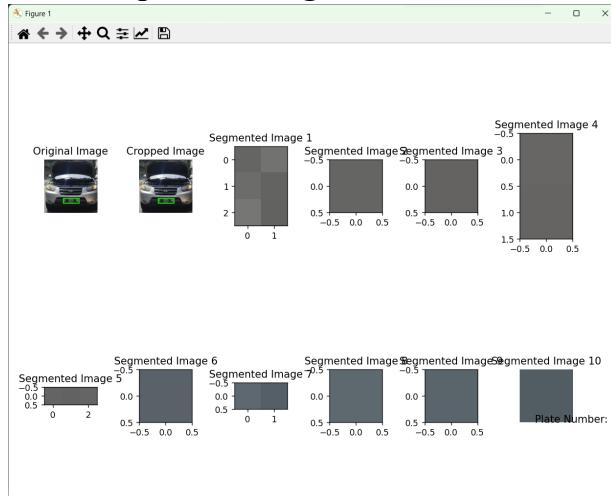
Gambar 5.35 hasil foto kedua

Sample foto ketiga:



Gambar 5.36 *sample* foto ketiga

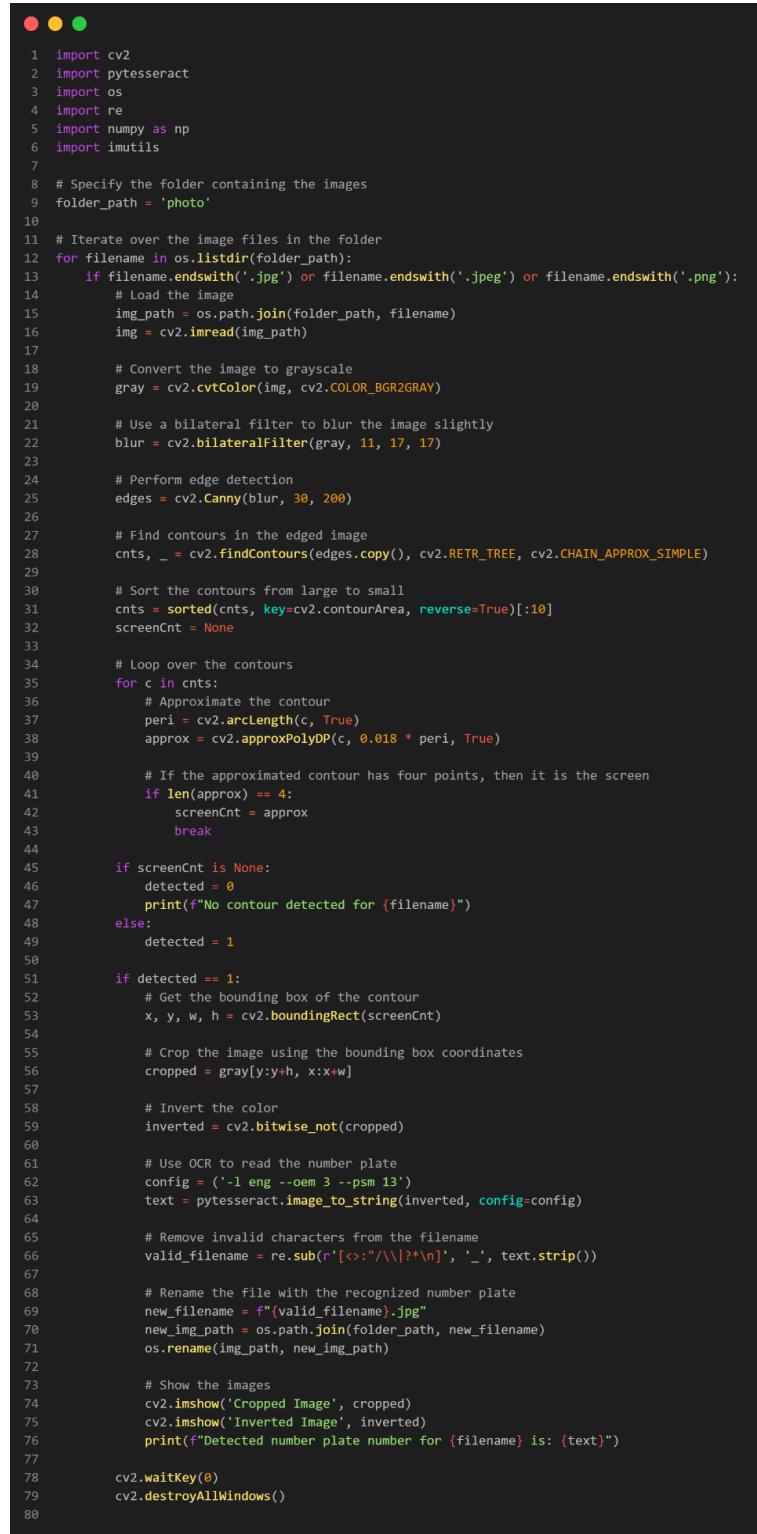
Hasil yang didapat dari *sample foto ketiga*:



Gambar 5.37 hasil foto ketiga

5.2.2.2. OCR menggunakan Pytesseract

Pada pengujian yang pertama kami menggunakan pytesseract untuk mendeteksi digit pada plat nomor yang terdeteksi, berikut *full code* dari program tersebut:



```

1 import cv2
2 import pytesseract
3 import os
4 import re
5 import numpy as np
6 import imutils
7
8 # Specify the folder containing the images
9 folder_path = 'photo'
10
11 # Iterate over the image files in the folder
12 for filename in os.listdir(folder_path):
13     if filename.endswith('.jpg') or filename.endswith('.jpeg') or filename.endswith('.png'):
14         # Load the image
15         img_path = os.path.join(folder_path, filename)
16         img = cv2.imread(img_path)
17
18         # Convert the image to grayscale
19         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
20
21         # Use a bilateral filter to blur the image slightly
22         blur = cv2.bilateralFilter(gray, 11, 17, 17)
23
24         # Perform edge detection
25         edges = cv2.Canny(blur, 30, 200)
26
27         # Find contours in the edged image
28         cnts, _ = cv2.findContours(edges.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
29
30         # Sort the contours from large to small
31         cnts = sorted(cnts, key=cv2.contourArea, reverse=True)[:10]
32         screenCnt = None
33
34         # Loop over the contours
35         for c in cnts:
36             # Approximate the contour
37             peri = cv2.arcLength(c, True)
38             approx = cv2.approxPolyDP(c, 0.018 * peri, True)
39
40             # If the approximated contour has four points, then it is the screen
41             if len(approx) == 4:
42                 screenCnt = approx
43                 break
44
45         if screenCnt is None:
46             detected = 0
47             print(f"No contour detected for {filename}")
48         else:
49             detected = 1
50
51         if detected == 1:
52             # Get the bounding box of the contour
53             x, y, w, h = cv2.boundingRect(screenCnt)
54
55             # Crop the image using the bounding box coordinates
56             cropped = gray[y:y+h, x:x+w]
57
58             # Invert the color
59             inverted = cv2.bitwise_not(cropped)
60
61             # Use OCR to read the number plate
62             config = ('-l eng --oem 3 --psm 13')
63             text = pytesseract.image_to_string(inverted, config=config)
64
65             # Remove invalid characters from the filename
66             valid_filename = re.sub(r'[\>:"/\|?*\n]', '_', text.strip())
67
68             # Rename the file with the recognized number plate
69             new_filename = f"{valid_filename}.jpg"
70             new_img_path = os.path.join(folder_path, new_filename)
71             os.rename(img_path, new_img_path)
72
73             # Show the images
74             cv2.imshow('Cropped Image', cropped)
75             cv2.imshow('Inverted Image', inverted)
76             print(f"Detected number plate number for {filename} is: {text}")
77
78             cv2.waitKey(0)
79             cv2.destroyAllWindows()
80

```

Gambar 5.38 full code pytesseract

Sample foto pertama:



Gambar 5.39 sample foto pertama

Hasil yang didapat dari sample foto kedua:



Gambar 5.40 hasil foto pertama

Sample foto kedua:



Gambar 5.41 sample foto kedua

Hasil yang didapat dari *sample foto* kedua:



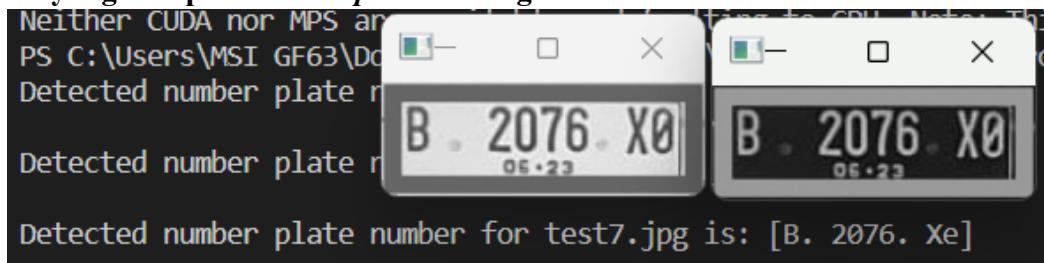
Gambar 5.42 hasil foto kedua

Sample foto ketiga:



Gambar 5.43 *sample foto* ketiga

Hasil yang didapat dari *sample foto* ketiga:



Gambar 5.44 hasil *sample foto* ketiga

5.2.3 Pengujian 3: Pengujian Sistem

Pada pengujian ini, akan dilakukan 5 skenario penggunaan drone kepada sistem yaitu Ketinggian 3m dari 3 macam *angle*, Ketinggian 5m dari tiga macam *angle*, Ketinggian 7m dari tiga macam *angle*, tiga macam kecepatan drone bergerak, dan tiga macam hasil lux meter dari waktu yang berbeda. berikut hasil dari 5 skenario tersebut :

5.2.3.1. Ketinggian 3m

Tabel 5.16 hasil pengujian pada angle 45 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	6	100%
Zebracross	6	4	66. 7%
Plat Nomor	6	1	16. 7%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{6}{6} \times 100\% = 100\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{4}{6} \times 100\% = 66. 7\%$

Nilai akurasi proses *Optical character recognition*: $\frac{1}{6} \times 100\% = 16. 7\%$

Tabel 5.17 hasil pengujian pada angle 60 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	6	100%
Zebracross	6	5	83. 4%
Plat Nomor	6	1	16. 7%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{6}{6} \times 100\% = 100\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{5}{6} \times 100\% = 83. 4\%$

Nilai akurasi proses *Optical character recognition*: $\frac{1}{6} \times 100\% = 16. 7\%$

Tabel 5.18 hasil pengujian pada angle 90 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	2	33. 4%
Zebracross	6	1	16. 7%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{2}{6} \times 100\% = 33.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{1}{6} \times 100\% = 16.7\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

5.2.3.2. Ketinggian 5m

Tabel 5.19 hasil pengujian pada angle 45 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	5	83.4%
Zebracross	6	5	83.4%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{5}{6} \times 100\% = 83.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{5}{6} \times 100\% = 83.4\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

Tabel 5.20 hasil pengujian pada angle 60 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	5	83.4%
Zebracross	6	4	66.7%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{5}{6} \times 100\% = 83.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{4}{6} \times 100\% = 66.7\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

Tabel 5.21 hasil pengujian pada angle 90 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	2	33.4%
Zebracross	6	3	50%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{2}{6} \times 100\% = 33.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{3}{6} \times 100\% = 50\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

5.2.3.3. Ketinggian 7m

Tabel 5.22 hasil pengujian pada angle 45 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	4	66.7%
Zebracross	6	3	50%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{4}{6} \times 100\% = 66.7\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{3}{6} \times 100\% = 50\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

Tabel 5.23 hasil pengujian pada angle 60 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	2	33.4%
Zebracross	6	3	50%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{2}{6} \times 100\% = 33.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{3}{6} \times 100\% = 50\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

Tabel 5.24 hasil pengujian pada angle 90 derajat

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	1	16.7%
Zebracross	6	1	16.7%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{1}{6} \times 100\% = 16.7\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{1}{6} \times 100\% = 16.7\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

5.2.3.4. Kecepatan

Tabel 5.25 hasil pengujian pada kecepatan 0,7-1,1 km/j

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	5	83.4%
Zebracross	6	2	33.4%
Plat Nomor	6	1	16.7%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{5}{6} \times 100\% = 83.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{2}{6} \times 100\% = 33.4\%$

Nilai akurasi proses *Optical character recognition*: $\frac{1}{6} \times 100\% = 16.7\%$

Tabel 5.26 hasil pengujian pada kecepatan 1,4-1,8 km/j

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	4	66.7%
Zebracross	6	1	16.7%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{4}{6} \times 100\% = 66.7\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{1}{6} \times 100\% = 16.7\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

Tabel 5.27 hasil pengujian pada kecepatan 3,2-4 km/j

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	1	16.7%
Zebracross	6	1	16.7%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{1}{6} \times 100\% = 16.7\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{1}{6} \times 100\% = 16.7\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

5.2.3.5. Lux Meter

Tabel 5.28 hasil pengujian pada intensitas cahaya

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	5	83. 4%
Zebracross	6	4	66. 7%
Plat Nomor	6	1	16. 7%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{5}{6} \times 100\% = 83.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{4}{6} \times 100\% = 66.7\%$

Nilai akurasi proses *Optical character recognition*: $\frac{1}{6} \times 100\% = 16.7\%$

Tabel 5.29 hasil pengujian pada intensitas cahaya

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	6	100%
Zebracross	6	5	83. 4%
Plat Nomor	6	1	16. 7%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{6}{6} \times 100\% = 100\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{5}{6} \times 100\% = 83.4\%$

Nilai akurasi proses *Optical character recognition*: $\frac{1}{6} \times 100\% = 16.7\%$

Tabel 5.30 hasil pengujian pada intensitas cahaya

Sistem	Total Pengujian	Pengujian Berhasil	Nilai Akurasi
Helm	6	2	33. 4%
Zebracross	6	3	50%
Plat Nomor	6	0	0%

Nilai akurasi pelanggaran tidak memakai helm : $\frac{2}{6} \times 100\% = 33.4\%$

Nilai akurasi pelanggaran berhenti di zebracross : $\frac{3}{6} \times 100\% = 50\%$

Nilai akurasi proses *Optical character recognition*: $\frac{0}{6} \times 100\% = 0\%$

5.2.4 Pengujian 4: Pengujian Notifikasi

Pengujian ini dilakukan untuk mengetahui apakah notifikasi yang berada pada *frontend* dapat mengirimkan hasil pelanggaran yang diproses oleh *backend* dari *folder* ‘proof’ dengan baik. pengujian ini dilakukan dengan memantau *folder* hasil pemrosesan menggunakan banyak *file* yang diproses didalam *folder* dan banyak *file* terkirim dengan output menampilkan ke *frontend*. Dalam upaya menguji fitur notifikasi pelanggaran pada aplikasi, kami menggunakan lima foto pelanggaran berbeda yang terhitung sebagai pelanggaran dengan format penamaan foto ‘{Jenis Pelanggaran}_[Waktu Pelanggaran]_[Plat Nomor].jpg’.

Tabel 5.31 hasil pengujian pada notifikasi

Skenario	Akurasi		
	Percobaan 1	Percobaan 2	Percobaan 3
Video 1	100%	100%	100%
Video 2	100%	100%	100%
Video 3	100%	100%	100%

Pengujian fitur notifikasi *frontend* pada aplikasi website telah dilakukan dengan metode yang sistematis dan komprehensif. Setiap aspek dari fitur tersebut telah diperiksa secara mendalam, mulai dari definisi kasus uji, persiapan data, eksekusi, hingga verifikasi. Melalui serangkaian proses ini, kami telah memastikan bahwa semua potensi skenario dan kasus uji telah ditangani dan diuji coba dengan cermat. Hasil dari pengujian tersebut menunjukkan bahwa fitur berfungsi dengan sempurna dan memenuhi semua kriteria yang telah ditetapkan, dengan tidak ada kesalahan atau ketidaksesuaian yang ditemukan. Oleh karena itu, dapat disimpulkan bahwa pengujian yang dilakukan memiliki akurasi 100%. Ini menandakan kualitas dan keandalan fitur yang tinggi, serta menjamin bahwa pengguna akan menerima notifikasi pelanggaran dengan tepat dan konsisten sesuai ekspektasi.

5.2.5 Pengujian 5: User Acceptance Test (UAT)

Berdasarkan hasil *User Acceptance Test* (UAT) yang telah dilakukan terhadap sistem deteksi pelanggaran lalu lintas dengan menggunakan drone, kami mendapatkan bahwa teknologi ini mendemonstrasikan kinerja yang memuaskan. teknologi ini dianggap memiliki potensi besar untuk mendukung penegakan hukum lalu lintas di

masa mendatang. kuesioner *User Acceptance Test* ini diberikan ke 26 responden polisi menggunakan *google form*, berikut hasil dari kuesioner tersebut :

Tabel 5.32 keterangan dan bobot nilai kuisioner

Keterangan	Bobot
SS : Sangat Setuju	5
ST : Setuju	4
N : Netral	3
TS : Tidak Setuju	2
STS : Sangat Tidak Setuju	1

Tabel 5.33 persentase hasil dari jawaban kuisioner

No.	Pertanyaan	Jawaban				
		STS	TS	N	ST	SS
1.	Apakah tampilan dari aplikasi ini menarik?	0%	0%	3.9%	0%	96.1%
2.	Apakah tampilan dari aplikasi ini mudah dipahami?	0%	0%	3.9%	0%	96.1%
3.	Apakah deteksi pelanggaran berhenti di zebracross dan tidak menggunakan helm oleh aplikasi ini membantu dalam penegakan hukum lalu lintas?	0%	0%	3.9%	0%	96.1%
4.	Apakah Anda merasa pendekslsian huruf dan angka membantu dalam proses identifikasi pelanggar lalu lintas melalui plat nomor?	0%	0%	3.9%	0%	96.1%
5.	Secara keseluruhan, apakah aplikasi AUTROL membantu Anda dalam menjalankan tugas dan tanggung jawab?	0%	0%	3.85%	3.85%	92.3%

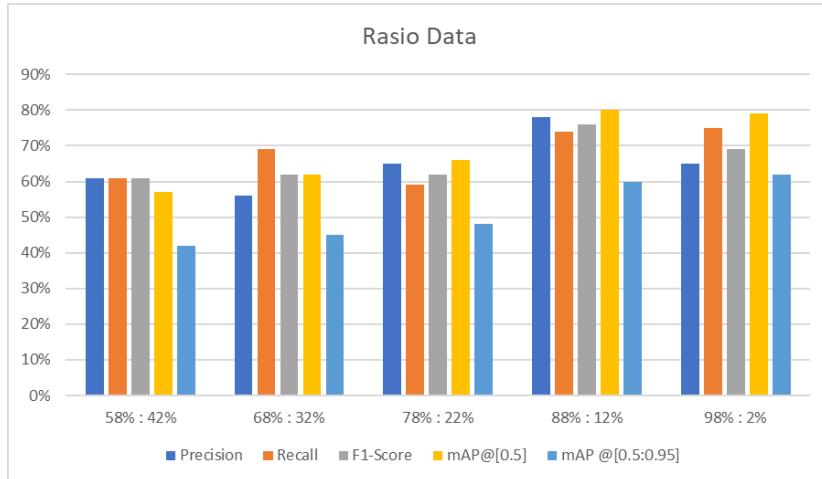
Dengan hasil data kuesioner oleh 26 responden pengguna didapatkan rata-rata yang dikumpulkan sebanyak 95.34% menjawab sangat setuju yang berada dalam nilai sangat puas dari pengguna.

5.3 Analisis Hasil Pengujian

5.3.1. Hasil analisis dari pengujian yang pertama (*Learning Rate*)

Pada pengujian pertama kami telah melakukan pengujian pada rasio data pada dataset. Dari pengujian yang dilakukan lalu kami menganalisis hasil pengujian tersebut, dan mendapatkan hasil sebagai berikut:

- Pada rasio data 58%:42% didapat nilai rata rata : 0,573
- Pada rasio data 68%:32% didapat nilai rata rata : 0,5926
- Pada rasio data 78%:22% didapat nilai rata rata : 0,6032
- Pada rasio data 88%:12% didapat nilai rata rata : 0,7394
- Pada rasio data 98%:2% didapat nilai rata rata : 0,7086



Gambar 5.30 diagram batang pada pengujian rasio data

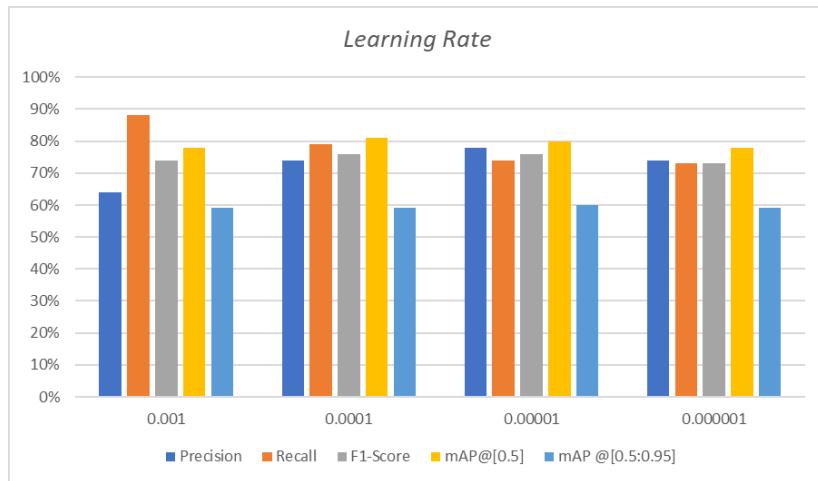
dari hasil pengujian tersebut dapat disimpulkan bahwa **nilai rata rata keakuratan yang terbaik ada pada rasio data 88%:12% dengan nilai 0,7394.**

5.3.2. Hasil analisis dari pengujian yang kedua (*Learning Rate*)

Pada pengujian kedua kami telah melakukan pengujian pada nilai *learning rate*. Dari pengujian yang dilakukan lalu kami menganalisis hasil pengujian tersebut, dan mendapatkan hasil sebagai berikut:

- Pada nilai *learning rate* 0,001 didapat nilai rata rata : 0,727
- Pada nilai *learning rate* 0,0001 didapat nilai rata rata : 0,742
- Pada nilai *learning rate* 0,00001 didapat nilai rata rata : 0,7394
- Pada nilai *learning rate* 0,000001 didapat nilai rata rata : 0,7182

dari hasil pengujian tersebut dapat disimpulkan bahwa **nilai rata rata yang terbaik ada pada nilai *learning rate* 0,0001 dengan nilai 0,742.**



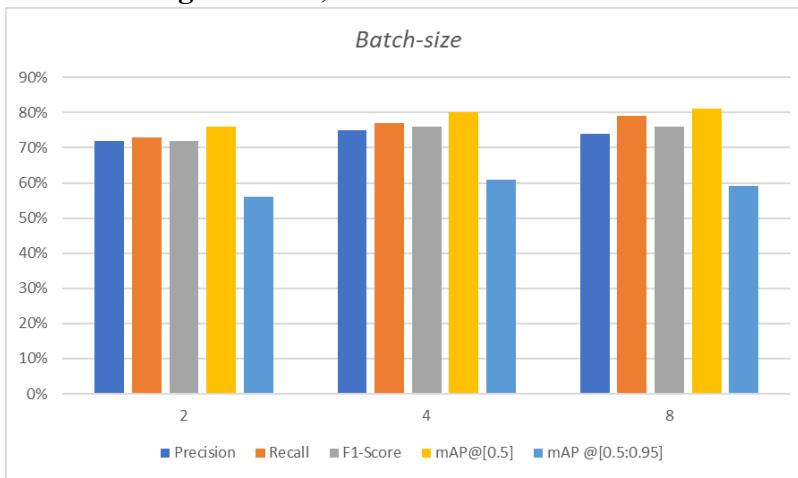
Gambar 5.45 diagram batang pada pengujian *learning rate*

5.3.3. Hasil analisis dari pengujian yang ketiga (*Batch Size*)

Pada pengujian ketiga kami telah melakukan pengujian pada nilai *batch size*. Dari pengujian yang dilakukan lalu kami menganalisis hasil pengujian tersebut, dan mendapatkan hasil sebagai berikut:

- Pada nilai *batch size* 2 didapat nilai rata rata : 0,704
- Pada nilai *batch size* 4 didapat nilai rata rata : 0,739
- Pada nilai *batch size* 8 didapat nilai rata rata : 0,742

dari hasil pengujian tersebut dapat disimpulkan bahwa **nilai rata rata yang terbaik ada pada nilai *batch size* 8 dengan nilai 0,742.**



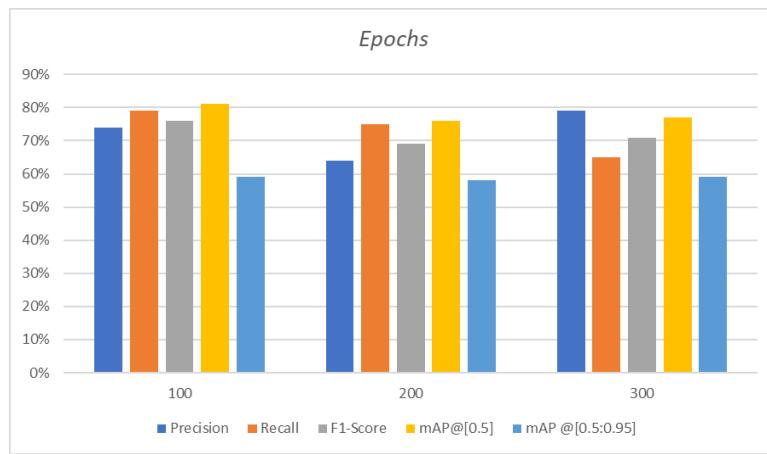
Gambar 5.46 diagram batang pada pengujian *batch size*

5.3.4. Hasil analisis dari pengujian yang keempat (*epoch*)

Pada pengujian keempat kami telah melakukan pengujian pada nilai *epoch*. Dari pengujian yang dilakukan lalu kami menganalisis hasil pengujian tersebut, dan mendapatkan hasil sebagai berikut:

- Pada nilai *epoch* 100 didapat nilai rata rata : 0,742
- Pada nilai *epoch* 100 didapat nilai rata rata : 0,689
- Pada nilai *epoch* 100 didapat nilai rata rata : 0,705

dari hasil pengujian tersebut dapat disimpulkan bahwa **nilai rata rata yang terbaik ada pada nilai *epoch* 100 dengan nilai 0,742.**



Gambar 5.47 diagram batang pada pengujian epoch

5.3.5. Hasil analisis dari pengujian yang kelima (*Optical Character Recognition*)

Pada pengujian kelima kami telah melakukan pengujian pada model dari OCR, yang pertama menggunakan *easyocr* dan yang kedua menggunakan *pytesseract*. Dari pengujian yang telah dilakukan dapat disimpulkan bahwa pembacaan digit pada plat nomor lebih akurat dan presisi menggunakan *pytesseract*.

Pengujian dilakukan dengan membandingkan kedua jenis OCR tersebut dengan menggunakan beberapa foto yang sama persis, sehingga hasil dapat lebih mudah untuk dibandingkan.

5.3.5.1. Pengujian OCR menggunakan *easyocr*

Tabel 5.34 hasil pengujian menggunakan *easyocr*

	Total Pengujian	Percobaan pertama	Percobaan kedua	Percobaan ketiga
Nilai Akurasi	3	100%	50%	0%
Nilai Error	3	0%	50%	100%

- Digit yang terbaca pada percobaan pertama : $\frac{4}{4} \times 100\% = 100\%$
- Digit yang terbaca pada percobaan kedua: $\frac{4}{8} \times 100\% = 50\%$
- Digit yang terbaca pada percobaan ketiga: $\frac{0}{7} \times 100\% = 0\%$

5.3.5.2. Pengujian OCR menggunakan *pytesseract*

Tabel 5.35 hasil pengujian menggunakan *pytesseract*

	Total Pengujian	Percobaan pertama	Percobaan kedua	Percobaan ketiga
Nilai Akurasi	3	100%	87,5%	85,7%
Nilai Error	3	0%	12,5%	14,3%

- Digit yang terbaca pada percobaan pertama : $\frac{4}{4} \times 100\% = 100\%$
- Digit yang terbaca pada percobaan kedua: $\frac{7}{8} \times 100\% = 87,5\%$
- Digit yang terbaca pada percobaan ketiga: $\frac{6}{7} \times 100\% = 85,7\%$

5.3.6. Hasil analisis dari pengujian yang keenam (Pengujian Sistem)

5.3.6.1. Ketinggian 3m

Tabel 5.36 hasil pengujian pada ketinggian 3m

Ketinggian	Angle Kamera	Nilai Akurasi		
		Helm	Zebracross	Plat nomor
3	45°	100%	66.7%	16.7%
3	60°	100%	83.4%	16.7%
3	90°	33.4%	16.7%	0%

- Nilai rata-rata akurasi helm : 77,8%
- Nilai rata-rata akurasi zebracross : 55,6%
- Nilai rata-rata akurasi plat nomor : 11,14%

Hasil dari pengujian pada ketinggian 3m adalah pada *angle* kamera 90 derajat, sistem pendeksi kesulitan mengenali objek kelas. Mengenai OCR platnomor, sistemnya kesulitan dengan kejernihan *image capturing*.

5.3.6.2. Ketinggian 5m

Tabel 5.37 hasil pengujian pada ketinggian 5m

Ketinggian	Angle Kamera	Nilai Akurasi		
		Helm	Zebracross	Plat nomor
5	45°	83.4%	83.4%	0%
5	60°	83.4%	66.7%	0%
5	90°	33.4%	50%	0%

- Nilai rata-rata akurasi helm : 66,74%
- Nilai rata-rata akurasi zebracross : 66,7%
- Nilai rata-rata akurasi plat nomor : 0%

Hasil dari pengujian pada ketinggian 5m adalah pada *angle* kamera 90 derajat, sistem pendeksi kesulitan mengenali objek kelas. Mengenai OCR platnomor, sistemnya kesulitan dengan ketinggian dan kejernihan gambar dari *image capturing*.

5.3.6.3. Ketinggian 7m

Tabel 5.38 hasil pengujian pada ketinggian 7m

Ketinggian	Angle Kamera	Nilai Akurasi		
		Helm	Zebracross	Plat nomor
7	45°	66.7%	50%	0%
7	60°	33.4%	50%	0%
7	90°	16.7%	16.7%	0%

- Nilai rata-rata akurasi helm : 38,94%
- Nilai rata-rata akurasi zebracross : 38,9%
- Nilai rata-rata akurasi plat nomor : 0%

Hasil dari pengujian pada ketinggian 7m adalah pada *angle* kamera 90 derajat, sistem pendektsian kesulitan mengenali objek kelas. Mengenai OCR platnomor, sistemnya kesulitan dengan ketinggian dan kejernihan gambar dari *image capturing*.

5.3.6.4. Kecepatan

Tabel 5.39 hasil pengujian pada kecepatan

Kecepatan Drone	Nilai Akurasi		
	Helm	Zebracross	Plat nomor
0.7-1.1	83.4%	33.4%	16.7%
1.4-1.8	66.7%	16.7%	0%
3.2-4	16.7%	16.7%	0%

- Nilai rata-rata akurasi helm : 55,6%
- Nilai rata-rata akurasi zebracross : 22,27%
- Nilai rata-rata akurasi plat nomor : 5,56%

Hasil dari pengujian kecepatan drone, menghasilkan data yang kurang akurat yang dikarenakan keterbatasan *frame per second* pada sistem *live streaming* yang hanya bisa mencapai 30-60fps.

5.3.6.5. Intensitas Cahaya

Tabel 5.40 hasil pengujian pada lux meter

Pencahayaan	Nilai Akurasi		
	Helm	Zebracross	Plat nomor
7536 lux	83.4%	66.7%	16.7%
17644 lux	100%	83.4%	16.7%
2251 lux	33.4%	50%	0%

- Nilai rata-rata akurasi helm : 72.27%
- Nilai rata-rata akurasi zebracross : 66,7%
- Nilai rata-rata akurasi plat nomor : 11,14%

Hasil dari pengujian pencahayaan ini, menghasilkan jika kondisi lux tinggi lebih bagus untuk pendekslsian melakukan klasifikasi objek. sisi negatifnya jika objek tersebut tertutup oleh bayangan yang membuat objek tersebut menjadi gelap, hal itu dapat mengakibatkan pendekslsian kesulitan dalam mendekksi objek.

5.3.6.6. Analisis Keseluruhan

1. Skenario Terbaik

mengenai data yang kami dapat, hasil dari pengujian terbaik sistem adalah menggunakan ketinggian drone mencapai 3m sampai 5m, penggunaan angle kamera 45° sampai 60°, pencahayaan lebih dari 7000 lux, dan pergerakan drone yang tidak lebih dari 1 km/j atau menggunakan posisi diam yang baik dari view drone.

Berikut hasil rata rata akurasi jika menggunakan skenario terbaik setiap sistem :

- Nilai rata-rata akurasi dari seluruh skenario helm : 90.51%
- Nilai rata-rata akurasi dari seluruh skenario helm zebracross :69.1%
- Nilai rata-rata akurasi dari seluruh skenario helm plat nomor : 11.92%

2. Evaluasi

Dalam upaya meningkatkan keamanan dan ketertiban lalu lintas, penggunaan drone telah diadopsi untuk mendekksi pelanggaran. Namun, kurangnya akurasi pada sistem deteksi zebracross dikarenakan posisi drone yang terkadang berubah. Perubahan posisi ini dapat mengakibatkan variasi pada data posisi titik pada bounding box kendaraan yang berada di dalam bounding box. Selanjutnya, berdasarkan hasil pengujian **sub sistem 5.3.5.2 dan sub**

sistem 5.3.6, terdapat beberapa poin penting yang berkaitan dengan penggunaan *Pytesseract* sebagai alat *Optical Character Recognition* (OCR) yang perlu diperhatikan :

1. Kualitas Gambar

Sub sistem 5.3.5.2 menekankan betapa pentingnya kualitas gambar bagi efektivitas OCR. Gambar yang memiliki resolusi rendah, *noise*, pencahayaan yang kurang memadai, atau detail yang hilang mempengaruhi akurasi pengenalan teks.

2. Pre-processing

Dalam pengujian, ditemukan bahwa tanpa tahapan *Pre-processing* yang tepat seperti binarisasi, penghilangan *noise*, peningkatan kontras, dan skalasi gambar, kinerja *Pytesseract* cenderung tidak optimal.

3. Orientasi Teks

Gambar yang memiliki teks dengan orientasi miring atau rotasi menyulitkan pengenalan. Hal ini telah ditegaskan oleh temuan dalam **sub sistem 5.3.5.2**.

Dari poin-poin diatas dan dengan pengujian yang telah kami lakukan, pengenalan pada digit plat nomor akan menghasilkan nilai yang terbaik dengan keadaan kamera yang memiliki sudut tegak lurus dengan posisi plat nomor, pengujian tersebut tidak sempat didokumentasikan oleh kami, maka berdasarkan hasil yang ada pada tabel pengujian menunjukkan dengan ketinggian 3 meter dan sudut 45° mampu mendapatkan hasil akurasi sebesar 16,7% dengan melakukan 6 kali pengujian. Pengenalan digit dari plat nomor pun menjadi tidak akurat ketika posisi ketinggian kamera dan kemiringan sudut dari kamera diubah, bahkan pada ketinggian 7 meter keatas pengenalan digit pada plat nomor tidak dapat berfungsi sama sekali dikarenakan jarak yang cukup jauh untuk program OCR mampu membaca digit dari plat nomor tersebut, dengan jarak yang jauh tentu resolusi gambar dari plat nomor itu menjadi tidak sebaik ketika kamera hanya berjarak 3 meter.

Maka pengenalan digit pada plat nomor yang pada umumnya berada pada posisi ketinggian yang cukup tinggi belum menjadi solusi yang optimal, berdasarkan dengan data yang didapat oleh kami pembacaan digit plat nomor akan lebih optimal apabila menggunakan posisi kamera dengan range angle 0° - 45° . Dengan kesimpulan ini kami mampu memberikan saran mengenai penelitian berikutnya, bisa dengan menggunakan jenis kamera yang memiliki resolusi sangat baik sehingga pembacaan dari digit plat nomor mampu dilakukan dengan baik oleh program OCR. Saran yang berikutnya mengenai device dan metode yang digunakan

untuk mentransfer data dari drone ke laptop atau server, produk yang kami buat masih memiliki kekurangan pada bagian ini, sehingga pembacaan OCR menjadi kurang baik. Metode ini dapat diperbaiki dan dikembangkan lebih lanjut dengan penelitian-penelitian berikutnya.

5.3.7. Uji Validitas Kuesioner *User Acceptance Test (UAT)*

Pada bagian ini, dilakukan uji validitas dari kuesioner **sub sistem 5.2.5**, uji validitas adalah indikator penting yang menunjukkan sejauh mana alat tes mampu melaksanakan fungsi ukurannya dengan presisi. Secara esensial, validitas berkaitan erat dengan kemampuan alat ukur untuk menggali secara tepat apa yang sebenarnya ingin diukur, memastikan bahwa alat tersebut bekerja sesuai dengan tujuannya. Salah satu teknik yang populer untuk menguji validitas adalah korelasi Product Moment dari Pearson. Teknik ini bukan sekadar mengukur hubungan antara dua variabel, tetapi lebih kepada validitas dan konsistensi internal dari setiap butir dalam alat ukur.

Saat hasil korelasi dari suatu item dianalisis, nilai yang diperoleh akan dibandingkan dengan standar tabel nilai korelasi Product Moment. Apabila nilai korelasi dari suatu item lebih tinggi dari yang ditentukan dalam tabel, maka item tersebut dapat dinyatakan valid. Sebaliknya, jika nilainya lebih rendah, validitas item tersebut dipertanyakan. Dengan demikian, proses ini memastikan bahwa setiap item dalam alat ukur berkontribusi secara efektif dalam mengukur konstruk yang diinginkan [46].

Rumus Korelasi *Product Moment Pearson* :

$$r = \frac{n \sum X_i Y_i - \sum X_i \sum Y_i}{\sqrt{(n \sum X_i^2 - (\sum X)^2)(n \sum Y_i^2 - (\sum Y)^2)}} \quad [46]$$

Keterangan :

r = Koefisien korelasi

$\sum X$ = Penjumlahan variabel X

$\sum Y$ = Penjumlahan variabel Y

$\sum X \sum Y$ = Perkalian penjumlahan variabel X dan Y

Tabel 5.41 tabel hasil uji validasi kuesioner

Responden	Pertanyaan					Total
	1	2	3	4	5	
1	5	5	5	5	5	25
2	5	5	5	5	5	25
3	5	5	5	5	5	25
4	5	5	5	5	5	25
5	5	5	5	5	5	25
6	5	5	5	5	5	25
7	5	5	5	5	5	25
8	5	5	5	5	5	25
9	5	5	5	5	4	24
10	3	3	3	3	3	15
11	5	5	5	5	5	25
12	5	5	5	5	5	25
13	5	5	5	5	5	25
14	5	5	5	5	5	25
15	5	5	5	5	5	25
16	5	5	5	5	5	25
17	5	5	5	5	5	25
18	5	5	5	5	5	25
19	5	5	5	5	5	25
20	5	5	5	5	5	25
21	5	5	5	5	5	25
22	5	5	5	5	5	25

Responden	Pertanyaan					Total
	1	2	3	4	5	
23	5	5	5	5	5	25
24	5	5	5	5	5	25
25	5	5	5	5	5	25
26	5	5	5	5	5	25
r Hitung	0,99500549	0,995005	0,995005	0,995005	0,931796	
r Tabel	0,8783	0,8783	0,8783	0,8783	0,8783	
V/T	V	V	V	V	V	

Berdasarkan Tabel 5.41 yang melibatkan 5 pertanyaan, semua pertanyaan terbukti valid. Ini menunjukkan bahwa setiap pertanyaan pada kuesioner tersebut relevan dan mampu mengukur apa yang seharusnya diukur. Dengan demikian, kuesioner ini dapat diandalkan untuk mengumpulkan data yang akurat dan konsisten sesuai dengan tujuan penelitiannya.

5.4 Kesimpulan dan Ringkasan CD-5

Dari semua pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem dapat berjalan cukup baik, meskipun ada beberapa fungsi yang masih membutuhkan perbaikan dan peninjauan lebih lanjut. Walaupun begitu, pengujian mampu membuktikan bahwa dengan melakukan pengujian pada lapangan secara real time bisa mendapatkan hasil yang berbeda dengan parameter yang diinginkan. Sejauh ini hasil yang didapat dari pengujian secara langsung cukup memuaskan dan dapat diambil analisisnya untuk diteliti lebih lanjut.

DAFTAR PUSTAKA

- [1] Ridwan Arifin, (Oktober 2022). "Polisi kini bisa tilang dari udara pakai drone pelanggaran apa yang diincar". <https://www.oto.detik.com/>.
- [2] Surti Risanti, (Agustus 2022). "Drone: Pengertian, jenis, dan fungsinya". <https://www.fortuneidn.com/>.
- [3] Shanthi.K.G, Sesha Vidhya S, Vishakha K, Subiksha S, Srija.K.K, Srinee Mamtha.R. Algorithms for Face Recognition Drones, June 2021 .
- [4] Sugam Dembla, Niyati Dolas, Ashish Karigar, Dr. Santosh Sonavane. Machine Learning based Object Detection and Classification using Drone. International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 06, Juni 2021.
- [5] Nourdine Aliane, Javier Fernandez, Mario Mata, Sergio Bemposta. A System for Traffic Violation Detection. sensors : www.mdpi.com/journal/sensors .
- [6] Serafina Ophelia, (November 2022). "Polisi mulai uji coba tilang elektronik pakai drone". <https://www.kompas.com/>.
- [7] Seokwon Yeom, Don-Ho Nam. Moving Vehicle Tracking with a Moving Drone Based on Track Association. Applied Sciences: April 2021.
- [8] Donho Nam, Seokwon Yeom. Moving Vehicle Detection and Drone Velocity Estimation with a Moving Drone. International Journal of Fuzzy Logic and Intelligent Systems: March 2020.
- [9] Maciej Ł. Pawelczyk, Marek Wojtyra. Real World Object Detection Dataset for Quadcopter Unmanned Aerial Vehicle Detection. Institute of Electrical and Electronics Engineers (IEEE): August, 2020.
- [10] Sining Cheng, Jiaxian Qin, Yuanyuan Chen, Mingzhu Li. "Moving Target Detection Technology Based on UAV Vision". Juli 2022,<https://www.hindawi.com/>.
- [11] Mouna Elloumi, Riadh Dhaou, Benoît Escrig, Hanen Idoudi, Leila Saidane. Monitoring road traffic with a UAV-based system. HAL Open Science : Mei 2019.
- [12] Seyedshams Feyzabadi. Joint Deep Learning for Car Detection. Electrical Engineering and Computer Science University of California, Merced: July 2017.
- [13] Sugam Dembla, Niyati Dolas, Ashish Karigar, Dr. Santosh Sonavane. Machine Learning based Object Detection and Classification using Drone. International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 06, Juni 2021.
- [14] Aswini, N., S. V. Uma, and V. Akhilesh. "Drone to Obstacle Distance Estimation Using YOLO V3 Network and Mathematical Principles." *Journal of Physics: Conference Series*. Vol. 2161. No. 1. IOP Publishing, 2022.
- [15] *Alpha Testing vs Beta Testing*. (n.d.). Diakses tanggal 23 Desember 2022, dari <https://www.practitest.com/qa-learningcenter/resources/alpha-testing-vs-beta-testing/#:~:text=Alpha%20Testing%20is%20done%20within,tested%20to%20the%20same%20depth> Čorović, Alekса, et al. "The real-time detection of traffic participants using YOLO algorithm." *2018 26th Telecommunications Forum (TELFOR)*. IEEE, 2018.
- [16] Anggadhita, Mahada Panji, and Yuni Widiastiwi. "Breaches Detection in Zebra Cross Traffic Light Using Haar Cascade Classifier." 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS). IEEE, 2020.
- [17] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). *You Only Look Once: Unified, Real-Time Object Detection*. <http://arxiv.org/abs/1506.02640>
- [18] Sarosa, M., & Muna, N. (2021). *IMPLEMENTASI ALGORITMA YOU ONLY LOOK ONCE (YOLO) UNTUK DETEksi KORBAN BENCANA ALAM*. 8(4). <https://doi.org/10.25126/jtiik.202184407>

- [19] Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. <http://arxiv.org/abs/2207.02696>
- [20] Jacob Solawetz. (2022, July 17). *What is YOLOv7? A Complete Guide.* <https://blog.roboflow.com/yolov7-breakdown/>
- [21] Salma Ghoneim. (2019, April 2). *Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?* Towards Data Science. <https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124>
- [22] Gowtham S R. (2022, June 30). *Confusion Matrix to No Confusion Matrix in Just 5mins.* Towards AI. <https://pub.towardsai.net/confusion-matrix-179b9c758b55>
- [23] Harikrishnan N B. (2019, December 11). *Confusion Matrix, Accuracy, Precision, Recall, F1 Score.* Analytics Vidhya. <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>
- [24] Ahmed Fawzy Gad. (2020). *Evaluating Object Detection Models Using Mean Average Precision (mAP).* Paperspace. <https://blog.paperspace.com/mean-average-precision/>
- [25] Chinmayee Deshpande. (2023, February 7). *The Best Guide to Know What Is React.* <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>
- [26] Tailwind CSS. (n.d.). *Rapidly build modern websites without ever leaving your HTML.* Retrieved August 6, 2023, from <https://tailwindcss.com/>
- [27] React Router. (n.d.). *Feature Overview v6.13.0 | React Router.* Retrieved August 6, 2023, from <https://reactrouter.com/en/main>
- [28] argparse. (n.d.). *Parser for command-line options, arguments and sub-commands.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/argparse.html>
- [29] io. (n.d.). *Core tools for working with streams.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/io.html>
- [30] Fredrik Lundh. (n.d.). *PIL is the Python Imaging Library.* Python. Retrieved August 6, 2023, from <https://pillow.readthedocs.io/en/stable/>
- [31] datetime. (n.d.). *Basic date and time types.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/datetime.html>
- [32] torch. (2023, May 8). *Tensors and Dynamic neural networks in Python with strong GPU acceleration.* Python. <https://pypi.org/project/torch/>
- [33] opencv-python. (2023, June 30). *Wrapper package for OpenCV python bindings.* Python. <https://pypi.org/project/opencv-python/>
- [34] numpy. (2023, July 31). *Fundamental package for array computing in Python.* Python. <https://pypi.org/project/numpy/>
- [35] flask. (n.d.). *Flask's documentation.* Python. Retrieved August 6, 2023, from <https://flask.palletsprojects.com/en/2.3.x/>
- [36] os. (n.d.). *Miscellaneous operating system interfaces.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/os.html>
- [37] re. (n.d.). *Regular expression operations.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/re.html?highlight=re#module-re>
- [38] time. (n.d.). *Time access and conversions.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/time.html?highlight=time#module-time>
- [39] json. (n.d.). *JSON encoder and decoder.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/json.html?highlight=json#module-json>
- [40] math. (n.d.). *Mathematical functions.* Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/math.html?highlight=math#module-math>

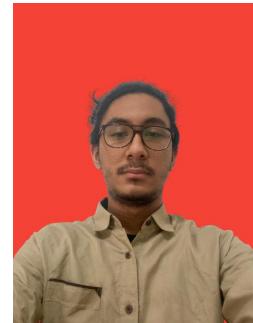
- [41] pytesseract. (n.d.). *Python-tesseract is a python wrapper for Google's Tesseract-OCR*. Python. Retrieved August 6, 2023, from <https://pypi.org/project/pytesseract/>
- [42] watchdog. (n.d.). *Filesystem events monitoring*. Python. Retrieved August 6, 2023, from <https://pypi.org/project/watchdog/>
- [43] threading. (n.d.). *Thread-based parallelism*. Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/threading.html?highlight=threading#module-threading>
- [44] queue. (n.d.). *A synchronized queue class*. Python. Retrieved August 6, 2023, from <https://docs.python.org/3/library/queue.html?highlight=queue#module-queue>
- [45] flask_cors. (n.d.). *making cross-origin AJAX possible*. Python. Retrieved August 6, 2023, from <https://flask-cors.readthedocs.io/en/latest/>
- [46] Miftahuddin, & Fithriana AR. (2008). Korelasi Antara Validitas Pada Evaluasi Yang Digunakan Dalam Menilai Hasil Belajar Siswa Dengan Hasil Kegiatan MGMP Matematika Di Kabupaten Pidie.

LAMPIRAN CD-1

(Curriculum Vitae Mhs 1)

PERSONAL INFORMATION

Full Name : Zalva Ihilani Pasha
Gender : Male
Birth Place and Date : Manado, July 10th 2001
Nationality : Indonesia
Religion : Islam
Phone Number : 081519121716
Email : zalvapasha1@gmail.com



ACADEMIC STATUS

University: Telkom University
Major : Computer Engineering
Semester : 7

EDUCATION

Institutions	City and Province	Year
SMAN 36	East Jakarta, DKI	July 2016 – June 2019
Telkom University	Bandung, West Java	August 2019 - present

SUPPORTING ACTIVITIES AND TRAININGS

Activities and Trainings	Period	Place
Sanbercode React JS - Web Frontend Development	September 2022	Bandung

ORGANIZATIONAL EXPERIENCE

Organizations	Title	Period	Descriptions
SMAN 36 Photography Club	Member	2016 - 2017	Organized Photography Club
Batavia on Telkom	Member of Publication and Documentation Division	2019-2020	Organized region community
Batavia on Telkom	Head of Publication and Documentation Division	2020 - 2022	Organized Region Community

WORKING EXPERIENCE

Work	Year	Description
Internship at DISKOMINFO Kota Bandung	2022	Doing several works of UI/UX web design

SKILLS AND HOBBIES

Language Skills : Indonesian (Native), English (Advanced)
Computer Skills : Python, Adobe Design, Java Script, Microsoft Office
Hobbies and interests : Sightseeing, Learning about gadgets, Gaming, Diving
Others : Interested in doing field project or research.

(Curriculum Vitae Mhs 2)

PERSONAL INFORMATION

Full Name : Muhammad Iqbal Dharmawan
Gender : Male
Birth Place and Date : Bekasi, 24 July 2001
Nationality : Indonesia
Religion : Islam
Phone Number : 081280992872
Email : iqbaldaarm11@gmail.com



ACADEMIC STATUS

University: Telkom University
Major : Computer Engineering
Semester : 7

EDUCATION

Institutions	City and Province	Year
SMAN 12	West Bekasi, Bekasi	July 2016 – June 2019
Universitas Telkom	Bandung, West Java	August 2019 - present

PERSONAL ACHIEVEMENTS

Awards	Year	Description
Medali Emas Pertolongan Pertama PMR se-Jawa-Bali	2018	Competitions in the field of health and rescue

SUPPORTING ACTIVITIES AND TRAININGS

Activities and Trainings	Period	Place
Webinar IoT	October 2022	Zoom Meeting
International Symposium on Electronics and Smart Devices	November 2022	Zoom Meeting

ORGANIZATIONAL EXPERIENCE

Organizations	Title	Period	Descriptions
SMAN 12 Journalist Club	Vice Chairman	2018 - 2019	Organized journalist club
SMAN 12 PMR	Head of First Aid Division	2018 - 2019	Organized training sessions and competitions
Batavia on Telkom	Member of Publication and Documentation Division	2019 - 2021	Organized region community
Overclock 2021	Member of Publication and Documentation Division	2021-2021	Organized department event

WORKING EXPERIENCE

Work	Year	Description
Internship at DISKOMINFO Kota Bandung	2022	Doing several works of UI/UX web design

SKILLS AND HOBBIES

Language Skills : Indonesian (Native), English (Advanced), Sundanese(Intermediate)

Computer Skills : Phyton, Adobe design, Microsoft Office

Hobbies and interests : Gym, Gaming, Learning about gadgets, Graphic Design

Others : Interested in doing field projects or internships at start up companies.

(Curriculum Vitae Mhs 3)

PERSONAL INFORMATION

Full Name : Hadid Candra Diputra
Gender : Male
Birth Place and Date : Bogor, 19 March 2001
Nationality : Indonesia
Religion : Islam
Phone Number : 089638930451
Email : hadidcandra343@gmail.com



ACADEMIC STATUS

University: Telkom University
Major : Computer Engineering
Semester : 7

EDUCATION

Institutions	City and Province	Year
SMAN 1 CIOMAS	West Bogor, Bogor	July 2016 – June 2019
Universitas Telkom	Bandung, West Java	August 2019 - present

PERSONAL ACHIEVEMENTS

Awards	Year	Description
2nd Place Futsal West Java	2020	Competition in Sport, West Java

SUPPORTING ACTIVITIES AND TRAININGS

Activities and Trainings	Period	Place
Webinar Teknologi	October 2022	Zoom Meeting

ORGANIZATIONAL EXPERIENCE

Organizations	Title	Period	Descriptions
SMAN 1 CIOMAS Scout	Support	2017 - 2018	Directing scout members
Forcy on telkom	Logistics	2019 - 2022	Provide goods and stage

WORKING EXPERIENCE

Work	Year	Description
Internship at DPMPTSP Bogor, West Java	2022	do some employee work such as attendance and make a simple web

SKILLS AND HOBBIES

Language Skills : Indonesian (Native), English (Advanced), Sundanese(Intermediate)
Computer Skills : Microsoft office
Hobbies and interests : Futsal,Gaming
Others : Interested in doing work in field and making a project

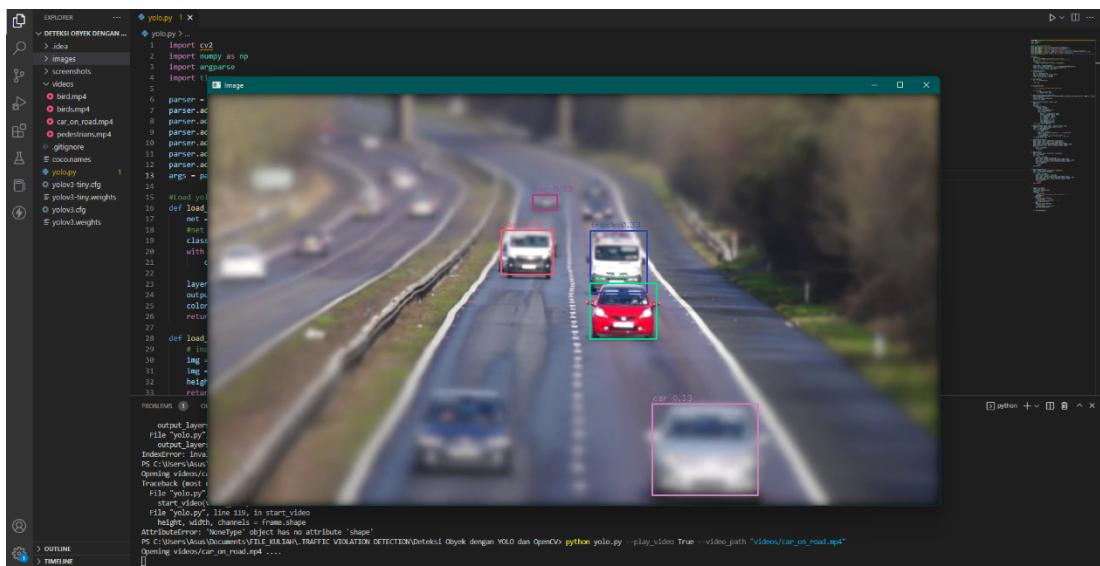
LAMPIRAN CD-2

Lampiran Spesifikasi Drone

Rencana spesifikasi drone yang akan digunakan:

- Nama Produk : DJI Mavic Pro Platinum
 - Kamera Resolusi : 12 MP, live view
 - Kapasitas Baterai : 3850 mAh
 - Sistem Kendali : Remote Control, Phone/Tablet
 - Fitur : GPS, yang dapat melacak letak drone tersebut.

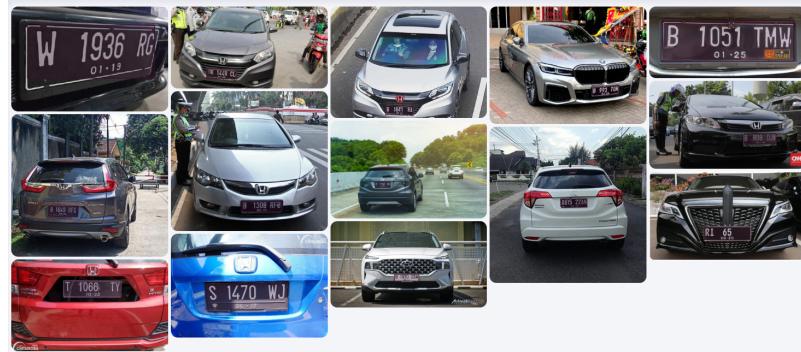
Lampiran Contoh Sistem Deteksi Kendaraan



LAMPIRAN CD-3



LAMPIRAN CD-4



Lampiran code

```
1 import cv2
2 import pytesseract
3 import os
4 import re
5 import numpy as np
6 import imutils
7
8 # Specify the folder containing the images
9 folder_path = 'photo'
10
11 # Iterate over the image files in the folder
12 for filename in os.listdir(folder_path):
13     if filename.endswith('.jpg') or filename.endswith('.jpeg') or filename.endswith('.png'):
14         # Load the image
15         img_path = os.path.join(folder_path, filename)
16         img = cv2.imread(img_path)
17
18         # Convert the image to grayscale
19         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
20
21         # Use a bilateral filter to blur the image slightly
22         blur = cv2.bilateralFilter(gray, 11, 17, 17)
23
24         # Perform edge detection
25         edges = cv2.Canny(blur, 30, 200)
26
27         # Find contours in the edged image
28         cnts, _ = cv2.findContours(edges.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
29
30         # Sort the contours from large to small
31         cnts = sorted(cnts, key=cv2.contourArea, reverse=True)[:10]
32         screenCnt = None
33
34         # Loop over the contours
35         for c in cnts:
36             # Approximate the contour
37             peri = cv2.arcLength(c, True)
38             approx = cv2.approxPolyDP(c, 0.018 * peri, True)
39
40             # If the approximated contour has four points, then it is the screen
41             if len(approx) == 4:
42                 screenCnt = approx
43                 break
44
45         if screenCnt is None:
46             detected = 0
47             print(f"No contour detected for {filename}")
48         else:
49             detected = 1
50
51         if detected == 1:
52             # Get the bounding box of the contour
53             x, y, w, h = cv2.boundingRect(screenCnt)
54
55             # Crop the image using the bounding box coordinates
56             cropped = gray[y:y+h, x:x+w]
57
58             # Invert the color
59             inverted = cv2.bitwise_not(cropped)
60
61             # Use OCR to read the number plate
62             config = ('-l eng --oem 3 --psm 13')
63             text = pytesseract.image_to_string(inverted, config=config)
64
65             # Remove invalid characters from the filename
66             valid_filename = re.sub(r'[^a-zA-Z0-9]', '_', text.strip())
67
68             # Rename the file with the recognized number plate
69             new_filename = f'{valid_filename}.jpg'
70             new_img_path = os.path.join(folder_path, new_filename)
71             os.rename(img_path, new_img_path)
72
73             # Show the images
74             cv2.imshow('Cropped Image', cropped)
75             cv2.imshow('Inverted Image', inverted)
76             print(f'Detected number plate number for {filename} is: {text}')
77
78             cv2.waitKey(0)
79             cv2.destroyAllWindows()
```

```

1 @app.route('/webcam_feed', methods=['GET', 'POST'])
2 def webcam_feed():
3     global cap
4     if request.method == 'POST':
5         req_data = json.loads(request.data)
6         video_source = req_data.get('source', '')
7         if video_source.isdigit():
8             video_source = int(video_source)
9             cap = cv2.VideoCapture(video_source)
10            cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1920)
11            cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
12        else:
13            def generate():
14                global timestamps_in_zebracross
15                while True:
16                    success, frame = cap.read()
17                    if not success:
18                        break
19                    ret, buffer = cv2.imencode('.jpg', frame)
20                    frame = buffer.tobytes()
21
22                    img = Image.open(io.BytesIO(frame))
23                    results = model(img, size=640)
24                    results.print()
25
26                    img = np.array(img)
27                    img_BGR = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
28
29                    zebrecross_boxes = []
30                    motor_boxes = []
31
32                    for pred in results.pred:
33                        boxes = pred[:, :4]
34                        confs = pred[:, 4]
35                        classes = pred[:, 5]
36
37                        for box, conf, cls in zip(boxes, confs, classes):
38                            x1, y1, x2, y2 = box
39                            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
40                            w, h = x2 - x1, y2 - y1
41
42                            class_name = classNames[int(cls)]
43
44                            if class_name == 'platnomon':
45                                color = (0, 255, 0)
46                            elif class_name == 'no-helm':
47                                color = (0, 0, 255)
48                            else:
49                                color = (0, 0, 0)
50
51                            cv2.rectangle(img_BGR, (x1, y1), (x2, y2), color, thickness=1)
52
53                            if class_name in ['motor', 'mobil']:
54                                center_x, center_y = (x1 + x2) // 2, (y1 + y2) // 2
55                                offset_y = 30
56                                center_y += offset_y
57                                cv2.circle(img_BGR, (center_x, center_y), 3, color, -1)
58
59                            conf = math.ceil((conf * 100)) / 100
60                            cls = int(cls)

```

```

61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

```

if class_name == 'zebracross':
 zebrecross_boxes.append((x1, y1, x2, y2))
if class_name == 'motor':
 motor_boxes.append((x1, y1, x2, y2))
desired_width = 600
desired_height = 900
if cls == 1 or cls == 2 :
 padded_x1 = max(0, x1 - 30)
 padded_y1 = max(0, y1 - 100)
 padded_x2 = min(img_BGR.shape[1], x2 + 30)
 padded_y2 = min(img_BGR.shape[0], y2 + 10)
 cropped_img = img_BGR[padded_y1:padded_y2, padded_x1:padded_x2]
 resized_img = cv2.resize(cropped_img, (desired_width, desired_height))
if class_name in ['no-helm']:
 center_x, center_y = (x1 + x2) // 2, (y1 + y2) // 2
 for motor_box in motor_boxes:
 x1i, y1i, x2i, y2i = motor_box
 if x1i <= center_x && x2i >= center_y && y1i <= center_y && y2i >= center_y:
 timestamp = int(time.time())
 image_filename = f"proof/{class_name}_{int(timestamp)}.jpg"
 cv2.imwrite(image_filename, resized_img)
if class_name in ['motor']:
 print(f'Processing class: {class_name}')
 center_x, center_y = (x1 + x2) // 2, (y1 + y2) // 2
 offset_y = 30
 center_y += offset_y
 for zebrecross_box in zebrecross_boxes:
 x1i, y1i, x2i, y2i = zebrecross_box
 if x1i <= center_x && x2i >= center_y && y1i <= center_y && y2i >= center_y:
 current_time = time.time()
 print(f'Object in zebrecross at time {current_time}')
 if class_name not in timestamps_in_zebracross:
 timestamps_in_zebracross[class_name] = current_time
 print(f'perhitungan (current_time) - {timestamps_in_zebracross[class_name]}')
 elif current_time - timestamps_in_zebracross[class_name] >= 5:
 print(f'Object has been in zebrecross for 3 seconds')
 image_filename = f"proof/{class_name}_{int(current_time)}.jpg"
 cv2.imwrite(image_filename, resized_img)
 del timestamps_in_zebracross[class_name]
frame = cv2.imencode('.jpg', img_BGR)[1].tobytes()
yield b"--frame\r\n"
yield b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n'
return Response(generate(), mimetype='multipart/x-mixed-replace; boundary=frame')

```
1 @app.route("/proof/<path:filename>")
2 def proof(filename):
3     return send_file(os.path.join("proof", filename))
```

```
5 @app.route("/get_images")
6 def get_images():
7     server_address = request.host_url.rstrip('/')
8     images = []
9     for filename in os.listdir(image_folder):
10         if filename.endswith(".jpg"):
11             split_filename = filename.split("_")
12             if len(split_filename) <= 2:
13                 continue
14             class_name = split_filename[0]
15             timestamp = datetime.datetime.fromtimestamp(int(split_filename[1]))
16             valid_filename = "_".join(split_filename[2:]).split(".")[0]
17             image_path = f"{server_address}/{image_folder}/{filename}"
18             images.append({
19                 "timestamp": str(timestamp),
20                 "image_path": image_path,
21                 "class_name": class_name,
22                 "valid_filename": valid_filename
23             })
24
25 return json.dumps({"images": images})
```

```
1 def start_file_watcher():
2     event_handler = ImageHandler()
3     observer = Observer()
4     observer.schedule(event_handler, path=image_folder, recursive=False)
5     observer.start()
```

```
1 class ImageHandler(FileSystemEventHandler):
2     def __init__(self):
3         self.q = queue.Queue()
4         t = Thread(target=self.process_images)
5         t.start()
6
7     def on_created(self, event):
8         if event.is_directory:
9             return
10        if event.src_path.lower().endswith(('.jpg', '.jpeg', '.png')):
11            self.q.put(event.src_path)
12
13    def process_images(self):
14        while True:
15            try:
16                img_path = self.q.get()
17                process_and_rename_number_plate(img_path)
18                time.sleep(3)
19            except queue.Empty:
20                continue
```

LAMPIRAN CD-5

Manual Book DJI Mavic Pro



User Manual V1.0

2016.10



Gambar 6.1 manual book DJI mavic pro

Aircraft

Aircraft Profile

The Mavic Pro aircraft comprise of a flight controller, video downlink, propulsion system and a Intelligent Flight Battery. This section introduces the features of the flight controller, video downlink and the rest of the component of the aircraft.

Flight Mode

The following flight modes are available for Mavic Pro:

P-mode (Positioning): P-mode works best when the GPS signal is strong. The aircraft utilizes the GPS and Forward and Downward Vision System to automatically stabilize itself, navigate between obstacles or track a moving object. Advanced features such as Tap-Fly and ActiveTrack are enabled in this mode. Note that handling gain values are reduced in P-mode.

S-mode(Sport): The aircraft is using GPS for positioning. Handling gain values of the aircraft are also adjusted in order to enhance manoeuvrability and increase the maximum flight speed. Since Forward and Downward Vision System is disabled, the aircraft will not be able to sense and avoid obstacle in S-mode. Ground Station and the Intelligent Flight functions are not available in Sport Mode.

The aircraft will switch to P-GPS state when it is using both GPS and Vision System for stabilization. When the Forward Vision System is enabled and lighting conditions are sufficient, the maximum flight altitude angle is 16° with a maximum flight speed of 22 mph (36 kph). When forward obstacle sensing are disabled, the maximum flight altitude angle is 25° and the maximum flight speed is 36 mph (58 kph).

The aircraft will switch to P-OPTI state when it is only using Vision System for stabilization.

The aircraft will automatically switch to P-ATTI mode when the GPS signal is weak and lighting conditions are too dark for the Forward and Downward Vision System. The aircraft will only use its barometer for positioning to control the altitude.



- The Forward Vision System is disabled in S-mode (Sport), which means the aircraft will not be able to automatically avoid obstacles on its route.
- The aircraft's maximum speed and braking distance are significantly increased in S-mode (Sport). A minimum braking distance of 30 meters is required in windless conditions.
- The descending speed is significantly increased in S-mode (Sport). A minimum breaking distance of 30 meters is required in windless conditions.
- The aircraft's responsiveness is significantly increased in S-mode (Sport), which means a small stick movement on the remote controller will translate into a large travel distance of the aircraft.

Be vigilant and maintain adequate maneuvering space during flight.

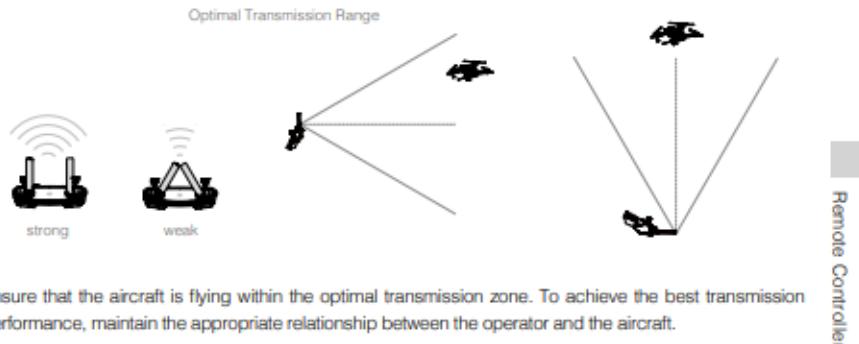


- Use the Flight Mode switch to change the flight mode of the aircraft.

Gambar 6.2 DJI mavic pro aircraft profile

Optimal Transmission Range

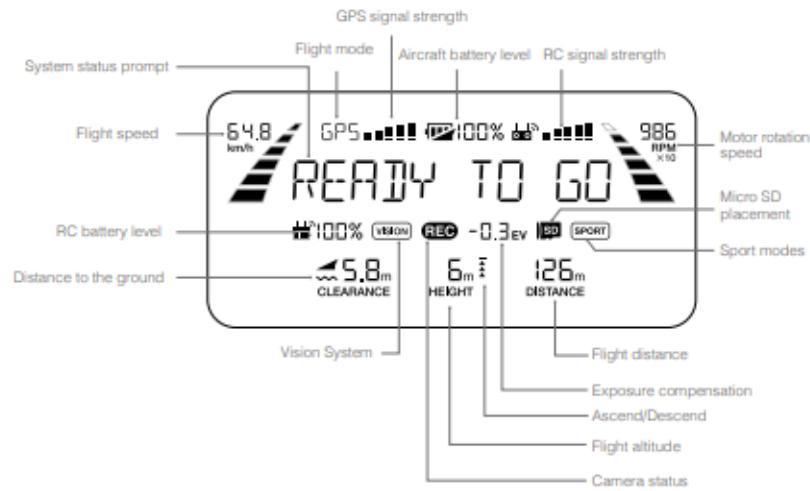
The transmission signal between the aircraft and the remote controller is most reliable within the area that is depicted in the image below:



Ensure that the aircraft is flying within the optimal transmission zone. To achieve the best transmission performance, maintain the appropriate relationship between the operator and the aircraft.

LCD Screen

LCD Screen displays various system statuses such as flight telemetries, battery level in real time. Refer to the figure below for the meaning of each icons in the LCD Screen.



Gambar 6.3 DJI mavic pro controller specification

Camera and Gimbal

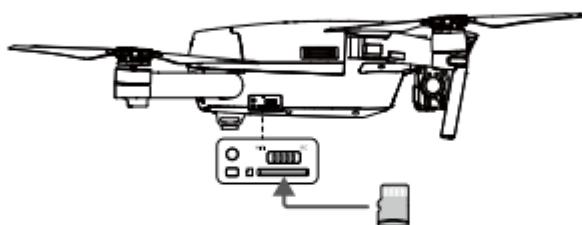
Camera Profile

The on-board camera uses the 1/2.3 inch CMOS sensor to capture video up to 4K at 30 fps with the Mavic Pro and 12 megapixel stills. You may choose to record the video in either MOV or MP4 format. Available picture shooting modes include burst, continuous, and interval mode. A live preview of what the camera sees can be monitored on the connected mobile device via the DJI GO app.

Camera Micro SD Card Slot

To store your photos and videos, insert the Micro SD card into the slot, as shown below, before turning on the Mavic Pro. The Mavic Pro comes with a 16 GB Micro SD card and supports Micro SD cards up to 64 GB. A UHS-1 Micro SD card is recommended due to their fast read and write speeds allowing you to save high-resolution video data.

Gimbal Camera



Do not remove the Micro SD card from the Mavic Pro when it is turned on.

To ensure the stability of the camera system, single video recordings are capped at 30 minutes.

Camera Data Port

Turn on the Mavic Pro and connect a USB cable to the Micro USB Port to download photos and videos to your computer.

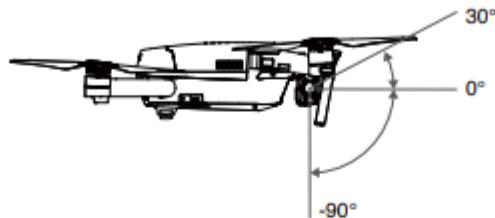


The aircraft must be turned on before attempting to access the files on the Micro SD card.

Gimbal

Gimbal Profile

The 3-axis gimbal provides a steady platform for the attached camera, allowing you to capture clear, stable images and video. The gimbal can tilt the camera within a 120° range.



Gimbal Camera

Use the gimbal dial on the remote controller to control camera tilt, or go to Camera View in the DJI GO app, tap and hold on the screen until a blue circle appears, then drag the circle to control the camera's tilt.

Gimbal Operation Modes

Two gimbal operation modes are available. Switch between the different operation modes on the camera settings page of the DJI GO app. Note that your mobile device must be connected to the remote controller for changes to take effect. Refer to the table below for details:

	Follow Mode	The angle between gimbal's orientation and aircraft's nose remains constant at all times.
	FPV Mode	The gimbal will synchronize with the movement of the aircraft to provide a first-person perspective flying experience.
		<ul style="list-style-type: none"> • Remove the Gimbal Clamp before power on the aircraft. • A gimbal motor error may occur in these situations: <ul style="list-style-type: none"> (1) the aircraft is placed on uneven ground or the gimbal's motion is obstructed. (2) the gimbal has been subjected to an excessive external force, such as a collision. Please take off from flat, open ground and protect the gimbal at all times. • Flying in heavy fog or clouds may make the gimbal wet, leading to temporary failure. The gimbal will recover full functionality after it dries. • It is normal for the gimbal to produce short pulse of beeping tone upon initialization.

Gambar 6.5 DJI mavic pro gimbal profile

Appendix

Specifications

Aircraft

Weight	1.62 lbs (734 g)
Weight (including gimbal cover)	1.64 lbs (743 g)
Dimensions	83 x 83 x 198 mm (folded)
Diagonal Length (propellers excluded)	335 mm
Max Ascent Speed	16.4 ft/s (5 m/s) in Sport Mode
Max Descent Speed	9.8 ft/s (3 m/s)
Max Speed	40.4 mph (65 kph) in Sport Mode without wind
Max Service Ceiling Above Sea Level	16404 feet (5000 m)
Max Flight Time	27 minutes (0 wind at a consistent 15.5 mph (25 kph))
Max Hovering Time	24 minutes (0 wind)
Avg. Flight Time	21 minutes (general flight, 15% battery left)
Max Flight Distance	8 mi (13 km, 0 wind)
Operating Temperature	32° to 104° F (0° to 40° C)
Satellite Positioning Systems	GPS/GLONASS

Gimbal

Controllable Range	Pitch: -90° to +30°, Roll: 0° or 90° (Horizontally and vertically)
--------------------	--

Forward Vision System

Sensing Range	Precision measurement range: 2 ft (0.7 m) to 49 ft (15 m) Detectable range: 49 ft (15 m) to 98 ft (30 m)
Operating Environment	Surfaces with clear patterns and adequate lighting (lux > 15)

Downward Vision System

Velocity Range	≤ 22.4 mph (36 kph) at 6.6 ft (2 m) above ground
Altitude Range	1 - 43 feet (0.3 - 13 m)
Operating Range	1 - 43 feet (0.3 - 13 m)
Operating Environment	Surfaces with a clear patterns and adequate lighting (lux > 15)

Camera

Sensor	1/2.3" CMOS Effective pixels: 12.35 Megapixels (Total pixels: 12.71 M)
--------	---

Lens	78.8° FOV, 28 mm (35 mm format equivalent) f/2.2 Distortion <1.5% Focus from 0.5 m to ∞
------	--

ISO Range	100 - 3200 (video), 100 - 1600 (photo)
-----------	--

Electronic Shutter Speed	8 s to 1/8000 s
--------------------------	-----------------

Max Image Size	4000x3000
----------------	-----------

Single shot

Burst shooting: 3/5/7 frames

Auto Exposure Bracketing (AEB):

3/5 bracketed frames at 0.7EV Bias

Interval

HDR

© 2016 DJI All Rights Reserved. 55

Gambar 6.6 DJI mavic pro appendix specification

Video Recording Modes	C4K: 4096x2160 24p, 4K: 3840x2160 24/25/30p 2.7K: 2720x1530 24/25/30p FHD: 1920x1080 24/25/30/48/50/60/96p HD: 1280x720 24/25/30/48/50/60/120p
Video Storage Bitrate	60 Mbps
Supported File Systems	FAT32 (\leq 32 GB), exFAT (> 32GB)
Photo	JPEG, DNG
Video	MP4, MOV (MPEG-4 AVC/H.264)
Supported SD Cards	microSD™. Max capacity: 64GB Class 10 or UHS-1 rating required.
Remote Controller	
Operating Frequency	2.400 GHz to 2.4835 GHz
Max Transmission Distance	FCC Compliant: 4.3 mi (7 km); CE Compliant: 2.5 mi (4 km) Unobstructed and free of interference.
Operating Temperature	32° to 104° F (0° to 40° C)
Battery	2970 mAh
Transmitter Power (EIRP)	FCC: \leq 26 dBm; CE: \leq 20 dBm
Operating Voltage	950 mA @ 3.7 V
Supported Mobile Device Size	Thickness supported: 6.5 - 8.5 mm, Max length: 160 mm Supported USB port types: Lightning, Micro USB (Type-B), USB Type-C™
Charger	
Voltage	13.05 V
Rated Power	50 W
Intelligent Flight Battery	
Capacity	3830 mAh
Voltage	11.4 V
Battery Type	LiPo 3S
Energy	43.6 Wh
Net Weight	Approx. 0.5 lbs (240 g)
Operating Temperature	41° to 104° F (5° to 40° C)
Max. Charging Power	100 W

Appendix

Gambar 6.7 DJI mavic pro appendix specification

Google Collab Receipt

	Colab Pro	
	\$11.09	
DATE	TIME	
Friday, July 14	10:22 PM	
Quantity	Item	Price (USD)
1	Colab Pro	\$9.99
	Subtotal	\$9.99
	VAT	\$1.10
	Total	\$11.09

STATUS
Completed

PAYMENT METHOD
Visa 6177 

TRANSACTION ID
COL:8347-4266-4968-74110

Gambar 6.8 google collab receipt

Google Drive Receipt



100 GB (Google One)

IDR 7,465.00

DATE

TIME

Saturday, July 15

2:27 AM

Quantity	Item	Price (IDR)
----------	------	-------------

1	100 GB (Google One)	IDR 6,725.0
---	------------------------	-------------

Subtotal	IDR 6,725.0
----------	-------------

VAT	IDR 740.0
-----	-----------

Total	IDR 7,465.0
-------	--------------------

STATUS

Completed

PAYMENT METHOD

Visa 6177



TRANSACTION ID

SOP.3316-8152-3367-25222

Gambar 6.9 google drive receipt