

IMPLEMENTASI *REAL TIME STREAM PROTOCOL* PADA CCTV
KAWASAN TANPA MEROKOK



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan Pendidikan Diploma
Empat (D-4) Program Studi Teknik Komputer dan Jaringan
Jurusan Teknik Elektro
Politeknik Negeri Ujung Pandang

MUHAMMAD MUBARAK A.R KUTY
425 19 042

PROGRAM STUDI D-4 TEKNIK KOMPUTER DAN JARINGAN
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI UJUNG PANDANG
MAKASSAR
2023

HALAMAN PENGESAHAN

Skripsi ini dengan judul **IMPLEMENTASI REAL TIME STREAM PROTOCOL PADA CCTV KAWASAN TANPA MEROKOK** oleh **Muhammad Mubarak A.R Kutu** NIM 425 19 042 telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang

Makassar, 20 September 2023

Mengesahkan,

Pembimbing I

Pembimbing II



Ir. Dahlia, M.T.
NIP.19641231 199103 2 003



Eddy Tungaddi, S.T., M.T.
NIP. 19790823 201012 1 001

Mengetahui,

Koordinator Program Studi
Teknik Komputer dan Jaringan



Eddy Tungaddi, S.T., M.T.
NIP. 197908232010121001

HALAMAN PENERIMAAN

Pada halaman ini, Jumat tanggal 29 September 2023 Tim Penguji Ujian Sidang Skripsi telah menerima dengan baik skripsi oleh mahasiswa Muhammad Mubarak AR. Kuty nomor induk mahasiswaa 425 19 042 dengan judul **“IMPLEMENTASI REAL TIME STREAM PROTOCOL PADA CCTV KAWASAN TANPA MEROKOK”**

Makassar, 29 September 2023

Tim Penguji Ujian Sidang Skripsi :

- | | | |
|--|--------------|---------|
| 1. Muh. Fajri Raharjo, S.T., M.T. | (Ketua) | (.....) |
| 2. Muhammad Nur Yasir Utomo, S.ST, M.Eng | (Sekertaris) | (.....) |
| 3. Fadli Tamrin, S.Kom.,M.Eng | (Anggota) | (.....) |
| 4. Zawiyah Saharuna, S.T., M.Eng | (Anggota) | (.....) |
| 5. Ir. Dahlia, M.T | (Anggota) | (.....) |
| 6. Eddy Tungadi, S.T., M.T | (Anggota) | (.....) |

KATA PENGANTAR

Alhamdulillah puji syukur atas segala Rahmat, hidayah dan karunia Allah SWT yang tak terhingga, sehingga penulis mampu menyelesaikan skripsi ini dengan baik. Sholawat serta salam kepada Rasulullah Shallallahu Alihi Wasallam yang senantiasa menjadi sumber inspirasi dan teladan terbaik bagi umat manusia.

Sebagai salah satu syarat untuk menyelesaikan studi serta memperoleh gelar Diploma IV (D-4/S1 Terapan) Pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang. Maka skripsi disusun dengan sebaik-baiknya. Penulis menyadari bahwa skripsi ini tidak mungkin terselesaikan tanpa adanya, dukungan, bantuan, bimbingan dan doa dari berbagai pihak selama proses penyusunan skripsi ini. Pada kesempatan ini penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Orang tua penulis yakni Bapak ALM. Abdul Rahim Kutu dan Ibu Patmawati yang selalu memberikan dukungan, kasih sayang, semangat dan doa terbaik yang tidak pernah putus sehingga penulis mampu menyelesaikan skripsi ini dengan sangat baik.
2. Bapak Prof. Ir. Ilyas Mansur, M.T selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Bapak Eddy Tungadi, S.T., M.T. selaku Koordinator Program Studi Teknik Komputer dan Jaringan.
5. Ibu Ir. Dahlia, M.T. selaku pembimbing I dan Eddy Tungadi, S.T., M.T. selaku pembimbing II atas segala ilmu, motivasi, nasihat, arahan, bimbingan, bantuan dan kesediaan waktu serta kesabarannya dalam membimbing penulis hingga dapat menyelesaikan penelitian ini.
6. Seluruh Dosen dan Staf Jurusan Teknik Elektro, khususnya Program Studi D4 Teknik Komputer dan Jaringan.

DAFTAR ISI

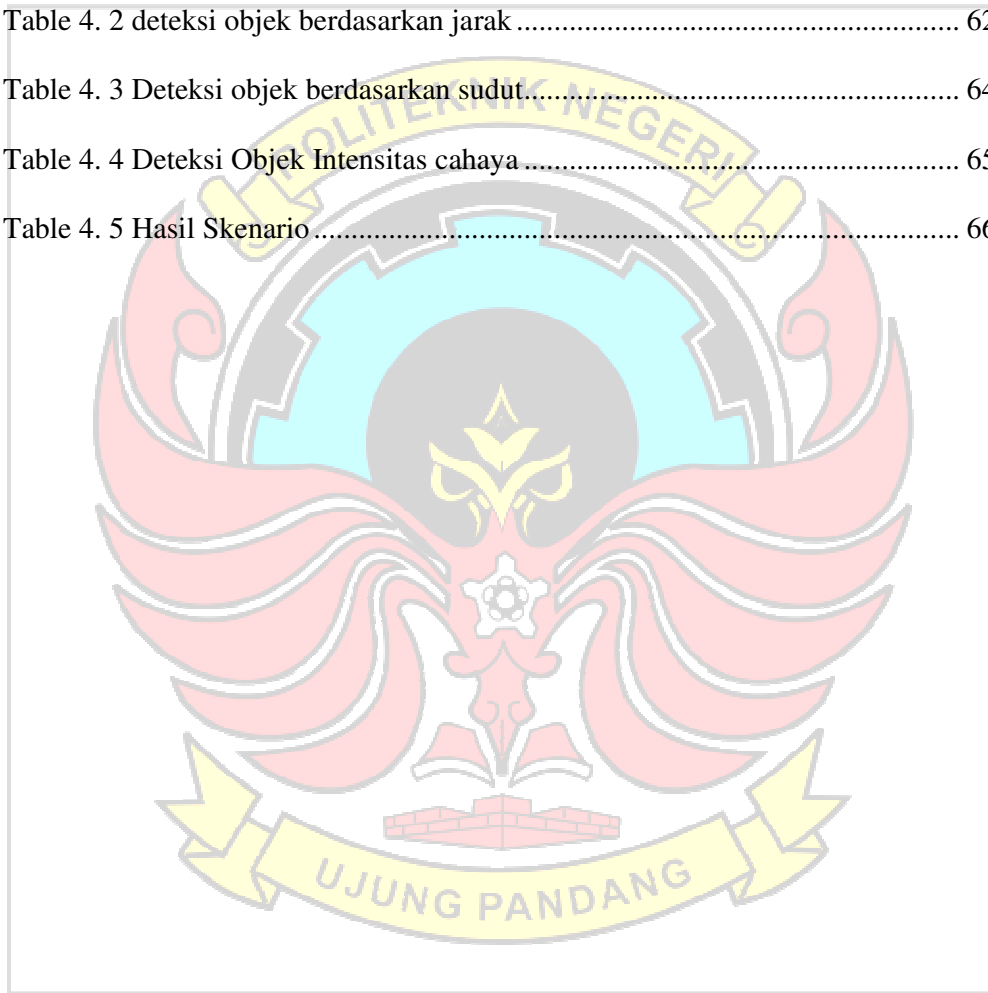
HALAMAN PENGESAHAN.....	ii
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	viii
BAB I PENDAHULUAN.....	2
1.1 Latar Belakang	2
1.2 Rumusan Masalah	4
1.3 Ruang Lingkup Penelitian.....	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 <i>You Only Look Once (YOLO)</i>	6
2.2 <i>You Only Look Once v5 (YOLOv5)</i>	7
2.3 OpenCV	9
2.4 <i>Python</i>	11
2.5 <i>Framework Flask Python</i>	12
2.6 <i>Cuda</i>	13
2.7 Blackbox Testing	14
2.8 Google Colab	14
BAB III METODOLOGI PENELITIAN	16
3.1 Tempat dan Waktu	16
3.2 Kebutuhan Sistem	16
3.2.1 Perangkat Keras (<i>Hardware</i>)	16
3.2.2 Perangkat Lunak (<i>Software</i>).....	17
3.3 Metode Penelitian	18
3.3.1 Studi Literatur	19
3.3.2 Pengumpulan Data	19
3.3.3 Perancangan Sistem	20
3.3.4 Implementasi.....	31
3.3.5 Pengujian.....	32
BAB IV HASIL DAN PEMBAHASAN	34
4.1 Hasil	34
4.2 Hasil Pembuatan Sistem.....	34

4.2.1	Pembuatan model.....	34
4.3	Pengujian Fungsional.....	60
4.4	Pengujian Skenario	61
4.4.1	Pengujian Dimensi	61
4.5	Hasil Pengujian	72
BAB V PENUTUP.....		79
5.1	Kesimpulan	79
5.2	Saran	80
DAFTAR PUSTAKA.....		81



DAFTAR TABEL

Tabel 3. 1 Perangkat Keras (<i>Hardware</i>)	16
Tabel 3. 2 Perangkat Lunak (<i>Software</i>)	17
Table 4. 1 Hasil <i>Blackbox Testing</i>	60
Table 4. 2 deteksi objek berdasarkan jarak	62
Table 4. 3 Deteksi objek berdasarkan sudut.....	64
Table 4. 4 Deteksi Objek Intensitas cahaya	65
Table 4. 5 Hasil Skenario.....	66



DAFTAR GAMBAR

Gambar 2.1 Sistem Deteksi Yolo.....	6
Gambar 2.2 Ilustrasi <i>YOLOv5</i>	8
Gambar 2.3 Arsitektur <i>YOLOv5</i>	9
Gambar 2.4 Roboflow.....	15
Gambar 3.1 Tahapan Metode Penelitian.....	18
Gambar 3.2 Desain Sistem.....	21
Gambar 3.3 Block Diagram Custom Mode	27
Gambar 3.4 Custome Model	29
Gambar 3.5 Denah Ruangan	30
Gambar 3.6 <i>Activity Diagram</i>	31
Gambar 4.1 Pembuatan workspace.....	35
Gambar 4.2 Upload gambar dataset.....	36
Gambar 4.3 gambar dataset untuk annotating.....	36
Gambar 4.4 gambar rokok di Annotating	37
Gambar 4.5 dataset gambar yang berhasil di Annotating	38
Gambar 4.6 menambahkan gambar untuk dijadikan model gambar custome	38
Gambar 4.7 Tahapan Augmentasi untuk model gambar.....	39
Gambar 4.8 model custmoe untuk dataset gambar	40
Gambar 4.9 Generate hasil model custome gambar.....	41
Gambar 4.10 Hasil Generate model gambar custome.....	42
Gambar 4.11 Hasil Pembuatan model gambar custome	43
Gambar 4.12 Versi Export gambar model dataset	44

Gambar 4. 13 Mengaktifkan GPU pada laptop untuk digunakan Google Colab..	44
Gambar 4.14 Tampilan note baru untuk memasukkan perintah	44
Gambar 4.15 Berhasil melakukan clone dan Install Yolov5.....	45
Gambar 4.16 berhasil mengimpor torch.....	46
Gambar 4.17 mendownload model gambar dataset dari roboflow	47
Gambar 4.18 Mengubah classes menjadi rokok	48
Gambar 4.19 konfigurasi train dan val dengan menggunakan folder roboflow ...	48
Gambar 4.20 melatih model gambar untuk versi Yolov5.....	49
Gambar 4.21 Last.pt hasil latih untuk Yolov5.....	51
Gambar 4.22 informasi sistem CCTV.....	52
Gambar 4.23 sistem CCTV memanggil ke server	52
Gambar 4.24 Berhasil menampilkan CCTV ke Flask.....	53
Gambar 4.25 sistem Yolov5 berhasil mendeteksi objek Rokok	53
Gambar 4.26 tampilan terminal apabila telah berhasil.....	54
Gambar 4.27 Tampilan Flask.....	55
Gambar 4.28 URL Flask yang berhasil menampilkan visual CCTV.....	55
Gambar 4.29 Hasil kegiatan merokok yang telah berhasil di simpan pada database	56
Gambar 4.30 sistem berhasil mengirim bukti kegiatan merokok ke databse.....	56
Gambar 4.31 botfathee	57
Gambar 4.32 token HTTP API BotFather	58
Gambar 4.33 Notifikasi kegiatan merokok ke telegram	59
Gambar 4.34 Tampilan halaman web	73

Gambar 4.35 Tampilan halaman web	73
Gambar 4.36 Tampilan halaman web	74
Gambar 4.37 Notifikasi bot telegram.....	75
Gambar 4.38 Notifikasi bot telegram.....	76
Gambar 4.39 Notifikasi bot telegram.....	77
Gambar 4.40 Database capture	78



Implementasi *Real Time Stream Protocol* Pada Cctv Kawasan Tanpa Merokok

RINGKASAN

Banyaknya pegawai kantor BPKAD kota palopo melakukan kegiatan merokok di dalam ruangan sehingga berdampak buruk pada kesehatan dan produktivitas pegawai. Penelitian ini bertujuan untuk membuat sistem pendeteksi pegawai yang melakukan kegiatan merokok di dalam kawasan tanpa rokok. Sistem akan mendeteksi kegiatan merokok di dalam ruangan kerja yang terdapat CCTV. Dengan adanya model atau algoritma *You Only Look Once* (YOLO) dalam bidang deteksi objek dalam pengolahan citra. Model YOLO dirancang untuk mendeteksi objek dalam sebuah gambar atau frame video dengan efisien dan cepat serta menggunakan RTSP yang ada di CCTV dapat melakukan *realtime* untuk melakukan pendeteksian kegiatan merokok di dalam ruangan. Untuk dapat mengirimkan bukti kegiatan merokok secara *realtime* digunakan telegram bot untuk sebagai pengiriman bukti kegiatan merokok tersebut dengan menggunakan library *teleport*. Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, penelitian ini berhasil mengidentifikasi objek rokok dan gerakan merokok pada kantor BPKAD kota palopo di ruangan bidang aset. Dalam sistem tersebut dilakukan pengujian jarak dari 1 meter sampai 3 meter serta sudut 0 derajat sampai 45 derajat dan intensitas cahaya pada ruangan yaitu 37 LUX telah berhasil mendeteksi benda rokok dan gerakan merokok pada ruangan bidang aset. Hasil akurasi sistem dapat mendeteksi kegiatan merokok sebesar 66,6%.

Kata Kunci : Yolov5, kegiatan merokok pegawai, CCTV RTSP, jarak, sudut, intensitas cahaya

BAB I PENDAHULUAN

1.1 Latar Belakang

Closed Circuit Television atau disingkat CCTV adalah sebuah perangkat kamera video digital yang digunakan untuk mengirim sinyal ke layar monitor di tempat tertentu dengan tujuan untuk memantau situasi dan kondisi suatu tempat, sehingga dapat mencegah terjadinya pelanggaran (Mayasari 2020). CCTV dapat digunakan untuk menampung rekaman dari setiap informasi yang dilihat dan didengar. CCTV adalah salah satu alat pengawas yang umum digunakan, dengan penempatan posisi yang sesuai dengan keinginan, agar dapat memantau keadaan aktivitas pegawai kantor (Arsyad et al. 2023).

Salah satu masalah yang terjadi di kantor Badan Pengelolaan Keuangan dan Aset Daerah Kota Palopo adalah pegawai yang merokok di area kawasan tanpa rokok. Apabila masuk jam istirahat atau jam makan siang maka beberapa pegawai laki-laki melakukan kegiatan merokok di dalam ruangan atau kawasan tanpa rokok yang menyebabkan asap rokok menyebar di sekeliling area tersebut. Tidak adanya bukti bahwa siapa yang melakukan kegiatan merokok di area kawasan tanpa rokok, maka dibutuhkan CCTV.

CCTV dapat memantau segala aktivitas kegiatan di area kawasan tanpa rokok, tetapi untuk melakukan pengecekan aktivitas yang terjadi, maka harus masuk ke dalam ruangan CCTV. Oleh karena itu, dibutuhkan *Real Time Stream Protocol* (RTSP) yang dapat digunakan untuk mempermudah dalam melakukan pengecekan CCTV tanpa harus masuk ke dalam ruangan pemantauan CCTV. RTSP merupakan protokol kontrol untuk pengiriman data yang memiliki

karakteristik real-time seperti *streaming* media. RTSP sering juga disebut sebagai *remote control* jaringan untuk *streaming server*. RTSP mempunyai kemampuan untuk mempertahankan keadaan dari sebuah sesi (mempertahankan hubungan antara video streaming server dan client secara konsisten). Protokol tersebut diharapkan bisa memberikan cara pemilihan jalur pengiriman yang optimum kepada klien sebagai dasar protokol pengiriman data (Rizki, Munadi, and Syahrial 2019). Walaupun menggunakan CCTV dan RTSP, tetapi masih belum bisa mendeteksi pegawai yang merokok, sehingga dibutuhkan sebuah algoritma yang dapat mendeteksi pegawai merokok pada kawasan dilarang merokok di Kantor Badan Pengelolaan Keuangan dan Aset Daerah Kota Palopo dengan menggunakan algoritma *You Only Look Once* (YOLO).

You Only Look Once (YOLO) merupakan salah satu metode *deep learning* yang populer digunakan dalam mendeteksi objek. Metode ini menggunakan jaringan saraf konvolusi yang dijadikan sebagai pendekatan *deep learning* untuk mendeteksi wajah pada suatu video (Shamrat et al. 2021).

Pada penelitian ini-penulis merekomendasikan sistem kamera CCTV menggunakan RTSP dan menggunakan algoritma YOLOv5 dalam memantau kedisiplinan pegawai untuk tidak merokok di area kawasan tanpa rokok. Dengan menggunakan RTSP maka dapat memudahkan kepala badan untuk memantau kegiatan merokok pegawai dalam hal merokok pada kawasan tanpa rokok. RTSP digunakan pada penelitian ini sebab RTSP dapat menampilkan *stream* CCTV secara *real time* serta terdapat bukti *screenshot* pegawai yang melakukan kegiatan merokok. Dengan ditambahkannya algoritma YOLOv5 pada CCTV yang

dilengkapi dengan RTSP, maka algoritma ini digunakan untuk mendeteksi pegawai yang merokok di area kawasan tanpa rokok. YOLOv5 digunakan pada penelitian ini karena memiliki nilai akurasi pendeteksi objek yang sangat tinggi serta memiliki beberapa fitur tambahan untuk mendukung ekstensibilitasnya di berbagai perangkat keras *platform* dari *Central Processor Unit* CPU hingga *Graphics Processing Unit* GPU.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, maka fokus permasalahan adalah sebagai berikut :

- 1) Bagaimana CCTV mengawasi pegawai Kantor Badan Pengelolaan Keuangan dan Aset Daerah Kota Palopo yang merokok pada kawasan tanpa rokok?
- 2) Bagaimana membuat sistem kamera CCTV yang dapat mendeteksi pegawai merokok di area kawasan tanpa rokok?

1.3 Ruang Lingkup Penelitian

Adapun ruang lingkup pada penelitian ini adalah sebagai berikut ;

- 1) Fokus tugas akhir mengenai pembuatan sistem mengidentifikasi aktivitas pegawai merokok pada kawasan tanpa rokok di lingkup area Badan Pengelolaan Keuangan dan Aset Daerah Kota Palopo
- 2) Sistem kamera CCTV dengan Ketinggian 3 meter dan jarak pegawai minimal 1 meter dan maksimal 4 meter dari CCTV dapat mendeteksi pegawai yang melakukan kegiatan merokok dalam kawasan tanpa rokok.
- 3) Sistem kamera CCTV dapat mendeteksi kegiatan merokok di dalam kawasan tanpa rokok

- 4) Sistem kamera CCTV kamera dapat menyimpan dan mengirim bukti kegiatan merokok di kawasan tanpa rokok.
- 5) Data yang digunakan pada penelitian ini adalah data dari kaggle, Unsplash dan menggunakan kamera smartphone untuk mengambil foto rokok.

1.4 Tujuan Penelitian

Adapun tujuan pada penelitian ini adalah sebagai berikut :

- 1) Mengukur kinerja algoritma YOLOv5 yang dirancang dengan basis RTSP dalam mengidentifikasi pegawai merokok di area kawasan tanpa rokok.
- 2) Mengidentifikasi pegawai melakukan kegiatan merokok dengan menggunakan YOLOv5.

1.5 Manfaat Penelitian

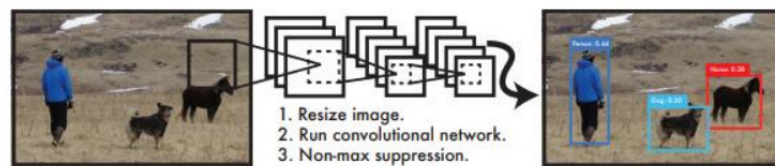
Adapun manfaat pada penelitian ini adalah sebagai berikut :

- 1) Pegawai Kantor Badan Pengelolaan Keuangan dan Aset Daerah Kota Palopo tidak merokok di area kawasan tanpa rokok.
- 2) Memberikan informasi/data pegawai yang melanggar aturan merokok tidak pada tempatnya.

BAB II TINJAUAN PUSTAKA

2.1 *You Only Look Once* (YOLO)

Metode YOLO pertama kali diperkenalkan oleh Joseph Redmon dan kawan-kawan pada tahun 2016 pada paper yang berjudul *You Only Look Once: Unified, Real-Time Object Detection*. *You Only Look Once* (YOLO) menjadi salah satu metode yang cukup populer dalam mempelajari dan mengenali objek (Muhaimin and Tjong 2021). YOLO merupakan salah satu pendekatan deep learning untuk mendeteksi objek secara *realtime* dengan basis *Convolutional Neural Network* (CNN). CNN sendiri merupakan pengembangan dari Multilayer Perceptron (MLP) yang mampu mengolah data dua dimensi berupa citra (Fu'adah et al. 2022). Dalam proses mendeteksi objek pada sebuah citra, YOLO menggunakan pendekatan jaringan konvolusi tunggal pada keseluruhan citra untuk memprediksi bounding box dan label kelas pada setiap bounding box (Kusuma, Usman, and Saidah 2021).



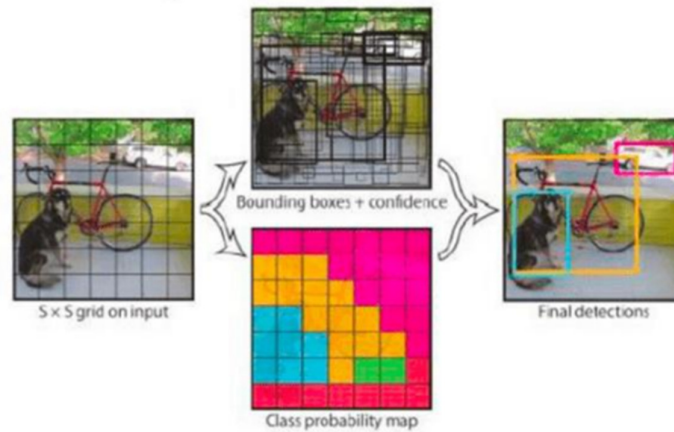
Gambar 2.1 Sistem Deteksi Yolo

Pada Gambar 2.1 proses yang terjadi adalah citra akan di proses dengan mengubah ukuran menjadi 448 x 448, selanjutnya proses yang terjadi adalah *non-max suppression* yaitu untuk menghasilkan *bounding-box* dan menentukan label dari setiap objek yang terdeteksi (Anggraini, Kusrini, and Al Fatta 2022).

2.2 *You Only Look Once v5 (YOLOv5)*

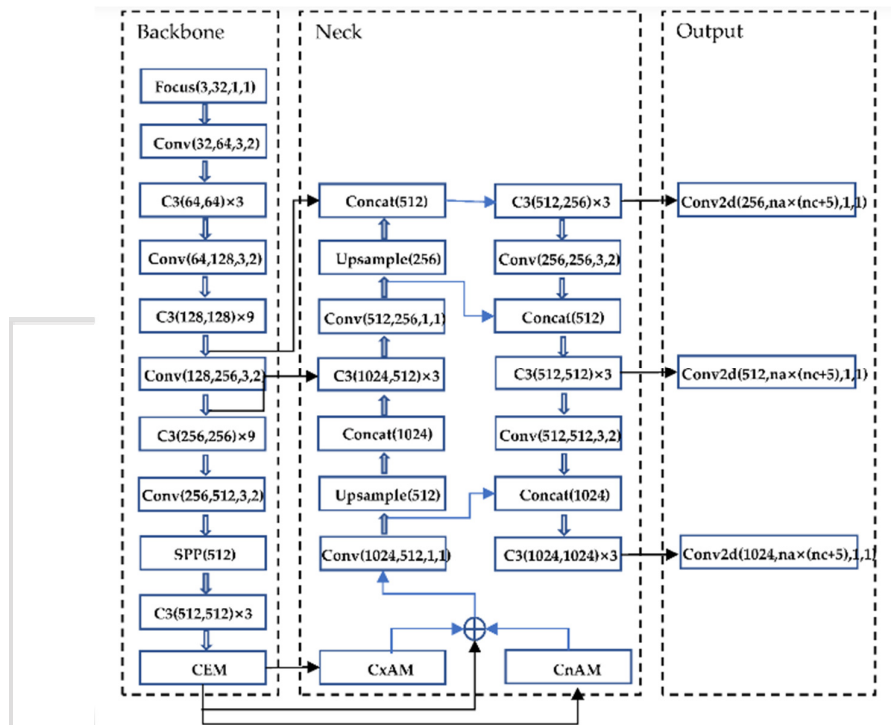
YOLO merupakan metode deep learning yang ditunjukkan untuk mendeteksi objek dan menyatukan komponen-komponen objek deteksi menjadi satu jaringan saraf di seluruh gambar. Pada April 2020 model pendeteksi objek generasi kelima yang dirilis yaitu YOLOv5. Arsitektur pada model ini pada dasarnya tidak jauh berbeda dengan generasi YOLO sebelumnya. Yang membedakan YOLOv5 dengan YOLO sebelumnya yaitu YOLOv5 ditulis dalam bahasa pemrograman Python sedangkan YOLO sebelumnya menggunakan bahasa C. versi terbaru saat ini dari pengembangan YOLO adalah YOLO-v5, berdasarkan YOLO-v4. Pada YOLO-v5 dengan kecepatan tercepat mencapai 140 *frames per second (FPS)*, dimana kecepatan pemrosesan meningkat dengan drastis. YOLO-v5 berukuran 90% lebih kecil dari YOLO-v4, sehingga memungkinkan YOLO-v5 dapat di-deploy ke perangkat yang disematkan. Peningkatan selanjutnya pada YOLO-v5 yaitu tingkat akurasi yang lebih tinggi dan kemampuan yang lebih baik untuk mengenali benda-benda kecil (SUTRISNO 2022).

Konsep dari YOLOv5 adalah dengan membagi dua gambar data input kedalam $S \times S$ grid cell. Setiap cell akan memprediksikan nilai confidence dan juga memprediksi banyak bounding box. Jika tidak ada objek didalam cell tersebut, maka nilai confidence akan menjadi nol. Setelah nilai confidence pada setiap cell didapatkan maka nilainya dibandingkan dengan nilai intersection over union (IoU) untuk mendapatkan bounding box prediksi disetiap kelas yang ada dengan perhitungan (Dio et al. 2022).



Gambar 2.2 Ilustrasi YOLOv5

Algoritma YOLOv5 menggunakan aktivasi leaky RELU dan sigmoid, fungsi optimasi Adaptive Moment (Adam) dan Stochastic Gradient Decent (SGD), serta fungsi loss Binary cross-entropy. Pada YOLOv5, digunakan CSPDarknet sebagai backbone untuk mengekstrak fitur informatif dari gambar input. Jika dibandingkan dengan Darknet53 yang digunakan pada YOLOv3, CSPDarknet telah mencapai peningkatan yang cukup besar dalam kecepatan pemrosesan dengan nilai akurasi yang setara. Pada bagian neck, YOLOv5 menggunakan Path Aggregation Network (PANet) sebagai mekanisme polimerasi parametrik untuk berbagai tingkat detektor. Kisi fitur terhubung ke semua lapisan fitur oleh kumpulan fitur adaptif yang disediakan PANet(Rath 2023) .



Gambar 2.3 Arsitektur YOLOv5

2.3 OpenCV

OpenCV (*Open Computer Vision*) adalah sebuah *library open source* untuk *computer vision* dan *machine learning*. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi *computer vision* dan mempercepat penggunaan *machine perception* pada produk komersil. Menjadi produk yang berlisensi BSD, OpenCV memudahkan bisnis untuk memanfaatkan dan memodifikasi kode. *Library* ini memiliki lebih dari 2500 algoritma yang dioptimalkan, yang mencakup serangkaian komprehensif *computer vision* dan algoritma mesin dan pembelajaran mesin yang mutakhir. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak pergerakan kamera, melacak objek

bergerak, mengekstraksi model objek 3D, menghasilkan awan titik 3D dari kamera stereo, menjahit gambar bersama untuk menghasilkan resolusi tinggi gambar seluruh adegan, temukan gambar serupa dari basis data gambar, hapus mata merah dari gambar yang diambil menggunakan flash, ikuti gerakan mata, kenali pemandangan dan membuat penanda untuk melapisinya dengan augmented reality dan lainnya(Surbakti and Putri 2022).

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) Library yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari *Computer Vision* adalah *Face Recognition*, OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) Library yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari *Computer Vision* adalah *Face Recognition*, *Face Detection*, *Face/Object Tracking*, *Road Tracking*, dll. OpenCV adalah library *Open Source* untuk *Computer Vision* untuk C/C++, OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk image/video.

OpenCV sendiri terdiri dari 5 *library*, yaitu: CV (untuk algoritma Image processing dan Vision); ML (untuk *machine learning library*); Highgui (untuk GUI, *Image* dan Video I/O); dan CXCORE (untuk struktur data, mendukung XML dan fungsi-fungsi grafis)(Sutarti, Samsuni, and Asseghaf 2019).

Object Detection atau deteksi objek merupakan bagian dari *Computer Vision*. *Object Detection* mengacu pada kemampuan komputer untuk mendeteksi sejumlah objek pada suatu gambar. Hal ini dapat dilakukan dengan cara mengambil *image feature* seperti garis, sudut, kontur dan warna dari sebuah gambar. Deteksi objek merupakan bagian dari *Object Recognition* atau identifikasi objek. Oleh karena itu dapat disimpulkan bahwa untuk deteksi objek pasti harus diidentifikasi terlebih dahulu objek tersebut. Sedangkan pada penelitian ini akan dilakukan deteksi objek kemudian di implementasikan deteksi jarak dan perhitungan jumlah objek (Indaryanto, Nugroho, and Suni 2021).

2.4 Python

Python adalah bahasa pemrograman model skrip (*scripting language*) yang berorientasi objek. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. *Python* merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan *source codenya*, *debugger* dan *profiler*, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, *GUI* (antarmuka pengguna grafis), dan basis datanya(Rahma et al. 2021).

Python merupakan bahasa pemrograman, dikembangkan oleh Guido Van Rossum dari Amsterdam, bahasa pemrograman ini bersifat *open source* atau gratis dalam arti yang sebenarnya, sehingga tidak ada batasan dalam pendistribusian maupun penyalinannya, *python* dapat berjalan dalam beberapa sistem informasi kebanyakan seperti *UNIX, PCs(DOS, WINDOWS, OS/2), Macintosh*, dan lainnya. terkhusus *Linux*, bahasa ini telah menjadi standarisasi dan disertakan pada paket instalasi di beberapa *distro linux* (Saputra et al. 2022).

2.5 Framework Flask Python

Flask merupakan web *framework* yang digunakan oleh *Python*. *Flask framework* tergolong ke dalam *micro-framework* karena tidak membutuhkan *tools* atau *library* tertentu serta memiliki database bawaan (Singh et al. 2019). *Flask* menggunakan *Jinja Template Engine* dan *Werkzeug WSGI ToolKit*. *Flask* menyusun kategorinya menjadi dua bagian yaitu: *Static File* yang memiliki semua kode status yang dibutuhkan untuk website seperti kode *CSS*, kode *JavaScript* dan file gambar, dan *Template File* yang berisi semua template *Jinja* termasuk halaman *HTML* (Ningtyas and Setiyawati 2021).

Python merupakan bahasa pemrograman yang berorientasi obyek dinamis, mudah dipahami, dan dapat digunakan untuk bermacam-macam pengembangan perangkat lunak. *Python* hadir dengan pustaka-pustaka standar yang dapat diperluas serta dapat dipelajari hanya dalam beberapa hari (Fitri, Rahmansyah, and Darwin 2017). Salah satu *framework* yang dimiliki oleh *Python* adalah *Flask*. Untuk membuat pengembangan lebih efisien maka digunakan *Flask* (Aziz and Anggora 2021). *Flask* diterapkan karena dapat berjalan pada program yang

berkapasitas energi kecil serta dengan *memory* yang rendah sehingga tidak memerlukan *resource* yang besar. Walaupun *framework Flask* ringan akan tetapi fungsinya masih bisa berjalan sesuai yang dibutuhkan (Ngantung and Pakereng 2021). Terdapat dua komponen dalam *Flask* yaitu Werkzeug dan Jinja2, Werkzeug yang berperan dalam menyediakan *routing*, *debugging*, dan *Web Server Interface Gateway (WSGI)* sedangkan Jinja2 sebagai *template engine*.

2.6 Cuda

Salah satu pengembang teknologi GPU untuk pasar profesional dan konsumen yaitu NVIDIA. NVIDIA mengembangkan CUDA (*Compute Unified Device Architecture*) yang merupakan platform pemrograman paralel dan API (*Application Programming Interface*) (Razak, Muttaqin, and Pradipta 2021). CUDA ini memungkinkan developer perangkat lunak untuk menggunakan GPU keluaran NVIDIA sebagai wadah pemrograman paralel. GPU seri Tesla dan GeForce RTX (high end GPU) merupakan contoh GPU keluaran NVIDIA yang umumnya diproduksi untuk pasar profesional, sehingga GPU seri ini memiliki kemampuan proses tiap thread yang cepat dan kuantitas thread yang banyak untuk memenuhi tuntutan pasar profesional dalam komputasi paralel. Sedangkan untuk pasar konsumen, NVIDIA mengeluarkan seri GeForce GTX (low end) yang memiliki kemampuan proses dan kuantitas thread lebih kecil dibanding seri Tesla dan GeForce RTX (Razak, Muttaqin, and Pradipta 2021).

CUDA dalam pemrogramannya membagi device ke dalam grid, block, dan thread. Jika dikomparasi dengan OpenCL. CUDA umumnya dipakai untuk pengolahan grafis seperti pada image dan video *processing* secara digital seperti

classification with localization, image segmentation, perubahan citra, pengenalan pola dan dapat diaplikasikan pada berbagai multi-disiplin keilmuan (Jupiyandi et al. 2019).

2.7 Blackbox Testing

Metode *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail software. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang output pakai. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan (Ilham et al. 2021)

2.8 Google Colab

Google colabatory atau Google colab adalah sebuah tools yang dikembangkan oleh Google. Tools ini memberikan fasilitas untuk mengolah data menggunakan teknik machine learning maupun deep learning, namun tool ini memiliki keterbatasan perangkat untuk melakukan komputasi. Google Colab menyediakan layanan GPU gratis sebagai backend komputasi yang dapat digunakan selama 12 jam. Google Colab dibuat diatas environment Jupyter sehingga mirip dengan Jupyter Notebook. Penggunaannya pun hampir sama dengan Jupyter Notebook hanya saja berbeda dalam hal media penyimpanannya. Media penyimpanan pada Google Colab adalah Google Drive dan tools ini berjalan pada sistem cloud. Google Colab menyediakan runtime Python 2 dan 3

yang telah dikonfigurasi sebelumnya dengan berbagai library, seperti TensorFlow, Matplotlib, dan Keras] (Kembuan, Rorimpandey, and Tengker 2020).

2.9 Roboflow

Roboflow merupakan sebuah platform yang tersedia di web yang memiliki banyak fungsi yang berhubungan dengan dataset. Dengan menggunakan roboflow ini dapat membagikan dataset sekaligus memproses dataset tersebut. Beberapa fitur yang digunakan pada penelitian ini diantaranya adalah melakukan annotate atau menandai objek yang akan di deteksi menggunakan bounding box, selain itu dapat digunakan juga pre-processing pada dataset misalnya melakukan grayscale, dan juga augmentasi dengan menggunakan roboflow. Roboflow ini menyediakan beberapa jenis augmentasi konvensional seperti *flip*, *rotate*, *brightness*, *exposure*, *shear* dan lain-lain. Berikut beberapa augmentasi konvensional yang disediakan roboflow pada gambar (Taufiq, Rizki, and Pratama 2022).



Gambar 2.4 Roboflow

BAB III METODOLOGI PENELITIAN

3.1 Tempat dan Waktu

Penelitian ini dilakukan di Kantor Badan Pengelolaan Keuangan dan Aset Daerah Kota Palopo pada Bulan Maret sampai Juli 2023.

3.2 Kebutuhan Sistem

Adapun beberapa kebutuhan sistem digunakan pada penelitian ini dibagi menjadi dua bagian yaitu perangkat keras (*hardware*) dan Perangkat Lunak (*Software*)

3.2.1. Perangkat Keras (*Hardware*)

Hardware adalah alat fisik komputer yang memiliki fungsi untuk melakukan beberapa proses seperti input, output, dan proses.

Tabel 3. 1 Perangkat Keras (*Hardware*)

NO	Perangkat keras	Deskripsi
1	Laptop / PC : a. Minimal Processor AMD Ryzen 5 3550H b. Memory RAM 16 GB c. SSD 512	Digunakan untuk merancang sistem deteksi serta untuk <i>script</i>
2	Graphic Card GPU Nvidia Geforce GTX 1650, 4GB	Digunakan untuk <i>run</i> pada CCTV yang sedang <i>real time stream</i>
3	Camera CCTV Merek HIKVISION	Digunakan untuk sebagai alat untuk mengawasi area sekitar
4	Monitor	Digunakan untuk media monitoring kejadian pada CCTV secara langsung

3.2.2. Perangkat Lunak (*Software*)

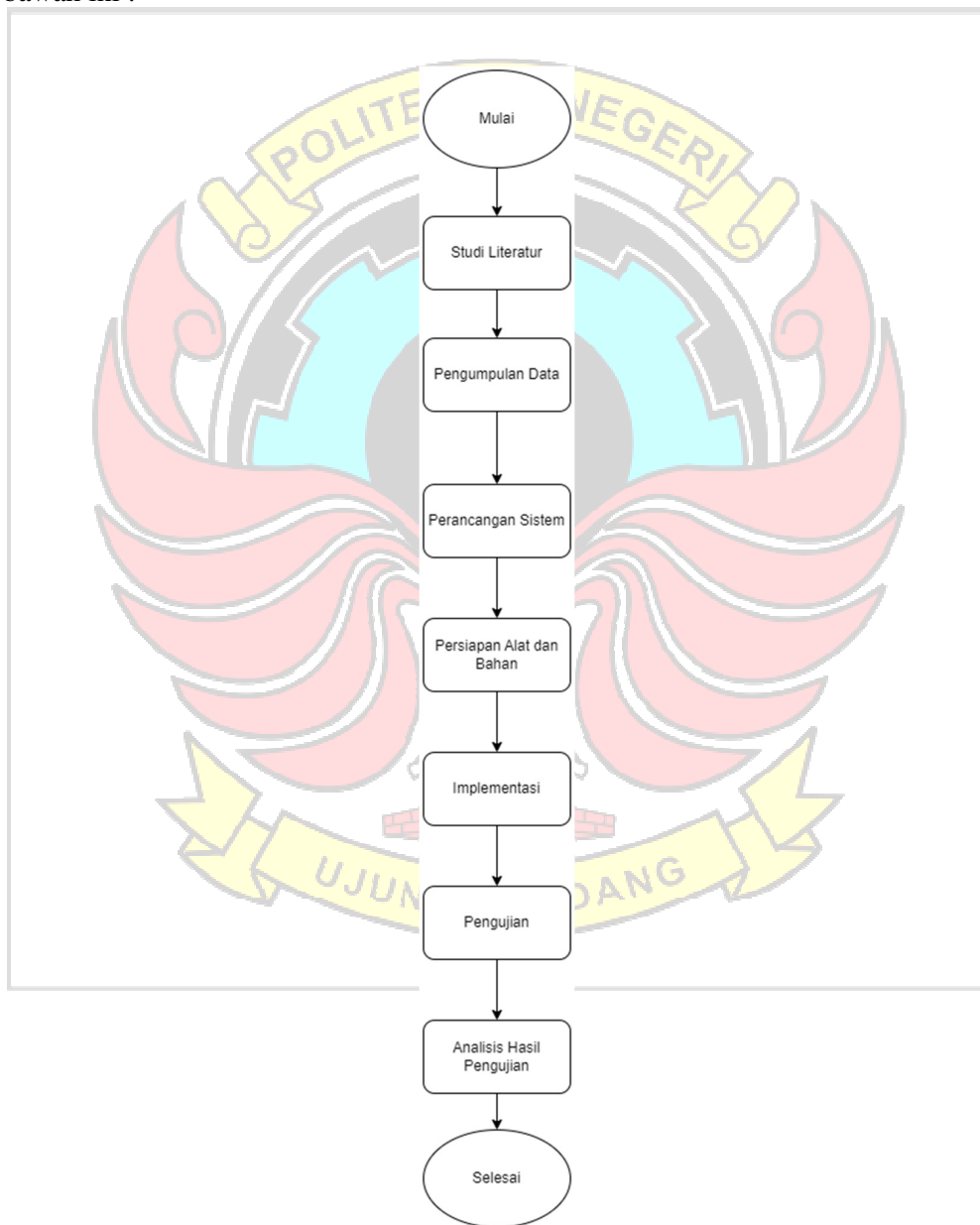
Merupakan kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya. perangkat lunak ini merupakan catatan bagi mesin komputer untuk menyimpan perintah, maupun dokumen serta arsip lainnya.

Tabel 3. 2 Perangkat Lunak (*Software*)

No	Perangkat Lunak	Deskripsi
1	Sistem Operasi Windows 11	Sebagai sistem operasi yang digunakan untuk melakukan pemrograman <i>Machine learning</i>
2	Yolov5	Digunakan untuk sebagai pendeteksi objek
3	Visual Studio Code	Digunakan untuk sebagai aplikasi editor melakukan penulisan <i>script</i> pada <i>machine learning</i>
4	Python	Digunakan untuk menjalankan program yang telah dirancang
5	XAMPP	Sebagai <i>tools</i> untuk database dan server php.
6	Cuda	Berfungsi sebagai grafik <i>streaming</i>
7	MySQL	Berfungsi sebagai database
8	OpenCV	Berfungsi untuk mengelola atau proses <i>Streaming CCTV</i> ke web server dan Yolov5
9	Library Pendukung Lainnya	Berfungsi sebagai pendukung untuk menjalankan YOLOv5
10	Telegram	Berfungsi mengirimkan hasil kegiatan merokok pegawai yang melakukan kegiatan merokok di kawasan tanpa rokok

3.3 Metode Penelitian

Metode Penelitian dibutuhkan dapat terarah dan terstruktur sehingga menghasilkan yang sesuai seperti tujuan penelitian. Metode penelitian yang digunakan dalam penelitian ini digambarkan diagram alir pada Gambar 3.1 di bawah ini :



Gambar 3.1 Tahapan Metode Penelitian

3.3.1 Studi Literatur

Studi literatur dilakukan untuk memperoleh atau mengumpulkan informasi atau data yang akurat dari berbagai sumber informasi baik berupa buku, jurnal, skripsi, atau tesis yang berkaitan dengan penelitian ini. Studi literatur merupakan kegiatan untuk memahami informasi yang sesuai dengan kasus terkait dari penelitian yang telah dilakukan. Berdasarkan studi literatur yang telah dilakukan, penulis memilih untuk menggunakan Yolov5 dan RTSP untuk konsep mendeteksi pegawai yang melakukan merokok di area kawasan tanpa merokok. Dengan menggunakan algoritma Yolov5 digabungkan dengan metode RTSP bukti pelanggaran yang telah dilakukan pegawai akan tampil di dalam telegram sehingga memudahkan untuk mengetahui pegawai pegawai yang melakukan kegiatan merokok di kawasan tanpa rokok.

3.3.2 Pengumpulan Data

Pada pengumpulan data penelitian ini bertujuan untuk sebagai dataset penelitian. Adapun data yang digunakan pada penelitian ini berupa gambar. Pada gambar dataset berupa gambar rokok yang telah di bakar maupun belum di bakar selanjutnya dijadikan sebagai bahan training dari sistem ini dalam mendeteksi terjadinya pegawai merokok selain itu telapak tangan dengan jari memegang rokok dan punggung tangan dengan jari memegang rokok juga menjadi dataset untuk melatih model tersebut agar dapat mendeteksi pegawai yang merokok di kawasan tanpa rokok.

Berikut parameter yang digunakan dalam mendeteksi faktor terjadinya pegawai merokok di dalam ruangan seperti :

a) Deteksi rokok

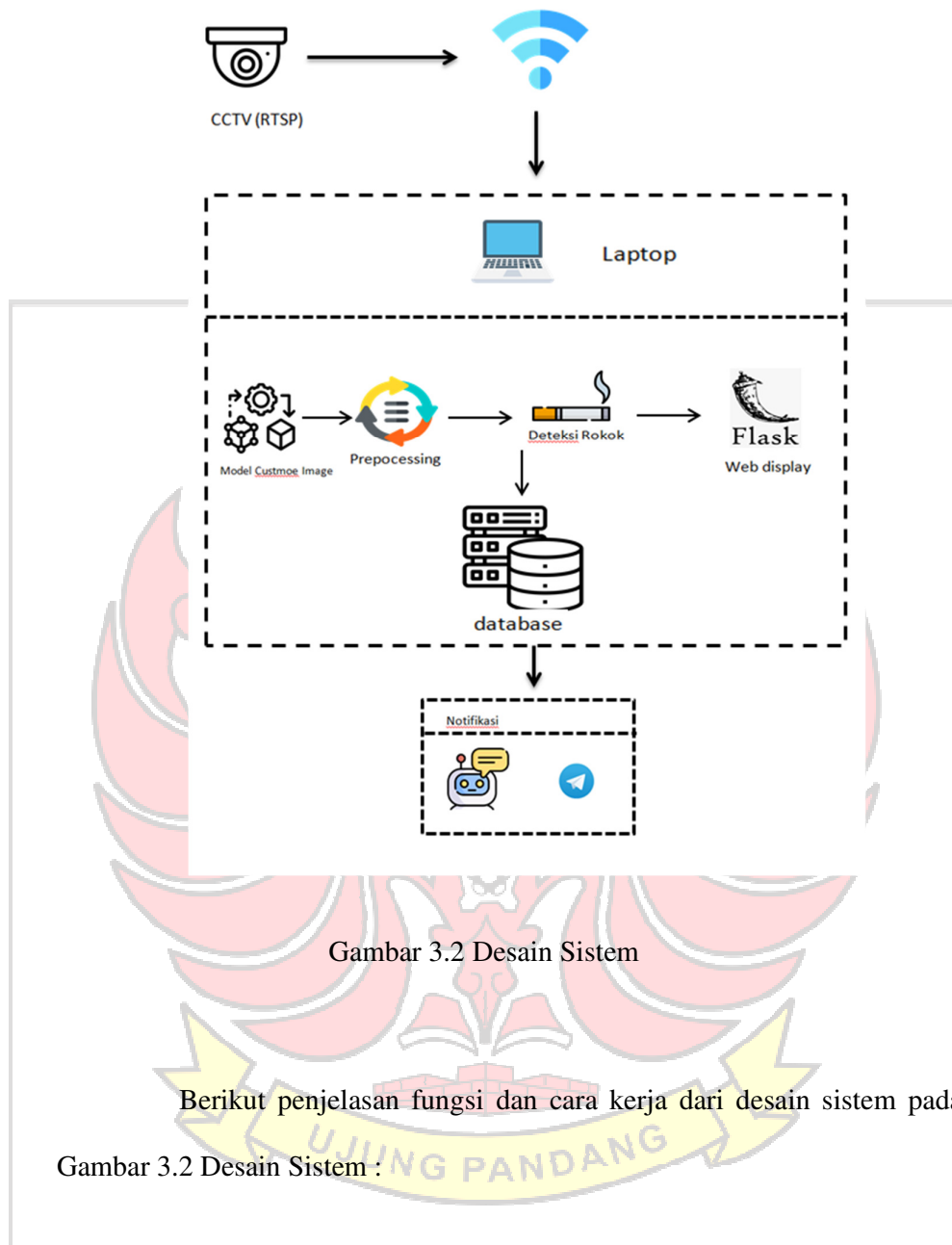
mendeteksi benda rokok sistem melakukan pendeteksi benda rokok yang berada di mulut maupun di sekitar mulut dengan menggunakan sistem *Roboflow* gambar rokok dapat dilakukan *annotate* secara manual sehingga memudahkan untuk menandai rokok tersebut selain itu menggunakan *Roboflow* juga menampilkan hasil *mAP* yang berfungsi untuk mengetahui presentase hasil semua gambar yang telah di *Annotate*

b) Deteksi telapak dan punggung tangan dengan jari memegang rokok

Selain benda rokok sistem juga melakukan pendeteksi telapak dan punggung tangan dengan jari memegang rokok yang bertujuan apabila objek yang terlalu jauh dari CCTV maka sistem mendeteksi telapak dan punggung tangan dengan jari memegang rokok dengan melatih gambar menggunakan *roboflow* agar sistem dapat mengenali gambar tersebut.

3.3.3 Perancangan Sistem

Pada tahapan perancangan sistem menjelaskan gambaran sistem yang dibuat sebagai acuan dalam penelitian yang dikerjakan. Pada perancangan sistem Mendeskripsikan struktur sistem secara keseluruhan. Ini bisa mencakup bagaimana perangkat keras dan perangkat lunak akan saling berinteraksi, serta bagaimana data akan mengalir melalui sistem.



Gambar 3.2 Desain Sistem

Berikut penjelasan fungsi dan cara kerja dari desain sistem pada Gambar 3.2 Desain Sistem :

1) CCTV

CCTV merupakan perangkat keras berfungsi sebagai menampilkan visual gambar di dalam ruangan. CCTV melakukan perekaman secara otomatis secara real time pada ruangan aset. Selain itu CCTV terhubung ke sistem dengan memberikan URL, RTSP port, Stream Type, dan nomor channel dari

CCTV. Sistem CCTV dapat mengelolah video dengan maksimal *resolution* yaitu 1920 x 1080 dengan maksimal *main stream* 25 fps.

2) Laptop

Laptop merupakan perangkat yang berfungsi untuk mengelolah aplikasi deteksi, menampilkan hasil visual dari CCTV ke web display, ketika terjadi

bounding box maka melakukan *screenshot* sehingga menyimpan bukti kegiatan merokok tersebut, menyimpan bukti kegiatan merokok ke dalam database, dan mengirimkan notifikasi telegram pada bot telegram

3) Model Custome Image

Model *Custome Image* merupakan pengolahan dataset gambar yang dijadikan objek serta digunakan sebagai model gambar pada Yolov5 untuk mendeteksi benda rokok.

4) Preprocessing

Preprocessing merupakan pengolahan OpenCV, Yolov5, dan Cuda. Pada Tahapan OpenCV digunakan untuk mengakses dan mengambil aliran video secara *real-time* pada CCTV selain itu OpenCV menampilkan hasil *bounding box* yang telah diolah pada sistem Yolov5. OpenCV juga berperan untuk menampilkan visual rekaman CCTV pada Flask. Pada tahapan Yolov5 digunakan untuk melakukan deteksi objek secara realtime dengan kecepatan

tinggi serta mengenali objek dalam citra atau video. Yolov5 dapat mengalinisis video secara kontinu selain itu Yolov5 memiliki kemampuan untuk melacak objek yang berguna pemantauan objek yang bergerak dan pelacakan yang berkelanjutan. Pada tahapan Cuda digunakan oleh Yolov5

untuk menghasilkan peningkatan signifikan dalam kecepatan deteksi. Dengan adanya Cuda model-model YOLOv5 dapat mengatasi banyak objek secara paralel, menghasilkan deteksi yang cepat dan akurat. Cuda juga dapat digunakan dalam pelatihan model-model YOLOv5.

5) Deteksi Rokok

Deteksi rokok dilakukan saat hasil visual rekaman CCTV diproses oleh OpenCV dan terdeteksi oleh YOLOv5, maka YOLOv5 melakukan pengaplikasian kotak pembatas (*bouding box*) terhadap objek rokok tersebut. Hasil dari *bouding box* YOLOv5 diolah oleh OpenCV dan ditampilkan pada halaman web.

6) Flask

Flask merupakan framework ditulis dalam bahasa pemrograman Python. Flask ini berfungsi untuk menampilkan visual CCTV secara *streaming* di halaman web.

7) Database

Database digunakan untuk menyimpan bukti bukti kegiatan merokok pegawai yang telah terdeteksi oleh sistem. Apabila sistem mendeteksi pegawai melakukan kegiatan merokok sistem melakukan pendeteksian dan *bouding box* pada objek rokok sehingga hasil rekaman video diubah menjadi

file gambar maka file gambar yang menjadi bukti kegiatan merokok tersebut dikirim sistem ke dalam database untuk disimpan. Database telah dapat menyimpan semua bukti pelanggaran yang telah di *bouding box* oleh sistem.

8) Telegram

Telegram merupakan platform pesan instan atau aplikasi pengiriman pesan. Telegram tersebut berfungsi untuk mengirimkan bukti kegiatan merokok pegawai dalam bentuk file gambar. Sistem melakukan identifikasi objek di dalam ruangan dengan bantuan rekaman CCTV secara real time.

Apabila sistem mendeteksi adanya kegiatan merokok pegawai dalam ruangan tersebut sistem mendeteksi kegiatan merokok tersebut dan sistem mengirimkan bukti kegiatan merokok kepada bot telegram yang sebelumnya telah dibuat.

1) *Desain sistem software*

Desain sistem software adalah sebuah gambaran sistem software yang digunakan dalam mendeteksi objek, dalam mendeteksi objek rokok, telapak tangan dengan jari memegang rokok, dan punggung tangan dengan jari memegang rokok. ada beberapa proses yang harus dilakukan agar gambar dataset tersebut dapat digunakan pada model Yolov5. Dapat dilihat pada gambar 3.3 merupakan alur proses sehingga data set gambar dapat digunakan pada model Yolov5.

a) *Custom dataset gambar*

Berikut penjelasan pada Gambar 3.3 sebagai berikut :

1. Dataset

Pada tahapan awal mengumpulkan gambar rokok dengan model gambar rokok yang berbeda beda untuk dijadikan sebagai dataset.

2. Labelling Data

Tahapan selanjutnya labelling data pada tahapan tersebut melakukan penandaan terhadap objek rokok dan gambar model model rokok pada setiap dataset dan memberikan *class* pada objek yang telah di *labelling*.

3. Preprocessing

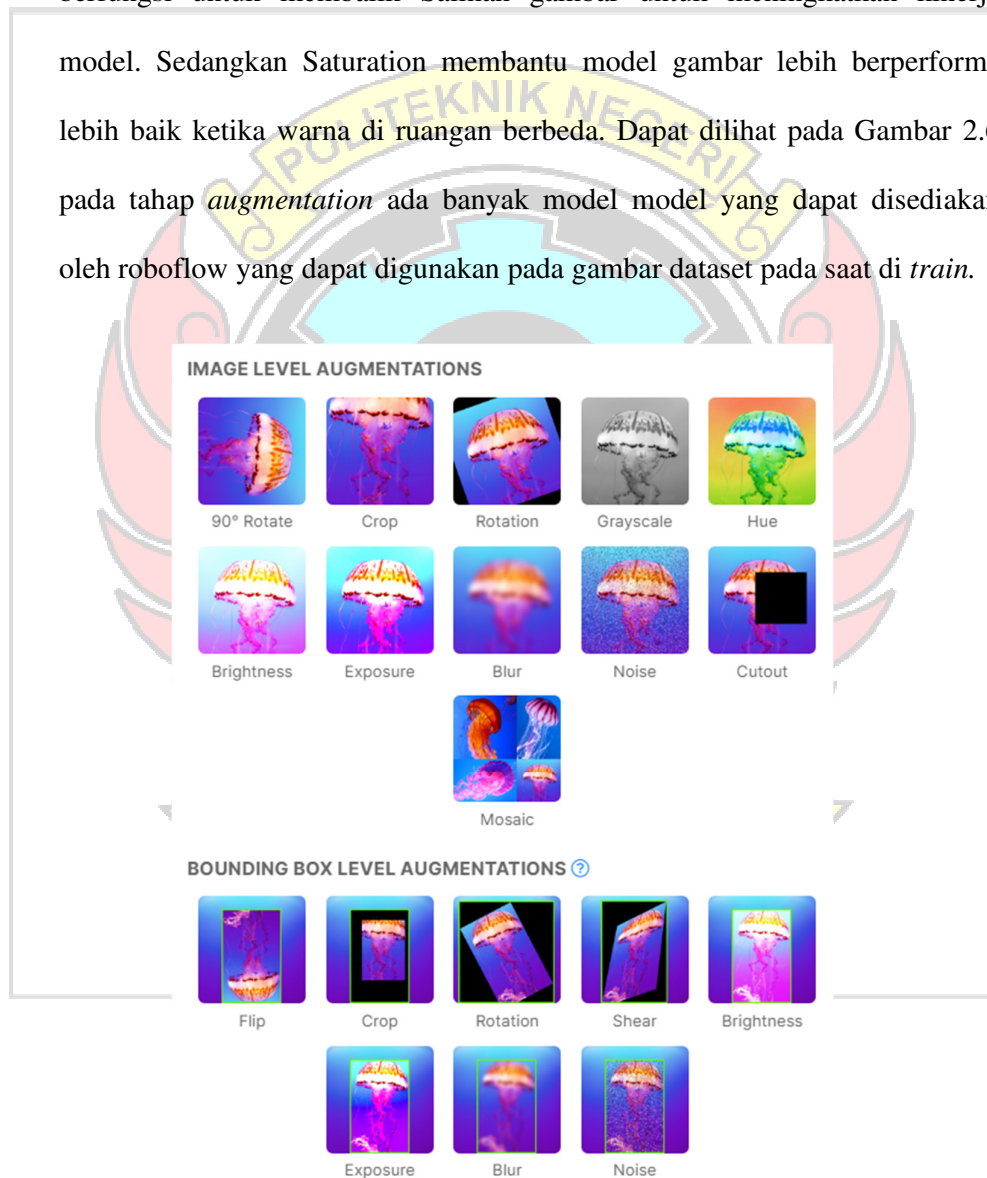
Tahapan selanjutnya adalah *Preprocessing* pada tahapan ini dataset yang telah di labelling dilakukan pelatihan model. Pada pelatihan model tersebut menggunakan 2 model pelatihan yaitu Auto-orient yang berfungsi untuk mengidentifikasi orientasi yang benar dari gambar dan secara otomatis memutar atau mengubah orientasi gambar yang tidak sesuai. Model yang kedua yaitu Resize image 380 x 640 berfungsi untuk mengubah semua ukuran dataset menjadi 380 untuk lebar dan 640 untuk tinggi sehingga dapat menghasilkan akurasi yang tinggi. Dapat dilihat pada Gambar 2.5 terdapat beberapa metode pada tahap *preprocessing* yang dapat digunakan pada gambar dataset yang di *train*.



Gambar 2.5 metode yang tersedia pada preprocessing

4. Augmentation

Tahapan selanjutnya adalah augmentation merupakan proses menambahkan variasi terhadap dataset untuk meningkatkan generalisasi model. Pada tahap ini menggunakan 2 variasi yaitu Flip dan Saturation. Flip berfungsi untuk membalik Salinan gambar untuk meningkatkan kinerja model. Sedangkan Saturation membantu model gambar lebih berperforma lebih baik ketika warna di ruangan berbeda. Dapat dilihat pada Gambar 2.6 pada tahap *augmentation* ada banyak model model yang dapat disediakan oleh roboflow yang dapat digunakan pada gambar dataset pada saat di *train*.



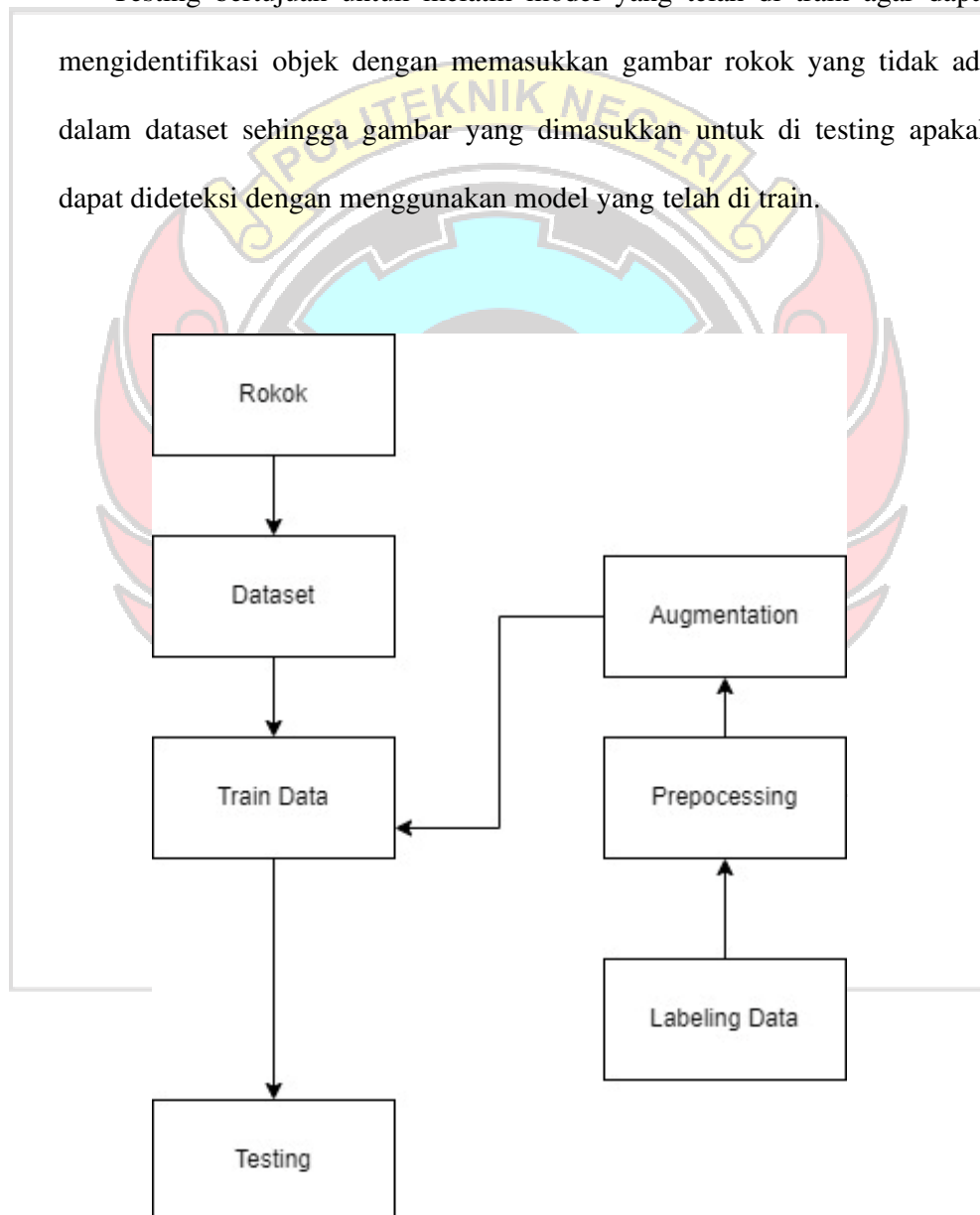
Gambar 2.6 model mode yang tersedia pada augmentation

5. Train Data

Train Data dilakukan setelah model model yang telah dipilih dilakukan pelatihan terhadap dataset gambar rokok tersebut.

6. Testing

Testing bertujuan untuk melatih model yang telah di train agar dapat mengidentifikasi objek dengan memasukkan gambar rokok yang tidak ada dalam dataset sehingga gambar yang dimasukkan untuk di testing apakah dapat dideteksi dengan menggunakan model yang telah di train.



Gambar 3.3 Block Diagram Custom Mode

a) Custmoe Model

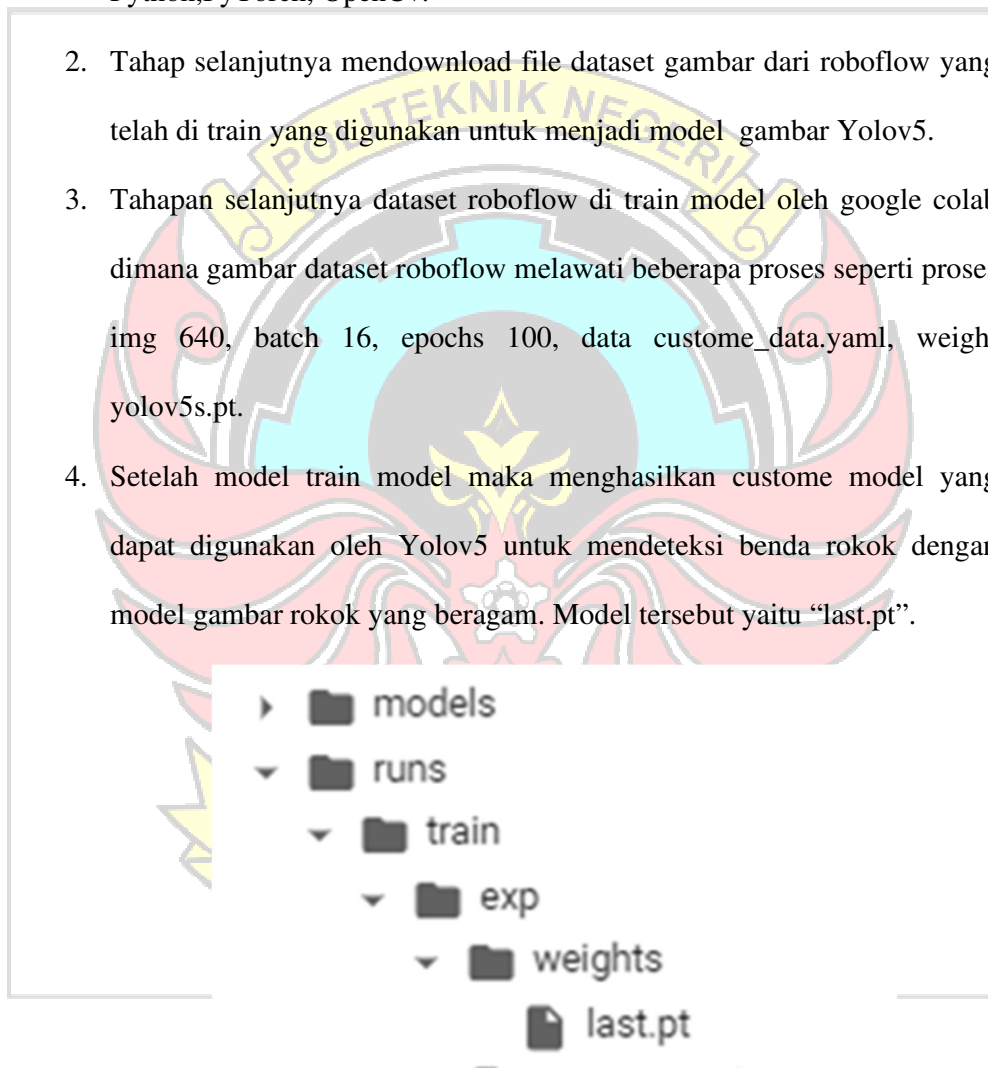
Berikut penjelasan pada Gambar 3.4 sebagai berikut :

1. Pada tahap awal mengklone (mendownload) repositori Yolov5 dari github ke dalam google colab dan menginstall requirement yaitu Pustaka Python,PyTorch, OpenCv.

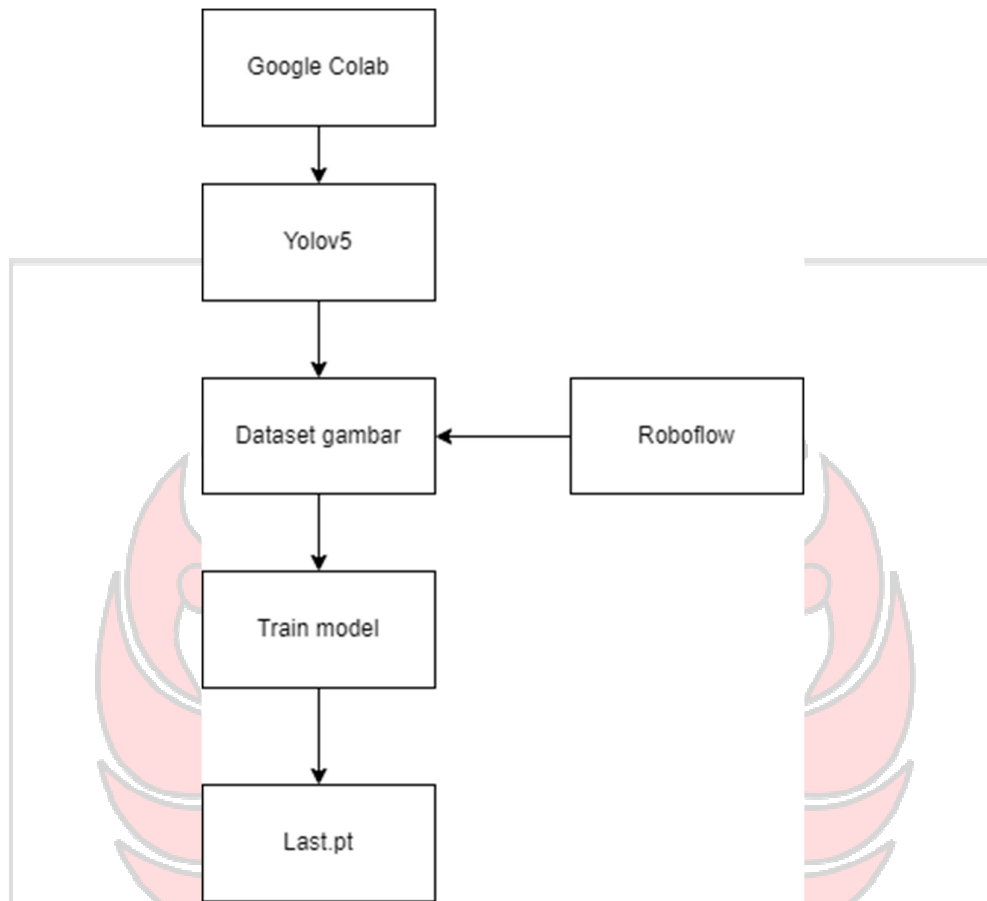
2. Tahap selanjutnya mendownload file dataset gambar dari roboflow yang telah di train yang digunakan untuk menjadi model gambar Yolov5.

3. Tahapan selanjutnya dataset roboflow di train model oleh google colab dimana gambar dataset roboflow melawati beberapa proses seperti proses img 640, batch 16, epochs 100, data custome_data.yaml, weight yolov5s.pt.

4. Setelah model train model maka menghasilkan custome model yang dapat digunakan oleh Yolov5 untuk mendeteksi benda rokok dengan model gambar rokok yang beragam. Model tersebut yaitu “last.pt”.



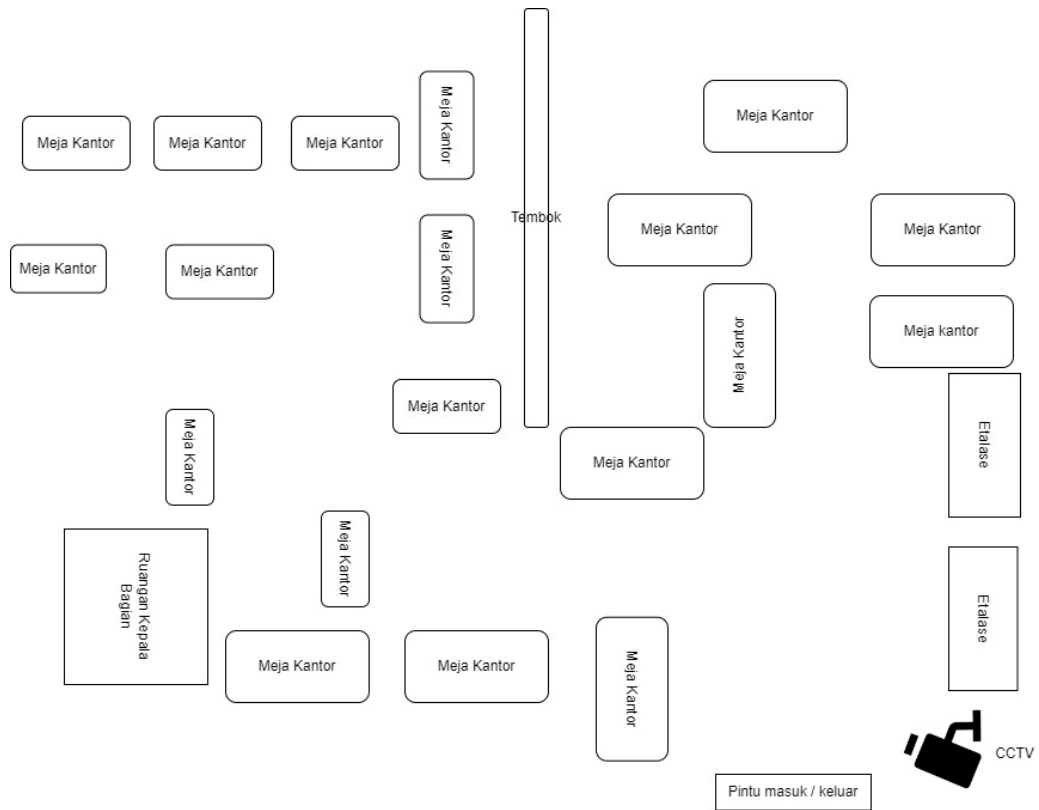
Gambar 2.7 model yang digunakan Yolov5 untuk mendeteksi objek rokok



Gambar 3.4 Custome Model

2) *Kondisi Ruangan*

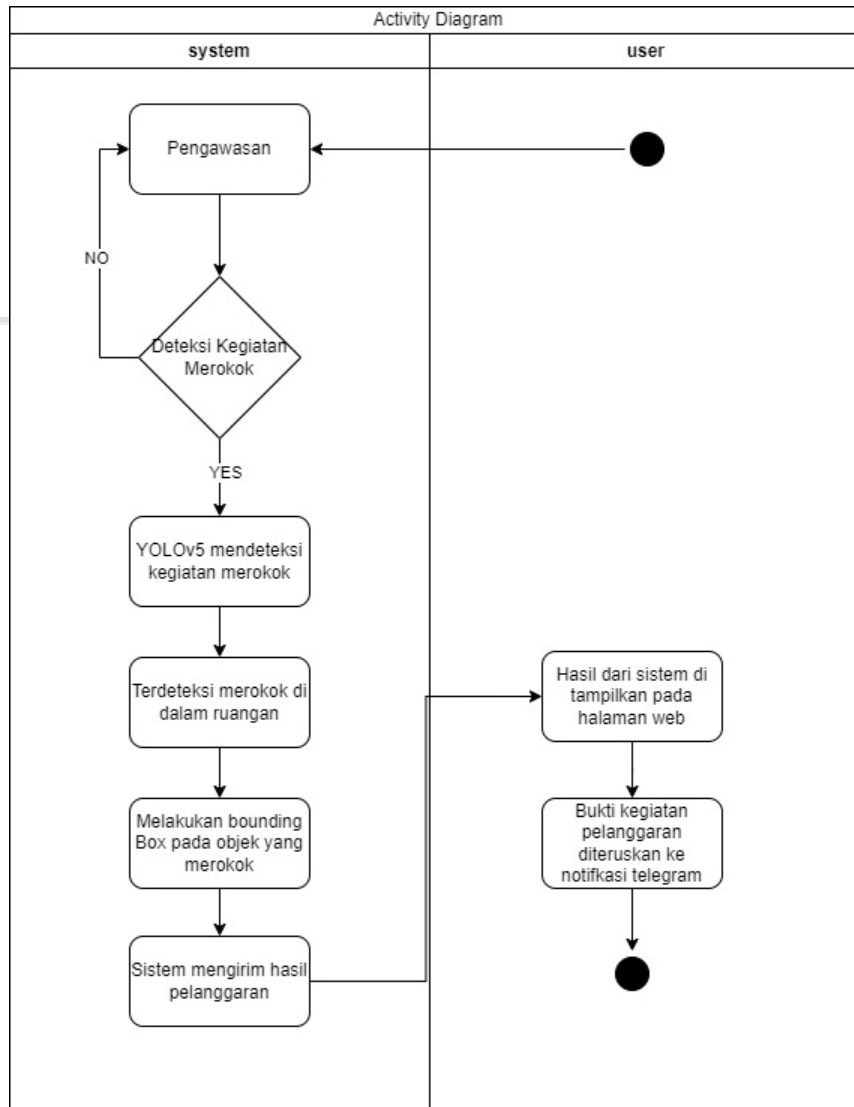
Kondisi Ruangan adalah sebuah gambaran denah ruangan yang menjadi pengujian sistem tersebut. Dari gambar denah tersebut menjelaskan tata letak benda dan tata letak CCTV yang digunakan untuk memantau pegawai kantor untuk tidak melakukan merokok di kawasan tanpa rokok. Pada gambar 3.5 adalah desain denah ruangan tempat CCTV yang bekerja.



Gambar 3.5 Denah Ruangan

3) *Activity Diagram*

Activity Diagram berguna untuk memvisualisasikan rangkaian aktivitas yang terjadi, aliran informasi, pengambilan keputusan, dan urutan tindakan dalam suatu sistem. Pada Gambar 3.6 merupakan *activity diagram* proses sistem CCTV mendeteksi pegawai yang melakukan merokok pada kawasan area tanpa rokok.



Gambar 3.6 Activity Diagram

3.3.4 Implementasi

Pada tahapan Implementasi merupakan tahap dimana suatu sistem telah dirancang untuk dioperasikan dalam keadaan yang sebenarnya. Sehingga melalui tahap ini sistem dapat diketahui kelayakannya dalam penggunaan.

3.3.5 Pengujian

Pada tahapan pengujian ini bertujuan untuk melakukan pengujian terhadap seluruh sistem yang telah dibangun dan dikonfigurasi.

1) Pengujian Fungsional

Pengujian fungsional menggunakan metode *black box* berguna untuk memastikan bahwa sistem input dan output dapat berjalan sesuai dengan sistem yang telah dibuat. Terdapat skenario yang diuji menggunakan *black box testing* dapat dilihat Table 3.1.

Table 3.1 *Black Box Testing*

Pengujian	Bahan uji	Hasil Pengujian
Black Box	Pembuatan Model	Keterangan
	Terhubung dengan CCTV	
	Mendeteksi objek	
	Menyimpan data ke database	
	Menampilkan ke flask	
	Mengirimkan notifikasi ke telegram	

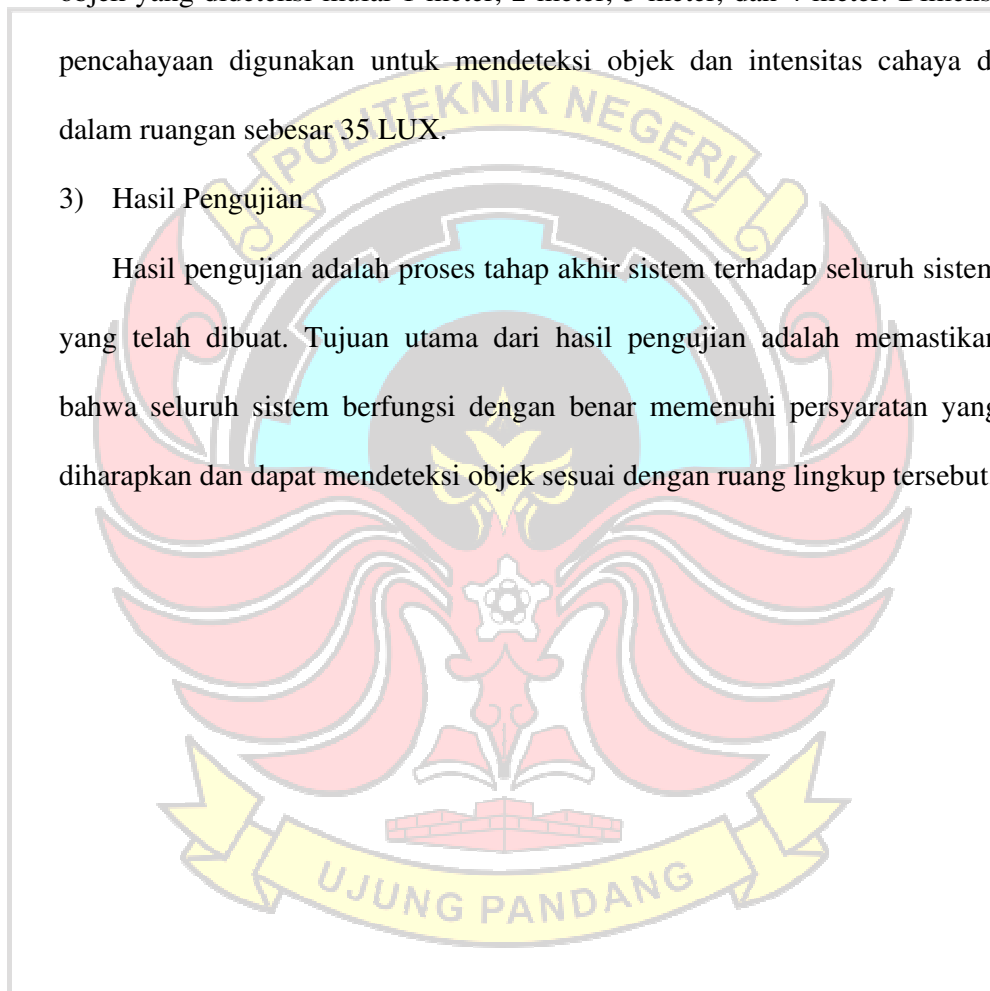
2) Skenario Pengujian

Skenario Pengujian adalah rangkaian langkah atau situasi yang telah direncanakan. Skenario pengujian didasarkan penelitian valid dan dirancang dengan sistematis dan objektif. Di dalam skenario pengujian yang dilakukan dengan mengikuti ruang lingkup penelitian. Pada pengujian tersebut mencakup dimensi sudut, dimensi jarak, dimensi Pencahayaan. Pada dimensi

sudut CCTV dapat menampilkan 110 derajat pandangan ruangan sedangkan untuk sudut objek yang diuji menggunakan sudut istimewa yaitu 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat. Dimensi jarak pada CCTV yaitu jarak ketinggian CCTV di dalam ruangan adalah 3 meter sedangkan jarak objek yang dideteksi mulai 1 meter, 2 meter, 3 meter, dan 4 meter. Dimensi pencahayaan digunakan untuk mendeteksi objek dan intensitas cahaya di dalam ruangan sebesar 35 LUX.

3) Hasil Pengujian

Hasil pengujian adalah proses tahap akhir sistem terhadap seluruh sistem yang telah dibuat. Tujuan utama dari hasil pengujian adalah memastikan bahwa seluruh sistem berfungsi dengan benar memenuhi persyaratan yang diharapkan dan dapat mendeteksi objek sesuai dengan ruang lingkup tersebut.



BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil dari penelitian ini adalah sebuah sistem kamera CCTV yang berbasis RTSP yang dapat melakukan pendeteksi kegiatan merokok di dalam ruangan kerja dengan jarak pegawai yang melakukan kegiatan merokok 1 meter, 2 meter, 3 meter, dan 4 meter dari jarak CCTV dengan ketinggian CCTV 3 meter. CCTV dapat melakukan pemantauan selama jam kerja dengan berbasis *real time stream protocol* dengan menampilkan visual CCTV kedalam tampilan web. Apabila pegawai terdeteksi melakukan kegiatan merokok dari CCTV maka OpenCV mengolah visual CCTV selanjutnya Yolov5 melakukan *bounding box* benda rokok. Yolov5 melakukan *screenshot* terhadap bounding box objek tersebut dan hasil dari *screenshot* berupa file gambar yang disimpan oleh database, sistem mengirimkan bukti kegiatan merokok kepada *bot* telegram dalam bentuk file gambar.

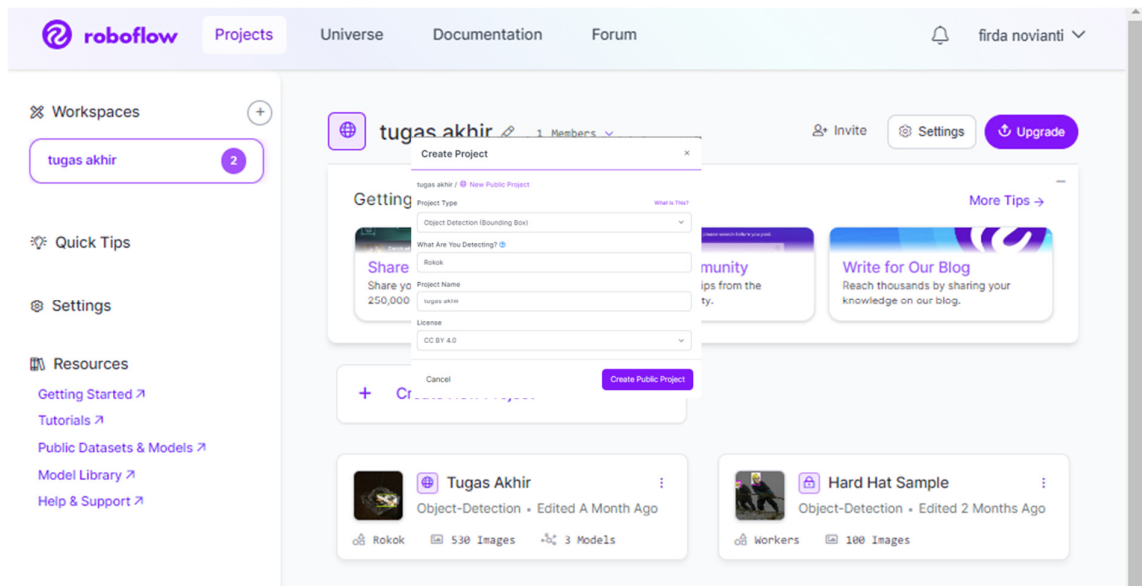
4.2 Hasil Pembuatan Sistem

Hasil pembuatan sistem adalah pembuatan sistem dan model yang digunakan untuk melakukan pendeteksian objek rokok pada kawasan tanpa rokok. Adapun proses dan hasil pembuatan sistem sebagai berikut

4.2.1 Pembuatan model

Pembuatan model tersebut melibatkan Roboflow sebagai pengolahan dan pengelolaan dataset gambar serta google colab sebagai pelatihan model.

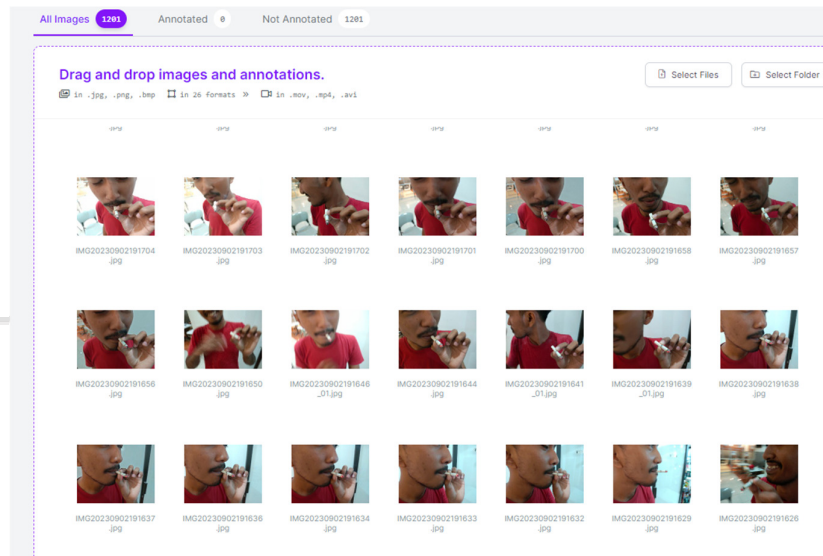
Pada tahap pertama akses website Roboflow.com dan melakukan pembuatan akun pada Roboflow.



Gambar 4.1 Pembuatan workspace

Dapat diperhatikan pada Gambar 4.1 Telah berhasil membuat *workspace* dengan nama tugas akhir. Pada workspace dapat melakukan *Project Custom* Model yang dibuat. Dapat dilihat pada Gambar 4.1 project type yang digunakan yaitu *object detection* (bounding box) hal ini bertujuan karena model tersebut digunakan Yolov5 untuk mendeteksi objek.

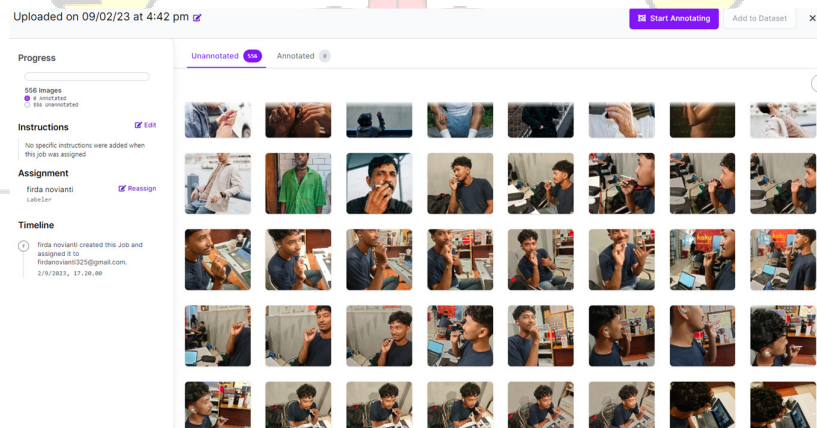
- 1) Setelah berhasil selanjutnya *upload* gambar dataset kedalam project yang telah dibuat. Pada dataset yang digunakan adalah gambar rokok. Adapun jumlah gambar sebanyak 1201.



Gambar 4.2 Upload gambar dataset

Dapat dilihat pada Gambar 4.2 telah berhasil melakukan upload gambar rokok maka langkah selanjutnya adalah melakukan *annotate* secara manual pada setiap gambar yang telah di upload.

- 2) Langkah berikutnya adalah melakukan annotating gambar, pada Gambar 4.3. terdapat dataset gambar yang dilakukan annotating dengan mengklik *button start annotating*



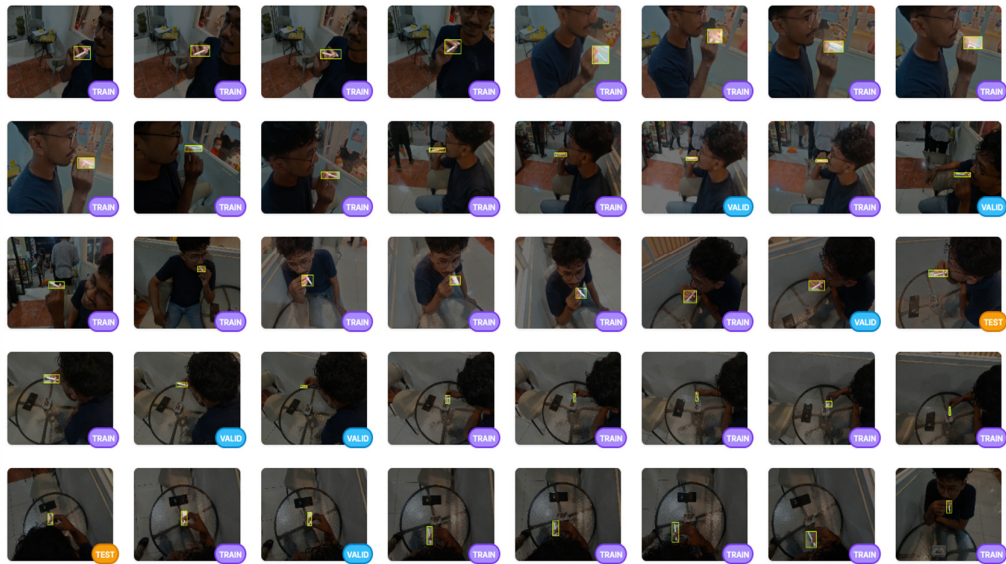
Gambar 4.3 gambar dataset untuk annotating



Gambar 4.4 gambar rokok di Annotating

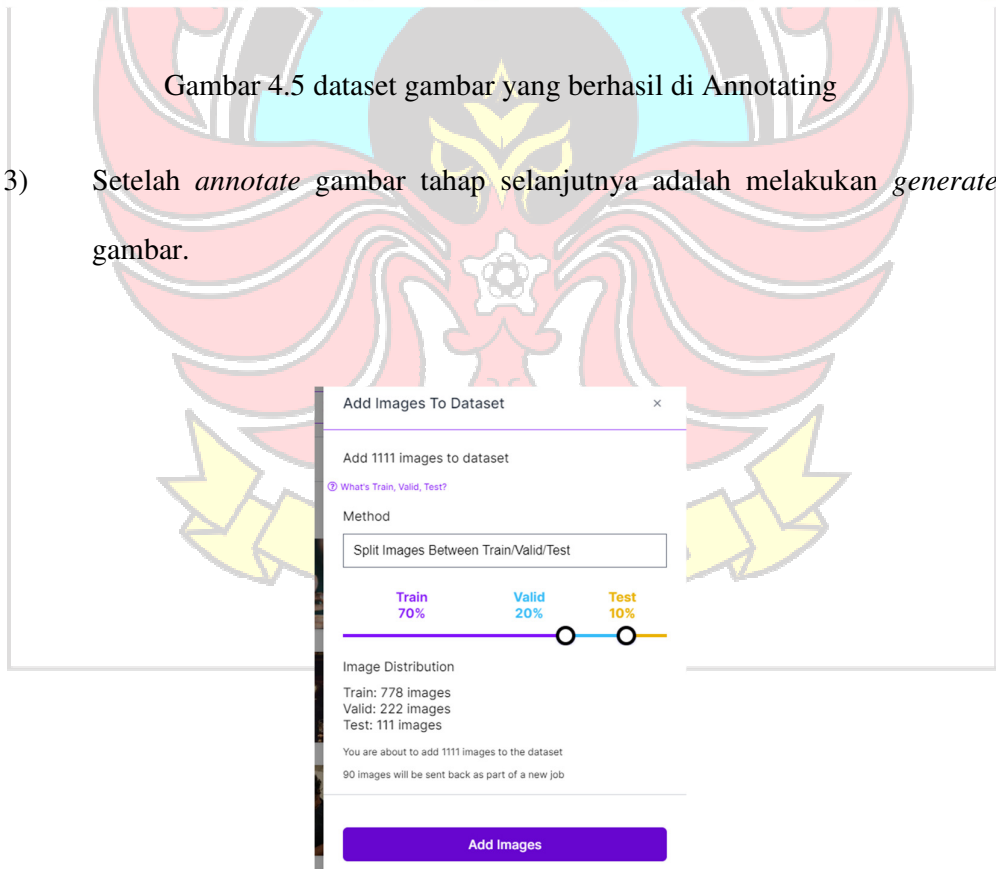
Pada Gambar 4.4 adalah gambar yang telah dilakukan *annotating* terhadap benda rokok setelah melakukan *annotating* pada gambar selanjutnya memberikan *class* pada *annotating* tersebut dengan *class* rokok.

Dapat dilihat pada Gambar 4.5 merupakan dataset gambar yang telah di *annotate* dikumpulkan ke halaman dashboard dimana gambar tersebut dilakukan *train*, *test*, dan *valid*. Untuk menghasilkan berapa persen setiap *mAP*, *Precision*, dan *Recal*



Gambar 4.5 dataset gambar yang berhasil di Annotating

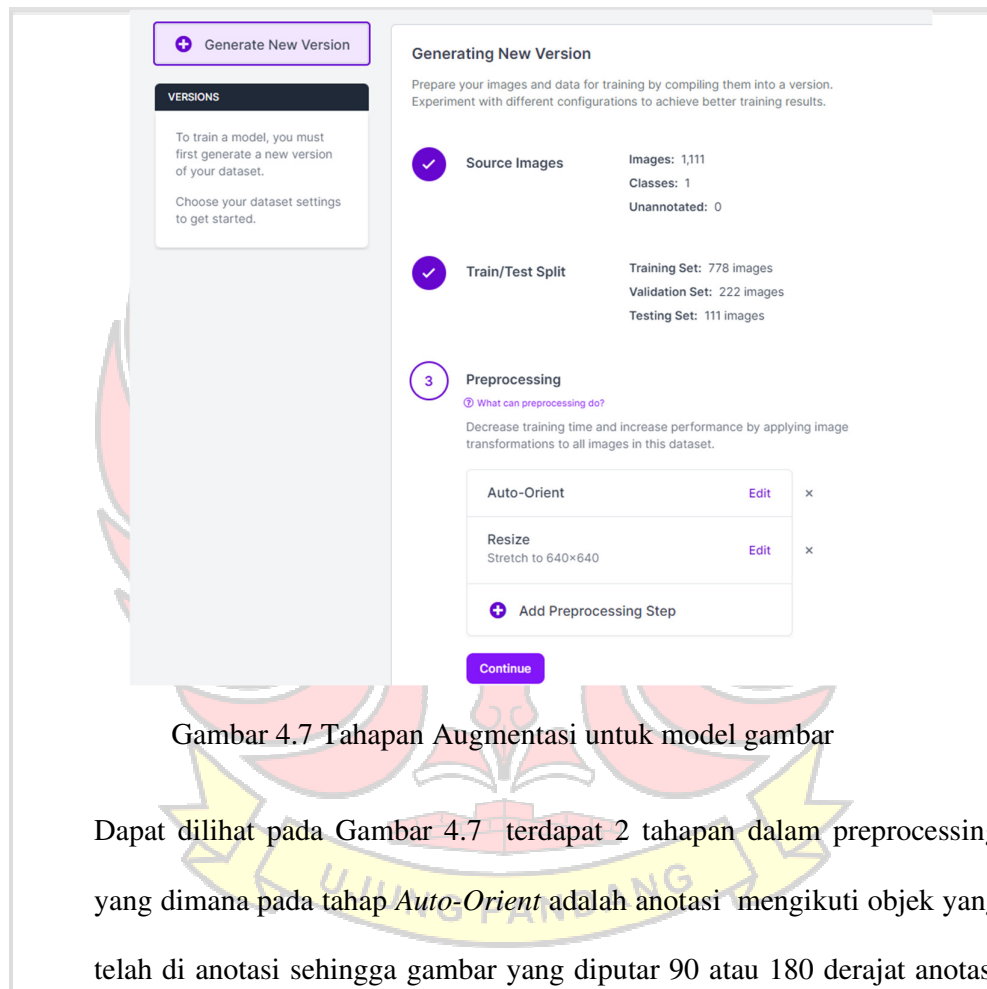
- 3) Setelah *annotate* gambar tahap selanjutnya adalah melakukan *generate* gambar.



Gambar 4.6 menambahkan gambar untuk dijadikan model gambar custome

Pada Gambar 4.6 terdapat 3 pembagian *Method* dimana pada *train* 70% yang dijadikan *training*, untuk *valid* 20% dari model gambar dan *Test* 10% dari model gambar dijadikan sebagian *Test*

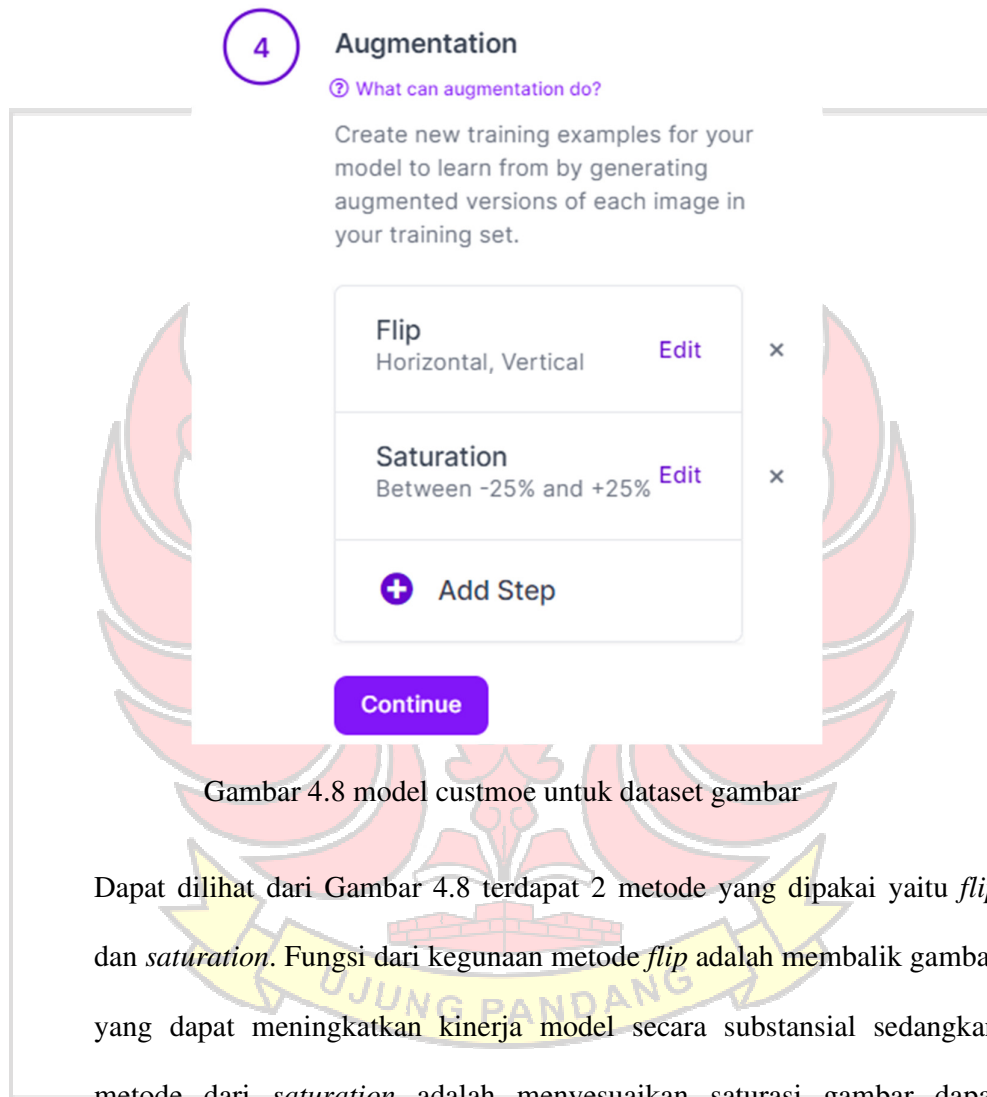
4) Pada tahap berikutnya melakukan *Preprocessing*



Gambar 4.7 Tahapan Augmentasi untuk model gambar

Dapat dilihat pada Gambar 4.7 terdapat 2 tahapan dalam preprocessing yang dimana pada tahap *Auto-Orient* adalah anotasi mengikuti objek yang telah di anotasi sehingga gambar yang diputar 90 atau 180 derajat anotasi mengikuti objek tersebut pada semua gambar yang telah di upload. Sedangkan tahap *resize* mengubah semua gambar ke dalam ukuran 640x640.

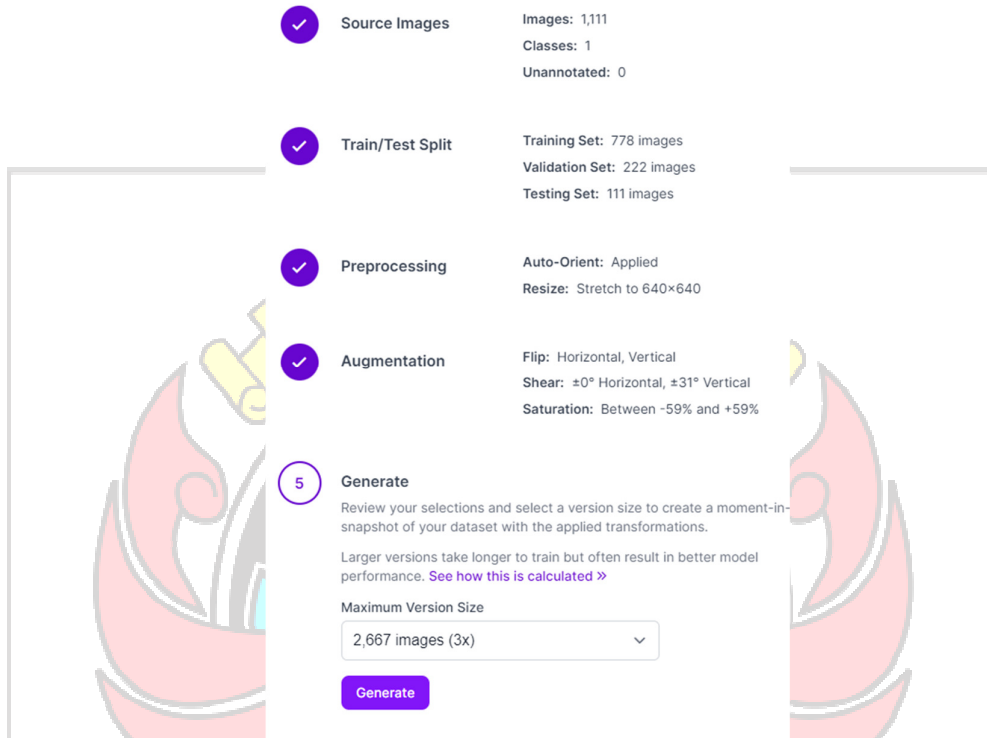
- 5) Tahap berikutnya adalah *augmentation* yaitu proses membantu meningkatkan kemampuan model Anda untuk menggeneralisasi sehingga bekerja lebih efektif pada gambar.



Gambar 4.8 model custmoe untuk dataset gambar

Dapat dilihat dari Gambar 4.8 terdapat 2 metode yang dipakai yaitu *flip* dan *saturation*. Fungsi dari kegunaan metode *flip* adalah membalik gambar yang dapat meningkatkan kinerja model secara substansial sedangkan metode dari *saturation* adalah menyesuaikan saturasi gambar dapat membantu model gambar lebih performa lebih baik ketika warna di ruangan berbeda (misalnya, jika keseimbangan putih berbeda diatur, pencahayaan berbeda berlaku, atau bahkan saat berkabut)

- 6) Langkah akhir melakukan generate dari semua model gambar dengan menggunakan metode yang sudah *custome*.



Gambar 4.9 Generate hasil model custome gambar

Pada tahapan generate ini memerlukan waktu yang lama karena setiap gambar yang di upload di custome sesuai dengan custome yang telah dipilih.

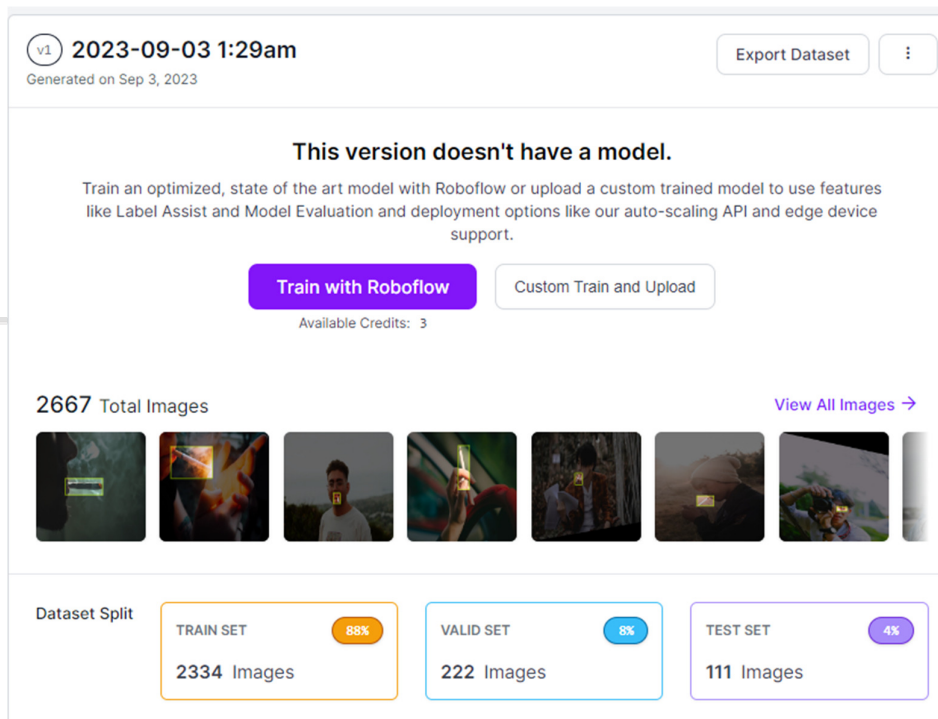
- 7) Setelah generate berhasil dapat diperhatikan pada Gambar 4.10 merupakan hasil dari dataset model gambar yang telah di custome yang menghasilkan 90.7% *mAP*, 92.9% *Precision*, 87.0% *Recall*.

```
tugas-akhir-kfgeo/1
90.7%    92.9%    87.0%
mAP      Precision Recall
```

Gambar 4.10 Hasil Generate model gambar custome

Dapat dilihat pada Gambar 4.10 *mAP* adalah adalah salah satu metrik yang digunakan untuk mengevaluasi dan membandingkan berbagai model deteksi objek. Semakin tinggi nilai *mAP*, semakin baik model dalam melakukan deteksi objek, *Precision* adalah mengukur sejauh mana model berhasil dalam mengidentifikasi objek dengan benar dari semua prediksi positif yang dibuat. Dalam konteks deteksi objek, ini berarti berapa banyak bounding box (kotak yang mengelilingi objek) yang benar-benar mengandung objek yang sesuai dengan yang diharapkan, dibagi dengan total bounding box yang dihasilkan oleh model, *recall* adalah *recall* mengukur sejauh mana model berhasil dalam menemukan semua objek yang sebenarnya ada. Ini berarti berapa banyak objek yang benar-benar terdeteksi dari semua objek yang seharusnya terdeteksi.

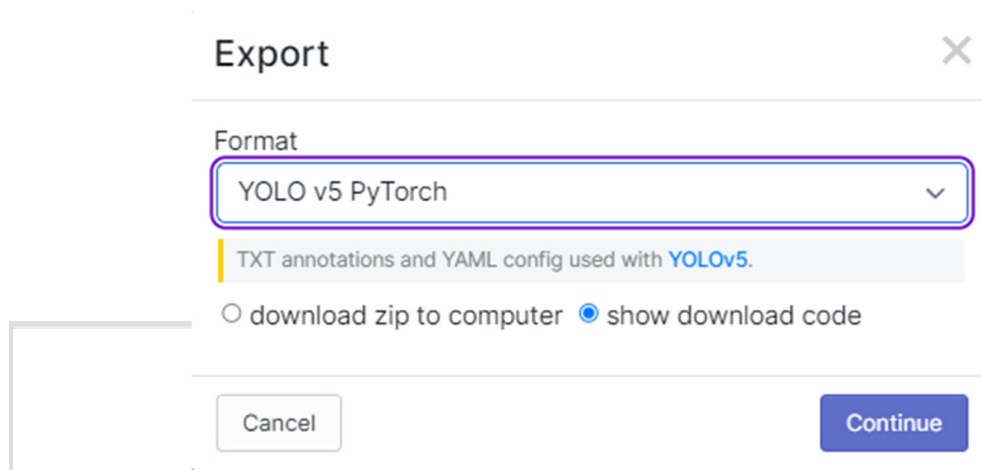
- 8) Selesai dari tahap generate maka terdapat hasilnya seperti pada Gambar 4.11 terdapat dataset *split* yang terdiri *train set* 2334 gambar, *valid set* 222 gambar, dan *test set* 111 gambar.



Gambar 4.11 Hasil Pembuatan model gambar custome

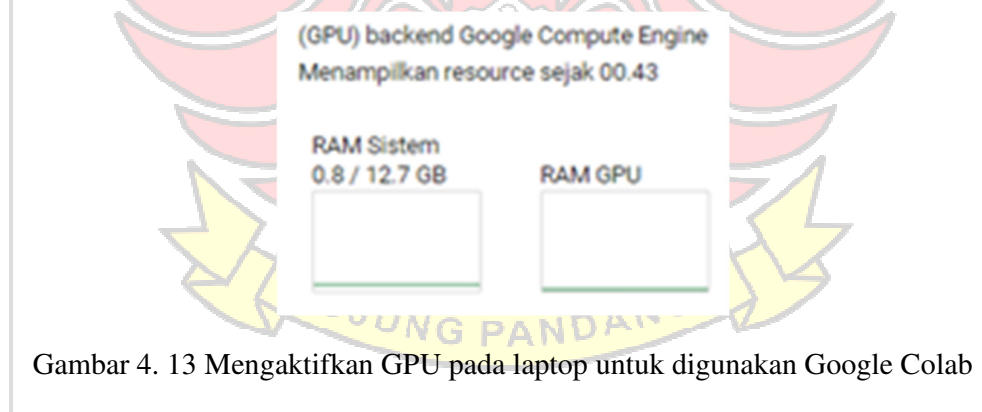
Pada Gambar 4.11 merupakan hasil gambar dataset yang telah di custome dengan beberapa metode sehingga dapat di *download* bisa digunakan untuk mendeploy dataset menjadi model Yolov5. Tetapi untuk mendownload deploy dataset tersebut harus membeli dari roboflow. Tetapi saya melatih dataset gambar yang telah di custome menggunakan google colab sehingga menghasilkan model Yolov5.

- 9) Dapat dilihat pada Gambar 4.12 tahap ini melakukan Export hasil pembuatan gambar yang telah di custome ke dalam versi YOLO v5 PyTorch agar dapat mendeploy ke dalam google colab.

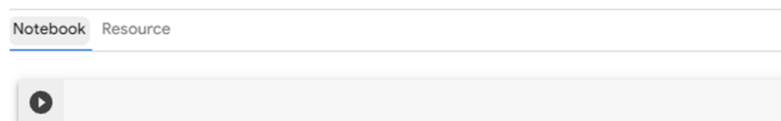


Gambar 4.12 Versi Export gambar model dataset

- 10) Langkah yang pertama adalah membuka website dari google colab dan membuat akun terlebih dahulu, selanjutnya membuat note baru dan melakukan konfigurasi untuk menggunakan GPU dalam melatih model gambar dari roboflow

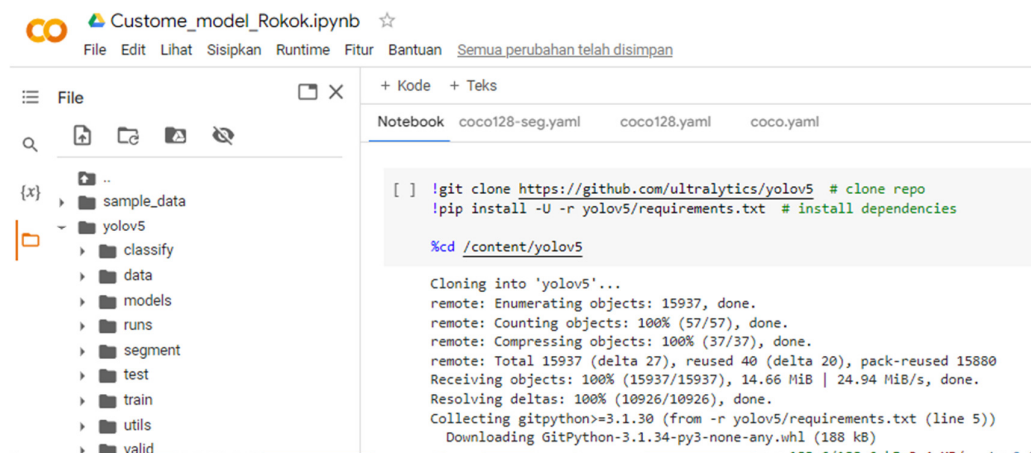


Gambar 4. 13 Mengaktifkan GPU pada laptop untuk digunakan Google Colab



Gambar 4.14 Tampilan note baru untuk memasukkan perintah

- 11) Pada tahap selanjutnya adalah melakukan beberapa perintah seperti pada Gambar 4.16 terdapat perintah yaitu `!git clone https://github.com/ultralytics/yolov5`, `!pip install -U -r yolov5/requirements.txt`, dan `%cd /content/yolov5`



```
Custom_model_Rokok.ipynb ☆
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan

+ Kode + Teks
Notebook coco128-seg.yaml coco128.yaml coco.yaml

[ ] !git clone https://github.com/ultralytics/yolov5 # clone repo
!pip install -U -r yolov5/requirements.txt # install dependencies

%cd /content/yolov5

Cloning into 'yolov5'...
remote: Enumerating objects: 15937, done.
remote: Counting objects: 100% (57/57), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 15937 (delta 27), reused 40 (delta 20), pack-reused 15880
Receiving objects: 100% (15937/15937), 14.66 MiB | 24.94 MiB/s, done.
Resolving deltas: 100% (10926/10926), done.
Collecting gitpython>=3.1.30 (from -r yolov5/requirements.txt (line 5))
  Downloading GitPython-3.1.34-py3-none-any.whl (188 kB)
```

Gambar 4.15 Berhasil melakukan clone dan Install Yolov5

Pada Gambar 4.16 terdapat perintah `!git clone https://github.com/ultralytics/yolov5` artinya perintah untuk mengklon (mendownload) repositori YOLOv5 dari GitHub ke dalam google colab. Setelah menjalankan perintah ini, terdapat file YOLOv5 di direktori tersebut. Sedangkan pada perintah `!pip install -U -r yolov5/requirements.txt` adalah perintah untuk menginstal semua dependensi yang diperlukan oleh YOLOv5, seperti pustaka Python seperti PyTorch, OpenCV, dan sebagainya. Perintah ini membantu memastikan bahwa semua komponen yang diperlukan untuk menjalankan YOLOv5 telah diinstal. Pada perintah `%cd /content/yolov5` adalah untuk mengubah direktori saat ini menjadi direktori yolov5

- 12) Pada tahapan selanjutnya memasukkan beberapa perintah seperti pada gambar 4.16 yaitu perintah `import torch` from IPython.display import Image # for displaying images `print('torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))`.



```
[3] import torch
from IPython.display import Image # for displaying images
print('torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))
torch 2.0.1+cu118 _CudaDeviceProperties(name='Tesla T4', major=7, minor=5, total_memory=15101MB, multi_processor_count=40)
```

Gambar 4.16 berhasil mengimpor torch

Pada gambar 4.16 perintah dari `import torch` from IPython.display import Image # for displaying images `print('torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))` yaitu pada “import torch” berfungsi mengimpor library PyTorch. PyTorch adalah salah satu framework yang populer digunakan untuk pengembangan model deep learning, termasuk model-model yang digunakan dalam tugas-tugas seperti pelatihan dan inferensi dengan model YOLOv5. Pada `from IPython.display import Image` adalah pernyataan impor yang mengimpor kelas Image dari modul IPython.display. Kelas ini digunakan untuk menampilkan gambar dalam lingkungan Jupyter Notebook atau IPython. Pada perintah `print('torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))`: adalah untuk pernyataan cetak yang mencetak informasi tentang versi PyTorch dan Status GPU.

- 13) Langkah berikutnya memasukkan perintah `!curl -L "https://app.roboflow.com/ds/DdMfBdFhIx?key=kziRZv0Muf"` > roboflow.zip; unzip roboflow.zip; rm roboflow.zip seperti pada gambar 4.17.

```
✓ 4d [4] !curl -L "https://app.roboflow.com/ds/DdMfBdFhIx?key=kziRZv0Muf" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip
extracting: valid/labels/pexels-cottonbro-studio-4786915_jpg.rf.d255ebfebcd6cb94bb9397f94c4d9cd0.txt
extracting: valid/labels/pexels-cottonbro-studio-5036267_jpg.rf.56840b2cb3387a1680140e8b4f193c8a.txt
extracting: valid/labels/pexels-cottonbro-studio-6037728_jpg.rf.2bd85ba23d9e589fd148c446b3b8d3eb.txt
extracting: valid/labels/pexels-cottonbro-studio-7319350_jpg.rf.26b845a13940675347a557e04b96686f.txt
extracting: valid/labels/pexels-dante-munoz-16346588_jpg.rf.a9f3931821bc4e8aab97bb265c20c82c.txt
extracting: valid/labels/pexels-dids-2678531_jpg.rf.b4cbf6a19678f079f8cb512e77add271.txt
extracting: valid/labels/pexels-dilara-irem-13363537_jpg.rf.b2f47fb41a8418f30942b59d78793466.txt
extracting: valid/labels/pexels-dilara-irem-14996717_jpg.rf.d4ccc0e65d2560677b0e7f83c74f986e.txt
extracting: valid/labels/pexels-dilara-irem-18096537_jpg.rf.dc1f23b07259eda9fbf95c4682afffb6.txt
extracting: valid/labels/pexels-ike-louie-natividad-3304346_jpg.rf.eea293b85dc4efa61ae09175d0e77b82.txt
extracting: valid/labels/pexels-irina-iriser-798124_jpg.rf.be39fa7754e2ab4c01f15ee313adf87d.txt
extracting: valid/labels/pexels-johanser-martinez-5372768_jpg.rf.2d1896f7c87dfcb7d045de468d5d0e3b.txt
extracting: valid/labels/pexels-khalid-garcia-1376088_jpg.rf.2a3c47c6373e26cfb518eacfee89f857.txt
extracting: valid/labels/pexels-levent-simsek-3885957_jpg.rf.940dcb1ea326c640e91b30670c0c763b.txt
extracting: valid/labels/pexels-lorenzo-cingue-4012266_jpg.rf.1947005d99d7141d4df318d093032596.txt
```

Gambar 4.17 mendownload model gambar dataset dari roboflow

Dapat kita pada Gambar 4.17 adalah perintah untuk melakukan mengunduh file dataset gambar dari roboflow. Setelah mengunduh maka akan terdapat file test,train, dan valid di dalam directory yolov5.

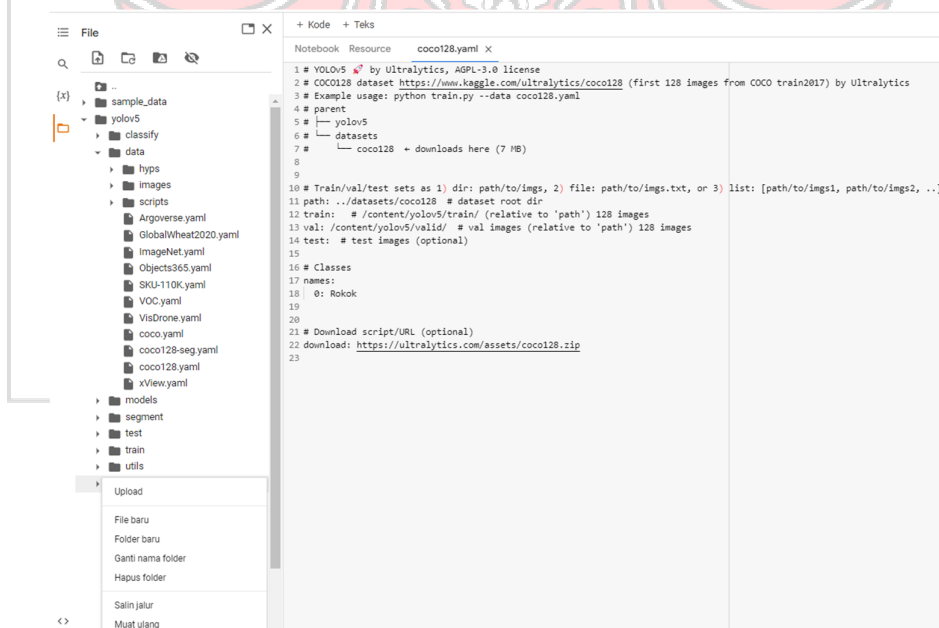
- 14) Tahapan selanjutnya masuk ke dalam folder coco128 untuk menghapus semua class yang di dalamnya dan menambahkan class name sesuai dengan name object dari roboflow yaitu rokok. Hal ini dikarenakan nama class tersebut adalah rokok pada roboflow sehingga pada google colab harus mengikuti nama class pada roboflow tersebut.



Gambar 4.18 Mengubah classes menjadi rokok

Pada Gambar 4.18 terdapat name Rokok hal ini dikarenakan name Rokok merupakan name object dari roboflow.

- 15) Langkah berikutnya melakukan konfigurasi pada train dan val pada coco128.yaml seperti pada Gambar 4.19



Gambar 4.19 konfigurasi train dan val dengan menggunakan folder roboflow

Pada Gambar 4.19 melakukan konfigurasi pada train dan val yang ada di dalam directory coco128.yaml dimana file direktori train dan val yang ada di dalam directory dalam yolov5 klik salin jalur selanjutnya kembali lagi ke coco128.yaml melakukan konfigurasi klik pada baris train dan val selanjutnya paste yang telah di salin jalur sebelumnya.

16) Tahapan berikutnya masuk pada halaman notebook kembali untuk melakukan deploy seperti pada Gambar 4.20.

```

python train.py --img 640 --batch 16 --epochs 100 --data coco128.yaml --weight yolov5s.pt --nosave --cache
89/99 4.27G 0.02578 0.008114 0 24 640: 100% 7/7 [00:01<00:00, 5.16it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:01<00:00, 3.92it/s]
      all 222 223 0.924 0.762 0.836 0.416

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
90/99 4.27G 0.02549 0.009328 0 26 640: 100% 7/7 [00:01<00:00, 4.16it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:02<00:00, 2.51it/s]
      all 222 223 0.917 0.771 0.84 0.42

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
91/99 4.27G 0.0267 0.008228 0 25 640: 100% 7/7 [00:01<00:00, 4.14it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:02<00:00, 3.03it/s]
      all 222 223 0.93 0.77 0.835 0.423

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
92/99 4.27G 0.02549 0.007873 0 23 640: 100% 7/7 [00:01<00:00, 5.20it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:01<00:00, 4.03it/s]
      all 222 223 0.919 0.76 0.825 0.424

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
93/99 4.27G 0.0255 0.008804 0 26 640: 100% 7/7 [00:01<00:00, 5.30it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:01<00:00, 3.89it/s]
      all 222 223 0.924 0.76 0.831 0.422

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
94/99 4.27G 0.02553 0.008309 0 23 640: 100% 7/7 [00:01<00:00, 4.96it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:01<00:00, 3.56it/s]
      all 222 223 0.902 0.78 0.83 0.429

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
95/99 4.27G 0.02487 0.007329 0 18 640: 100% 7/7 [00:01<00:00, 4.02it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:02<00:00, 2.40it/s]
      all 222 223 0.914 0.771 0.836 0.425

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
96/99 4.27G 0.02488 0.007931 0 27 640: 100% 7/7 [00:01<00:00, 4.31it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:01<00:00, 3.72it/s]
      all 222 223 0.903 0.767 0.838 0.429

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
97/99 4.27G 0.02421 0.006836 0 20 640: 100% 7/7 [00:01<00:00, 5.07it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:01<00:00, 4.00it/s]
      all 222 223 0.905 0.771 0.839 0.428

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
98/99 4.27G 0.0247 0.007374 0 26 640: 100% 7/7 [00:01<00:00, 5.14it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:01<00:00, 3.98it/s]
      all 222 223 0.914 0.78 0.841 0.426

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
99/99 4.27G 0.02236 0.007534 0 30 640: 100% 7/7 [00:01<00:00, 5.04it/s]
      Class Images Instances P R mAP50 mAP50-95: 100% 7/7 [00:02<00:00, 3.09it/s]
      all 222 223 0.909 0.776 0.841 0.427

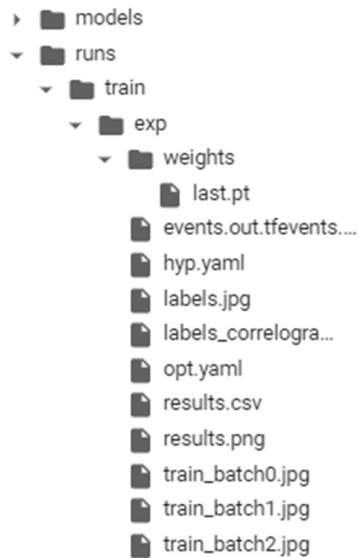
100 epochs completed in 0.109 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.4MB
Results saved to runs/train/exp

```

Gambar 4.20 melatih model gambar untuk versi YOLOv5

Pada Gambar 4.20 terdapat beberapa perintah `!python train.py --img 640 --batch 16 --epochs 100 --data custome_data.yaml --weight yolov5s.pt --nosave --cache`. Pada perintah `!python train.py` adalah untuk memanggil Bahasa python dan menjalankannya. `--img640` adalah untuk menetapkan gambar masukan menjadi 640x640 piksel. `--batch16` adalah model memproses 16 gambar sekaligus selama setiap iterasi pelatihan. `--epochs 100` adalah menetapkan jumlah epoch pelatihan menjadi 100. sebuah epoch adalah lintasan lengkap melalui keseluruhan dataset pelatihan. `--data custom_data.yaml` adalah berisi informasi tentang kumpulan data seperti jalur ke data pelatihan, validasi, dan label kelas. `--nosave` adalah mencegah pos pemeriksaan model disimpan selama pelatihan dan hanya menginginkan model akhir yang dilatih. Setelah menjalankan perintah tersebut terdapat file `last.pt` yang merupakan hasil dari deploy.

- 17) Pada tahapan akhir adalah melakukan download file `last.pt` seperti pada Gambar 4.21. file tersebut merupakan hasil model yang telah dilatih agar dapat digunakan pada `yolov5`. file tersebut merupakan file gambar rokok yang digunakan oleh algoritma `yolov5` untuk melakukan pendeteksian kegiatan merokok pegawai pada kawasan tanpa rokok.



Gambar 4.21 Last.pt hasil latih untuk YOLOv5

Pada Gambar 4.21 last.pt merupakan hasil deploy yang digunakan dalam mendeteksi object rokok pada sistem yoloV5.

18) Metode RTSP pada CCTV

Pada pengujian tersebut CCTV terhubung dengan server menggunakan metode RTSP. Dimana pada metode tersebut terdapat port RTSP pada CCTV yang digunakan untuk saling terhubung dengan server agar dapat menampilkan hasil visual secara real time.

IPv4 Address	<input type="text" value="192.168.1.64"/>
RTSP Port	<input type="text" value="554"/>
Stream Type	<input type="text" value="Main Stream(Normal)"/>
Device No.	<input type="text" value="1"/>

User List			Add
No.	User Name	Level	
1	admin	Administrator	

Gambar 4.22 informasi sistem CCTV

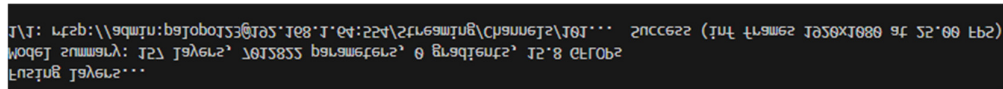
Pada Gambar 4.22 dapat dilihat pada IPv4 address merupakan IP kamera CCTV yaitu “192.168.1.64” yang berfungsi untuk memanggil URL CCTV pada sistem. Pada bagian RTSP Port terdapat “554” yang merupakan RTSP Port CCTV yang sangat berperan penting pada CCTV agar dapat RTSP ke dalam sistem. Di bagian Stream Type yaitu “Main Stream” yang merupakan Stream Type CCTV yang digunakan untuk stream ke sistem. Pada bagian Device no “1” merupakan Device no CCTV yang berfungsi untuk memanggil tampilan visual ke dalam sistem. Pada bagian User terdapat admin yang berfungsi untuk mengakses CCTV dan melakukan konfigurasi pada CCTV tersebut.

```
74 source= 'rtsp://admin:palopo123@192.168.1.64:554/Streaming/Channels/101'
```

Gambar 4.23 sistem CCTV memanggil ke server

Dapat dilihat pada Gambar 4.23 baris 74 codingan tersebut dimana terdapat “rtsp” yang berfungsi untuk menampilkan hasil visual pada CCTV secara real time. Pada bagian “admin” merupakan nama user yang digunakan untuk mengakses CCTV. Bagian “palopo123” merupakan password untuk mengakses CCTV tersebut. Di bagian “192.168.1.64” merupakan URL dari CCTV tersebut. Di bagian “Streaming” merupakan jenis type yang digunakan sesuai dengan

model type stream CCTV tersebut. Bagian channels berfungsi untuk memanggil channel dari CCTV tersebut. Pada bagian “101” 1 angka paling awal merupakan kode dari Stream Type sedangkan 01 merupakan channel dari CCTV.

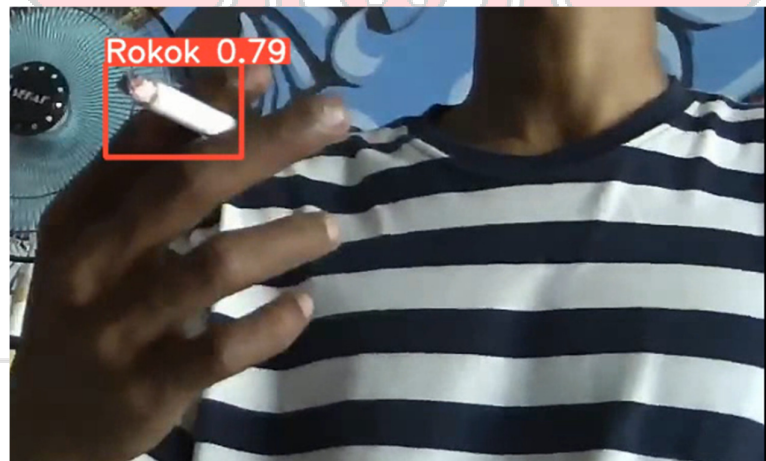


Gambar 4.24 Berhasil menampilkan CCTV ke Flask

Pada Gambar 4.24 dimana sistem dapat mengakses dan saling terhubung pada CCTV.

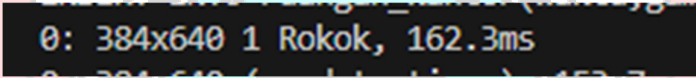
19) Deteksi Benda rokok

Pada bagian pengujian tersebut berkaitan dengan keberhasilan sistem Yolov5 yang telah dibuatkan model Rokok apakah Yolov5 dapat mendeteksi benda rokok tersebut dengan benar. Pada pengujian ini menggunakan CCTV yang telah terhubung dengan sistem secara real time.



Gambar 4.25 sistem Yolov5 berhasil mendeteksi objek Rokok

Dapat dilihat pada Gambar 4.25 merupakan gambar dari hasil visual CCTV yang diproses oleh OpenCV. hasil proses OpenCV bersifat video real stream tersebut diteruskan ke YOLOv5 untuk melakukan deteksi, apabila terdapat benda rokok maka YOLOv5 melakukan *bounding box* pada objek tersebut dan melakukan prediksi nilai pada objek sehingga apabila nilai prediksi di bawah 0.50 maka *bounding box* tersebut tidak tampil pada objek. Sedangkan nilai prediksi sama dengan 0.50 atau lebih dari 0.50 maka YOLOv5 menampilkan *bounding box* dan nilai di akurasi objek tersebut.



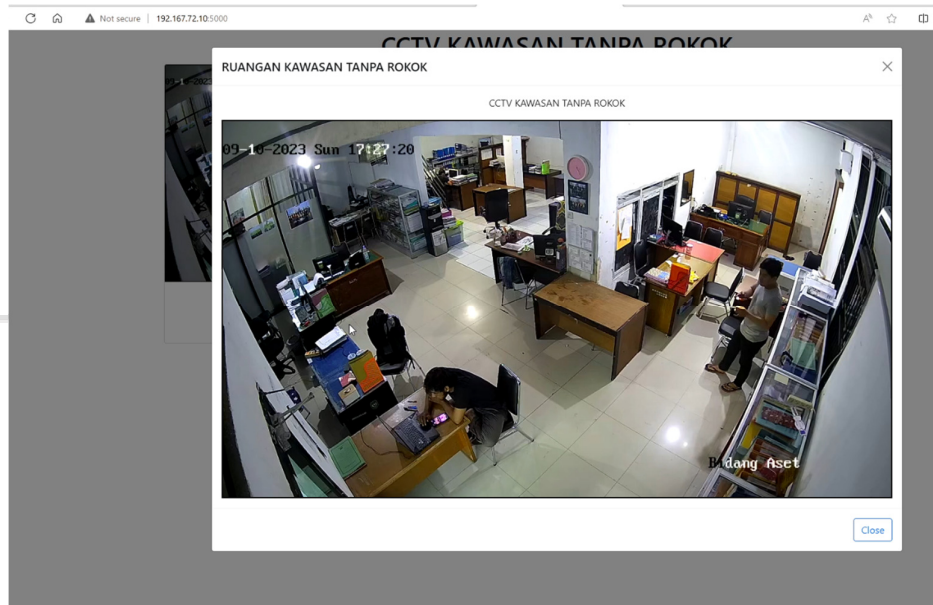
0: 384x640 1 Rokok, 162.3ms

Gambar 4.26 tampilan terminal apabila telah berhasil

apabila YOLOv5 mendeteksi benda rokok maka ada keterangan pada terminal seperti pada Gambar 4.26 selain itu YOLOv5 menyimpan foto tersebut.

20) Tampilan Flask

Pada tahapan ini flask berhasil menampilkan visual dari CCTV yang dimana OpenCV mengakses visual rekaman CCTV selanjutnya hasil visual tersebut diteruskan ke dalam YOLOv5 untuk melakukan pendeteksian setelah terdeteksi OpenCV akan mengambil hasil proses dari YOLOv5 dan selanjutnya OpenCV meneruskan hasil visual kepada Flask untuk ditampilkan. Dapat dilihat pada Gambar 4.27 merupakan tampilan flask secara real time.



Gambar 4.27 Tampilan Flask

```
* Serving Flask app 'detect'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
```

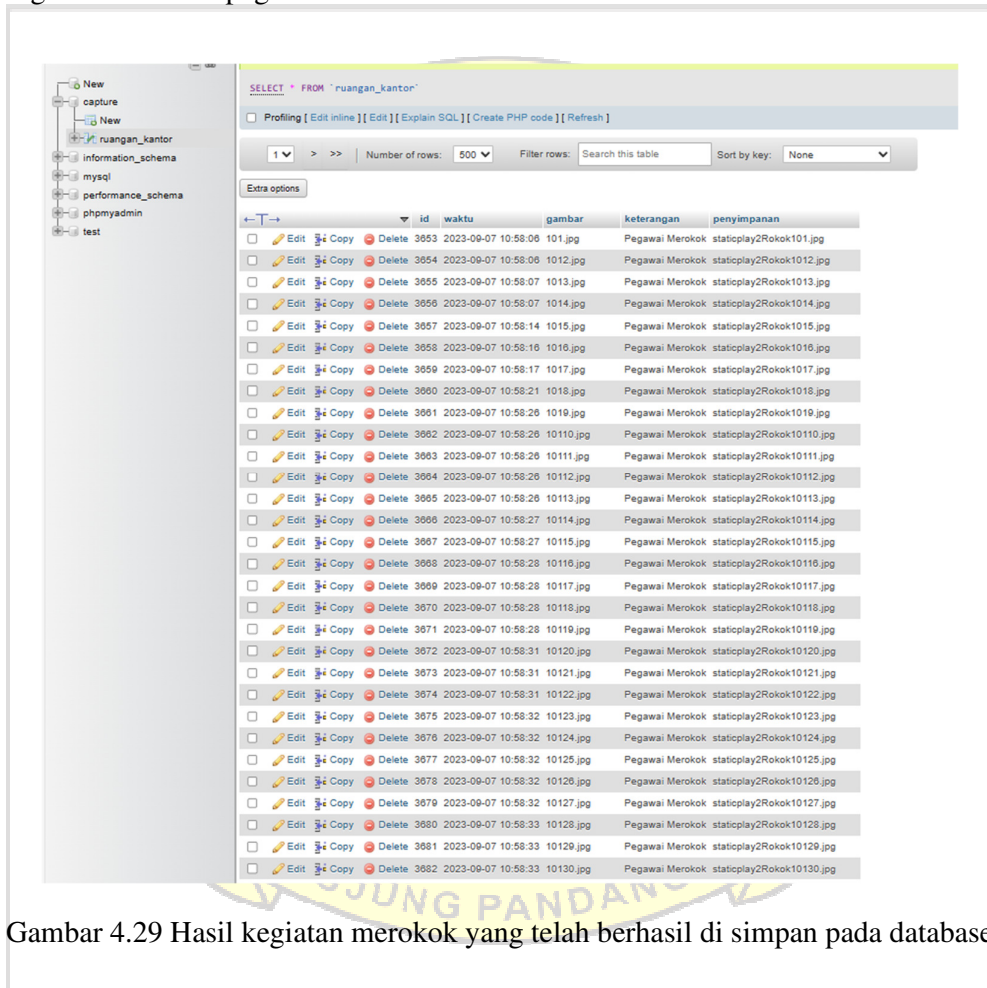
Gambar 4.28 URL Flask yang berhasil menampilkan visual CCTV

Apabila flask berhasil menampilkan hasil visual CCTV ke dalam halaman web maka terdapat beberapa keterangan seperti pada Gambar 4.28 terdapat URL halaman web.

21) Database MySQL

Pada pengujian database bertujuan agar sistem dapat mengirimkan bukti kegiatan merokok ke MySQL. Pada saat sistem Yolov5 mendeteksi benda objek rokok maka Yolov5 melakukan bounding box pada video tersebut. Selanjutnya Yolov5 menyimpan *screenshot* pada saat *bounding box* mendeteksi objek rokok. Setelah menyimpan *screenshot* kegiatan merokok dalam file gambar maka

sistem menyimpan bukti kegiatan merokok ke database MySQL. Pada database Capture terdapat beberapa table seperti id, waktu, gambar, keterangan, penyimpanan. Dapat lihat pada Gambar 4.29 terdapat kumpulan bukti kegiatan merokok pegawai. Pada Gambar 4.30 sistem telah berhasil menyimpan bukti kegiatan merokok pegawai dalam database.



Gambar 4.29 Hasil kegiatan merokok yang telah berhasil di simpan pada database

```
INSERT INTO ruangan_kantor(waktu,gambar,keterangan,penyimpanan) VALUES (now(), '08.jpg', 'Pegawai Merokok', 'staticplay4Rokok08.jpg')
```

Gambar 4.30 sistem berhasil mengirim bukti kegiatan merokok ke database

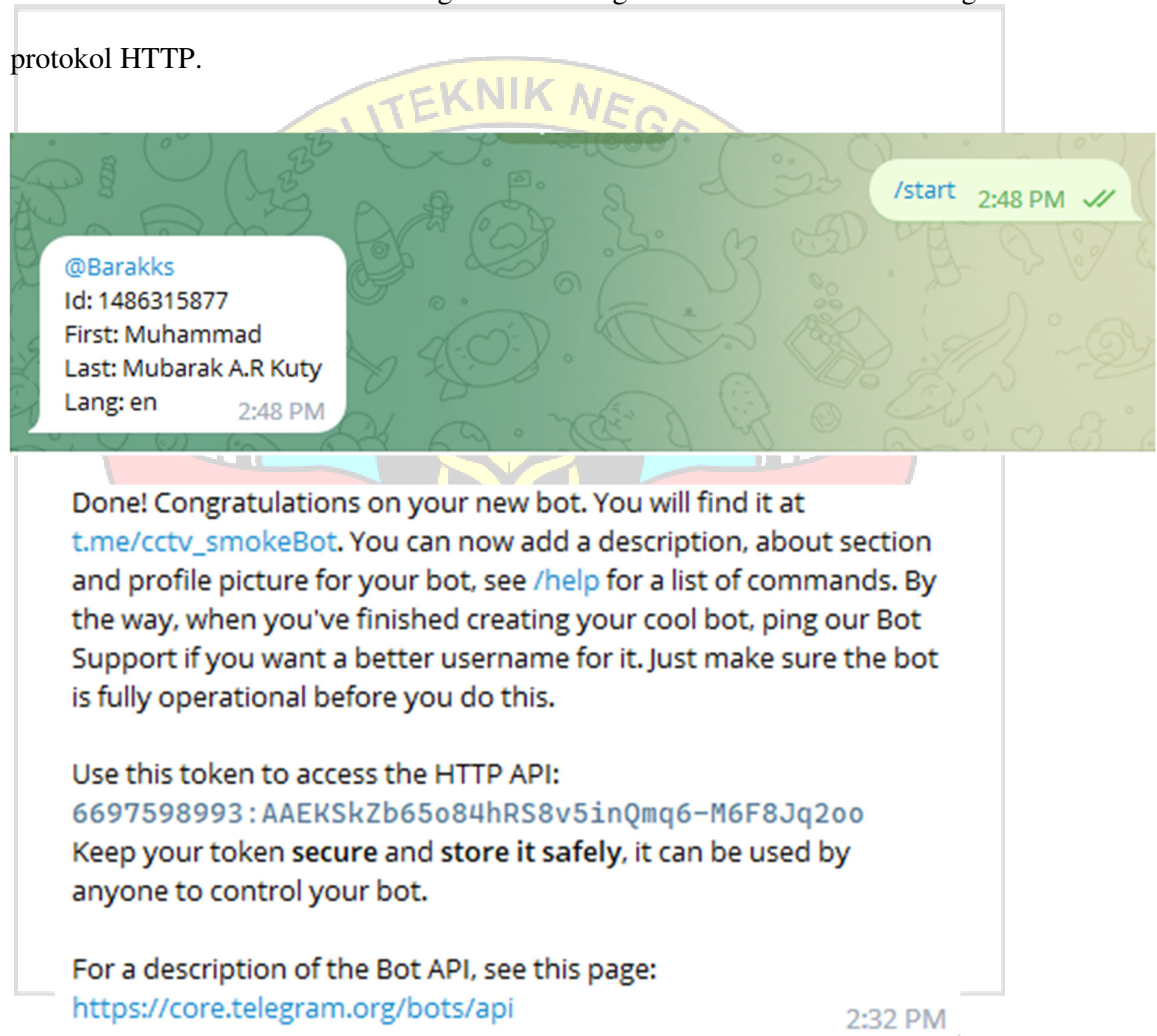
1) Notifikasi Telegram

Pada pengujian notifikasi telegram tahapan awal yang dilakukan adalah membuat bot telegram. Di dalam telegram terdapat akun botfather yang berfungsi untuk membuat user custome seperti pada Gambar 4.31 telah berhasil membuat custome bot.. Di tahapan akhir sistem mengirimkan bukti kegiatan merokok kepada bot yang telah di custome. Sistem mengirimkan bukti kegiatan merokok yang telah terdeteksi oleh Yolov5 selanjutnya bukti kegiatan merokok tersebut dikirim ke bot custome tersebut.



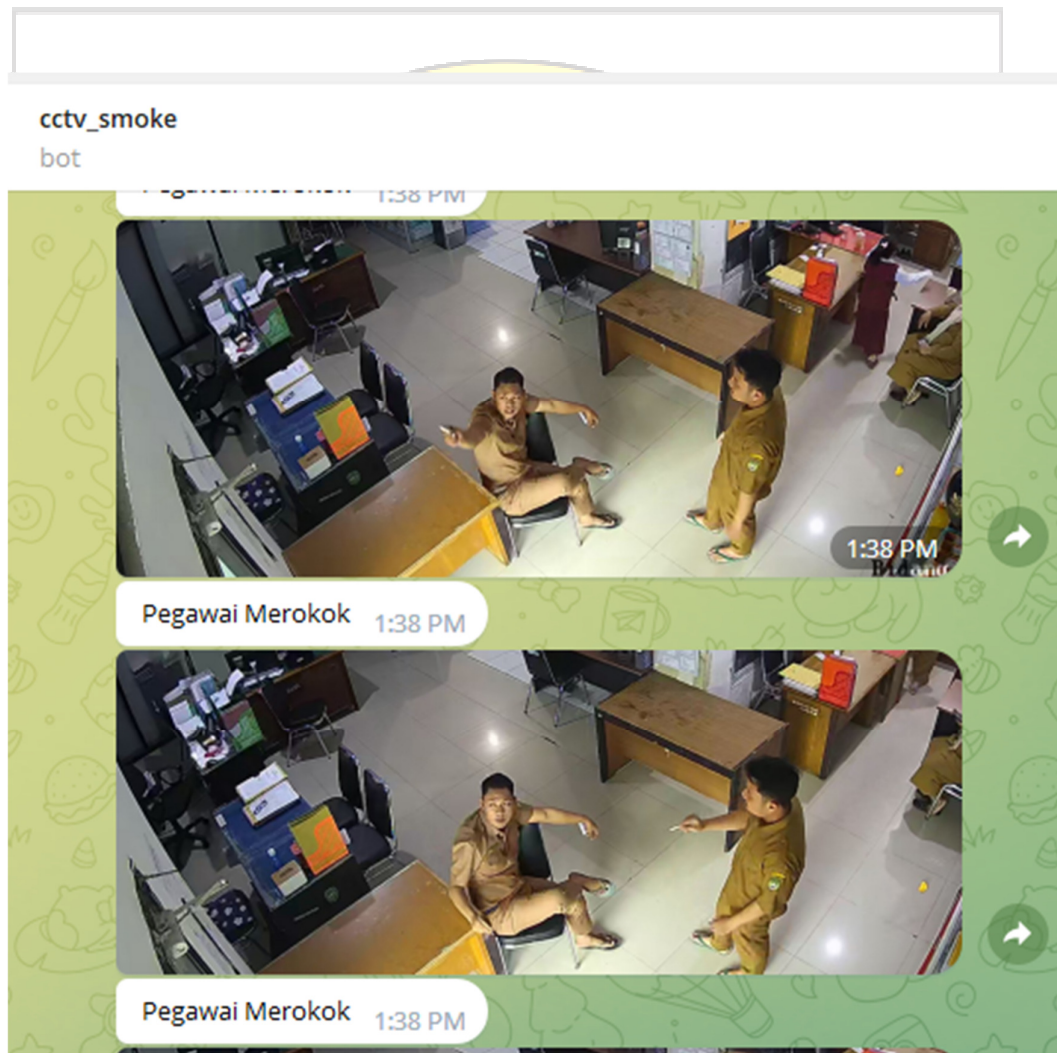
Gambar 4.31 botfathee

Tahapan berikutnya masukkan Id akun telegram dan token API yang di dapat dari botfather ke dalam sistem agar dapat mengakses HTTP API cat bot yang telah di custome dapat dilihat pada Gambar 4.32. fungsi dari id telegram agar sistem dapat mengirimkan bukti pelanggaran hanya pada 1 id saja sedangkan untuk HTTP API botfather berfungsi untuk mengambil data dari sistem dengan protokol HTTP.



Gambar 4.32 token HTTP API BotFather

Dapat dilihat pada Gambar 4.33 sistem telah mengirimkan bukti kegiatan merokok kepada bot telegram yang berisi file gambar pegawai melakukan kegiatan merokok keberhasilan sistem ini dikarenakan id telegram dan HTTP API bot father telah benar sehingga sistem tidak hanya mengirimkan 1 atau 2 bukti pelanggaran saja tetapi sistem akan mengirimkan semua pelanggaran pegawai yang melakukan kegiatan merokok di dalam kawasan tanpa rokok.



Gambar 4.33 Notifikasi kegiatan merokok ke telegram

4.3 Pengujian Fungsional

Pengujian Fungsional dilakukan untuk menguji sistem yang telah dibangun adapun metode yang digunakan *blackbox testing*. Pengujian ini dilakukan beberapa tahap proses pengujian berdasarkan skenario yang telah dibuat dapat dilihat pada Gambar 3.2 di Halaman 22. Adapun hasil yang diperoleh pada table

4.1 sebagai berikut

Table 4. 1 Hasil *Blackbox Testing*

Skenario Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil pengujian
Pembuatan Model	Melakukan custome gambar data set pada roboflow dan melakukan model gambar untuk model Yolov5 pada google colab	Model gambar Yolov5 dapat digunakan pada sistem	Berhasil
Terhubung dengan CCTV	Sistem dapat terhubung dengan CCTV melalui RTSP	Sistem dapat menampilkan gambar yang dihasilkan oleh CCTV secara Real Time	Berhasil
Mendeteksi rokok	CCTV menampilkan objek Rokok, telapak tangan dengan jari memegang rokok, dan punggung tangan dengan jari memegang rokok	Sistem dapat mendeteksi objek Rokok, telapak tangan dengan jari memegang rokok, dan punggung tangan dengan jari memegang rokok	Berhasil
Menyimpan bukti kegiatan merokok ke	CCTV menampilkan kegiatan	Sistem dapat menyimpan bukti kegiatan	Berhasil

Skenario Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil pengujian
database	pegawai merokok di kawasan tanpa rokok	merokok ke database	
Tampilan CCTV dapat ditampilkan di flask	CCTV menampilkan kegiatan di dalam ruangan kawasan tanpa rokok	Flask dapat menampilkan hasil rekaman CCTV	Berhasil
Mengirim notifikasi kegiatan merokok melalui telegram	CCTV menampilkan kegiatan merokok pegawai di kawasan tanpa rokok	Sistem mengirimkan bukti kegiatan merokok di notifikasi telegram	Berhasil

4.4 Pengujian Skenario

Pada tahap pengujian Skenario dilakukan beberapa pengujian dengan melibatkan beberapa dimensi seperti dimensi sudut, dimensi pencahayaan, dan dimensi jarak.

4.4.1 Pengujian Dimensi

Pada pengujian Dimensi tersebut melibatkan ketinggian CCTV 3 meter dan sudut pandang maksimal 110 derajat. Untuk jarak objek yang dideteksi mulai dari 1 meter, 2 meter, 3 meter, dan 4 meter. Sedangkan sudut objek terhadap CCTV mulai dari sudut 0 derajat, 30 derajat, dan 45 derajat, 60 derajat, dan 90 derajat Adapun untuk pencahayaan di dalam ruangan sebesar 37 LUX.

A. Pengujian jarak

Pada pengujian pertama adalah jarak. Dimana penelitian ini melakukan pendeteksian kegiatan merokok berdasarkan jarak objek merokok terhadap CCTV yang telah ditentukan. Ada 4 kategori jarak yang telah dilakukan untuk mendeteksi kegiatan merokok yaitu jarak 1 meter, 2 meter, 3 meter, dan 4 meter.

Terdapat objek merokok berjarak 1 meter terhadap CCTV dengan sudut 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat terdeteksi oleh CCTV sedang melakukan kegiatan merokok. Terdapat objek merokok berjarak 2 meter terhadap CCTV dengan sudut 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat terdeteksi oleh CCTV sedang melakukan kegiatan merokok. Terdapat objek merokok berjarak 3 meter terhadap CCTV dengan sudut 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat berhasil terdeteksi oleh CCTV sedang melakukan kegiatan merokok. Terdapat objek merokok berjarak 4 meter terhadap CCTV dengan sudut 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat tidak terdeteksi oleh CCTV sedang melakukan kegiatan merokok. Dapat dilihat pada Table 4.2 deteksi objek berdasarkan jarak

Table 4. 2 deteksi objek berdasarkan jarak

Jarak	Sudut	keterangan
1 meter	0 derajat	Berhasil
	30 derajat	Berhasil
	45 derajat	Berhasil
	60 derajat	Berhasil
	90 derajat	Berhasil
2 meter	0 derajat	Berhasil
	30 derajat	Berhasil
	45 derajat	Berhasil
	60 derajat	Berhasil
	90 derajat	Berhasil

Jarak	Sudut	keterangan
3 meter	0 derajat	Berhasil
	30 derajat	Berhasil
	45 derajat	Berhasil
	60 derajat	Berhasil
	90 derajat	Berhasil
4 meter	0 derajat	tidak berhasil
	30 derajat	tidak berhasil
	45 derajat	tidak berhasil
	60 derajat	tidak berhasil
	90 derajat	tidak berhasil

B. Pengujian sudut

Pada pengujian kedua adalah sudut pandang objek terhadap CCTV. Pada penelitian ini melakukan pendeteksian kegiatan merokok berdasarkan sudut pandang objek merokok terhadap CCTV. Ada 5 sudut yang telah dilakukan untuk mendeteksi kegiatan merokok yaitu sudut 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat. Pada sudut pandang objek merokok sebesar 0 derajat dan jarak objek pada CCTV sejauh 1 meter telah berhasil mendeteksi objek kegiatan merokok. Pada sudut pandang objek merokok sebesar 0 derajat dan jarak objek kegiatan merokok pada CCTV sejauh 2 meter telah berhasil mendeteksi objek kegiatan merokok. Pada sudut pandang objek merokok sebesar 0 derajat dan jarak objek merokok pada CCTV sejauh 3 meter berhasil mendeteksi objek kegiatan merokok. Pada sudut pandang objek merokok sebesar 0 derajat dan jarak CCTV pada objek sejauh 4 meter tidak berhasil mendeteksi objek kegiatan merokok. Untuk sudut pandang 30 derajat. Pada sudut 45 derajat, 60 derajat, dan 90 derajat dapat dilihat pada Table 4.3

Table 4. 3 Deteksi objek berdasarkan sudut

Sudut	Jarak	Keterangan
0 derajat	1 meter	berhasil
	2 meter	berhasil
	3 meter	berhasil
	4 meter	tidak berhasil
30 derajat	1 meter	berhasil
	2 meter	berhasil
	3 meter	berhasil
	4 meter	tidak berhasil
45 derajat	1 meter	berhasil
	2 meter	berhasil
	3 meter	berhasil
	4 meter	tidak berhasil
60 derajat	1 meter	berhasil
	2 meter	berhasil
	3 meter	berhasil
	4 meter	tidak berhasil
90 derajat	1 meter	berhasil
	2 meter	berhasil
	3 meter	berhasil
	4 meter	tidak berhasil

C. Pengujian intensitas cahaya

Pada pengujian ketiga adalah intensitas cahaya pada ruangan. Pada penelitian ini melakukan pendeteksian objek kegiatan merokok berdasarkan intensitas cahaya pada ruangan sebesar 37 LUX dengan jarak CCTV terhadap objek mulai dari 1 meter, 2 meter, 3 meter, dan 4 meter dan sudut pandang CCTV terhadap objek sebesar 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat.

Hasil penelitian dapat dilihat pada Table 4.4 sebagai berikut.

Table 4. 4 Deteksi Objek Intensitas cahaya

Intensitas cahaya	Jarak	Sudut	Keterangan
37 LUX	1 meter	0 derajat	Berhasil
		30 derajat	Berhasil
		45 derajat	Berhasil
		60 derajat	Berhasil
		90 derajat	Berhasil
37 LUX	2 meter	0 derajat	Berhasil
		30 derajat	Berhasil
		45 derajat	Berhasil
		60 derajat	Berhasil
		90 derajat	Berhasil
37 LUX	3 meter	0 derajat	Berhasil
		30 derajat	Berhasil
		45 derajat	Berhasil
		60 derajat	Berhasil
		90 derajat	Berhasil
37 LUX	4 meter	0 derajat	Tidak Berhasil
		30 derajat	Tidak Berhasil
		45 derajat	Tidak Berhasil
		60 derajat	Tidak Berhasil
		90 derajat	Tidak Berhasil

D. Kesimpulan Pengujian

Pada kesimpulan pengujian berdasarkan jarak, sudut, dan intensitas cahaya bahwa dengan intensitas cahaya di dalam ruangan sebesar 37 LUX dapat mendeteksi kegiatan merokok dengan jarak objek terhadap CCTV yang melakukan kegiatan merokok sejauh 1 meter, 2 meter, 3 meter, dan 4 meter serta sudut pandang objek terhadap CCTV melakukan kegiatan merokok sebesar 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat dapat mendeteksi kegiatan merokok tersebut. Sedangkan intensitas cahaya sebesar 37 LUX tidak dapat mendeteksi kegiatan merokok dengan jarak objek terhadap CCTV yang melakukan kegiatan merokok sejauh 4 meter serta sudut pandang CCTV terhadap

objek yang melakukan kegiatan merokok sebesar 0 derajat, 30 derajat, 45 derajat, 60 derajat, dan 90 derajat tidak dapat mendeteksi kegiatan merokok. Berdasarkan hasil pengujian maka diperoleh hasil akurasi sebagai berikut.

$$Akurasi = \frac{\text{berhasil dideteksi} - \text{tidak terdeteksi}}{\text{berhasil dideteksi}} \times 100\%$$


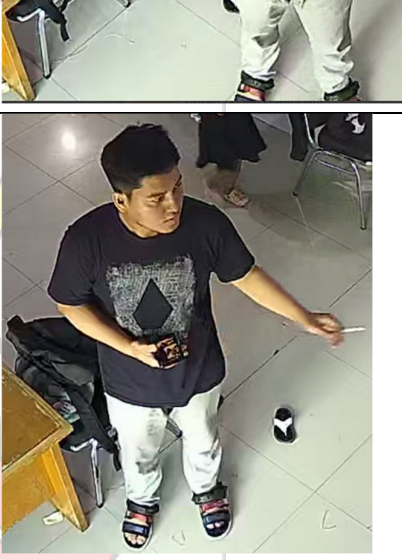

$$Akurasi = \frac{150 - 50}{150} \times 100\%$$



$$Akurasi = 66,6 \%$$




Jadi akurasi yang diperoleh sistem untuk mendeteksi kegiatan merokok adalah 66,6%



Table 4. 5 Hasil Skenario



Intensitas Cahaya	Jarak	Sudut	Keterangan	Gambar
37 LUX	1 meter	0 derajat	Berhasil	

Intensitas Cahaya	Jarak	Sudut	Keterangan	Gambar
		30 derajat	Berhasil	
		45 derajat	Berhasil	
		60 derajat	Berhasil	

Intensitas Cahaya	Jarak	Sudut	Keterangan	Gambar
		90 derajat	Berhasil	
37 LUX	2 meter	0 derajat	Berhasil	
		30 derajat	Berhasil	

Intensitas Cahaya	Jarak	Sudut	Keterangan	Gambar
		45 derajat	Berhasil	
		60 derajat	Berhasil	
		90 derajat	Berhasil	

Intensitas Cahaya	Jarak	Sudut	Keterangan	Gambar
37 LUX				
	3 meter	0 derajat	Berhasil	

Intensitas Cahaya	Jarak	Sudut	Keterangan	Gambar
		45 derajat	Berhasil	
		60 derajat	Berhasil	

Intensitas Cahaya	Jarak	Sudut	Keterangan	Gambar
		90 derajat	Berhasil	
37 LUX	4 meter	0 derajat	Tidak Berhasil	-
		30 derajat	Tidak Berhasil	-
		45 derajat	Tidak Berhasil	-
		60 derajat	Tidak Berhasil	-
		90 derajat	Tidak Berhasil	-

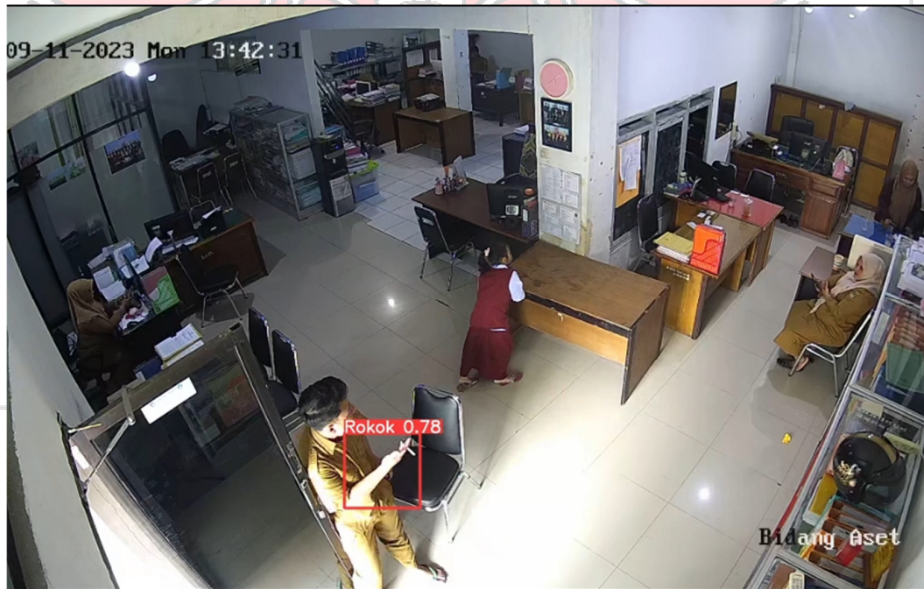
4.5 Hasil Pengujian

Pada hasil pengujian dilakukan pengujian sistem yang telah dibangun. Pada pengujian tersebut berada di lokasi kantor BPKAD kota palopo tepatnya di ruangan aset. Dalam pengujian sistem memantau pegawai dalam ruangan aset. Apabila pegawai terbukti melakukan kegiatan merokok, sistem melakukan proses yang telah dibangun.

Pada Gambar 4.34, Gambar 4.35, dan Gambar 4.36 merupakan hasil rekaman CCTV yang ditampilkan oleh halaman flask yang mendeteksi pegawai melakukan kegiatan merokok di kawasan tanpa rokok sehingga sistem melakukan bounding box pada objek serta melakukan *screenshot* pada objek menghasilkan file gambar untuk menyimpan ke database dan mengirimkan bukti ke telegram dalam bentuk file gambar.



Gambar 4.34 Tampilan halaman web

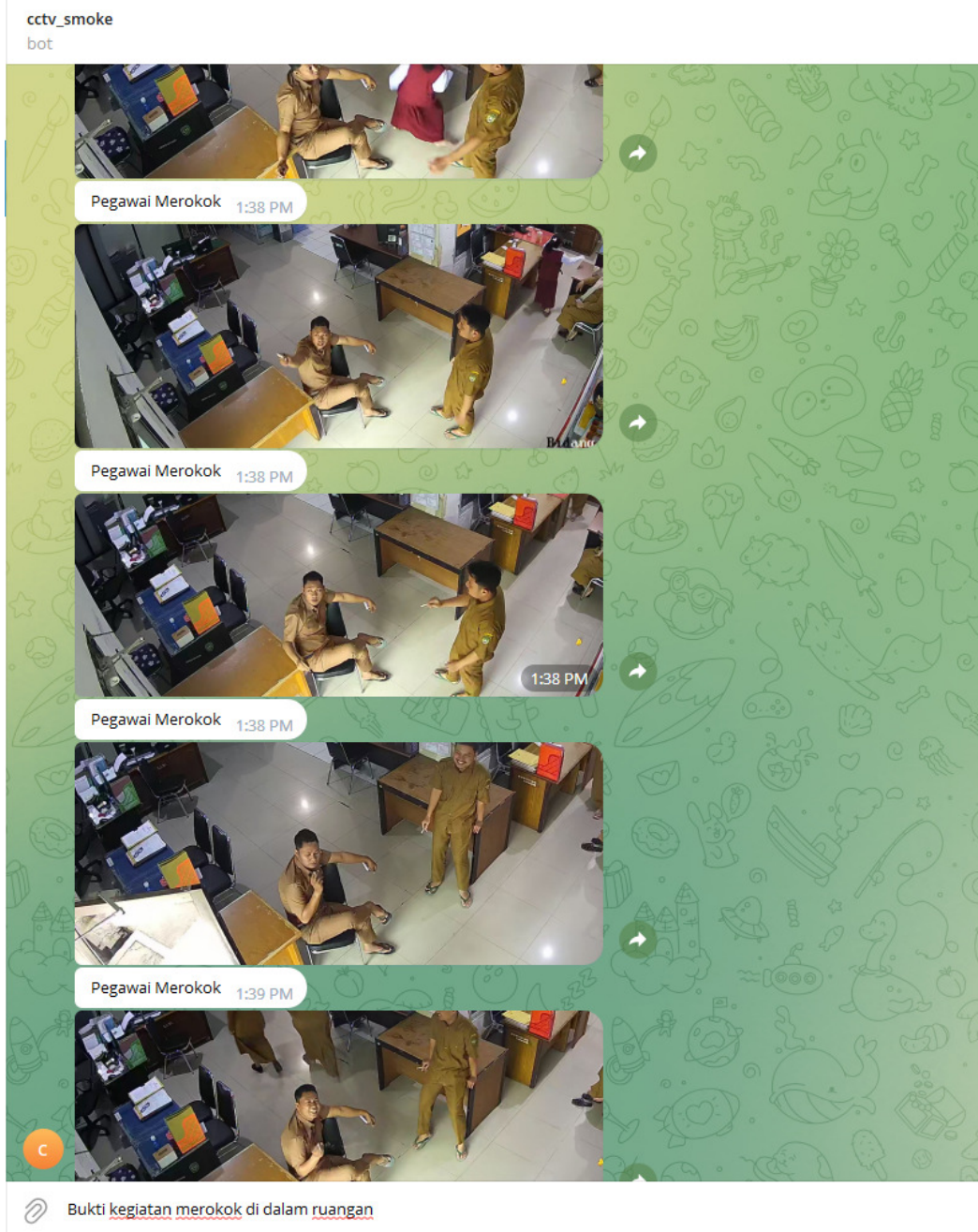


Gambar 4.35 Tampilan halaman web

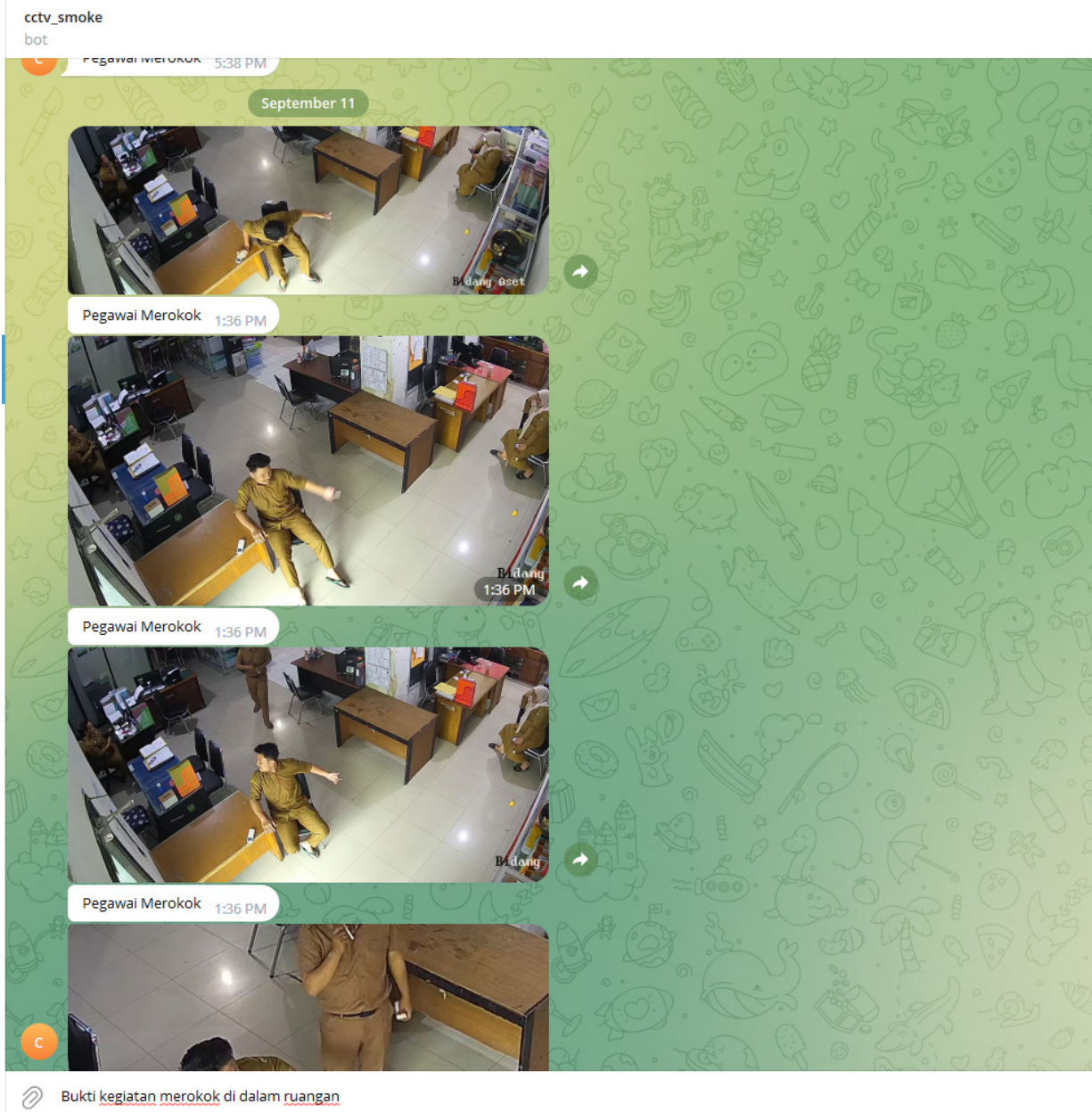


Gambar 4.36 Tampilan halaman web

Pada Gambar 4.37, Gambar 4.38, dan Gambar 4.39 merupakan notifikasi telegram bot. Pada saat Yolov5 mendeteksi adanya kegiatan merokok pada ruangan bidang aset maka Yolov5 melakukan *bounding box* selanjutnya dilakukan *screenshot* pada objek yang telah di *bounding box*. Hasil dari file gambar tersebut dikirim ke bot telegram untuk kegiatan merokok.

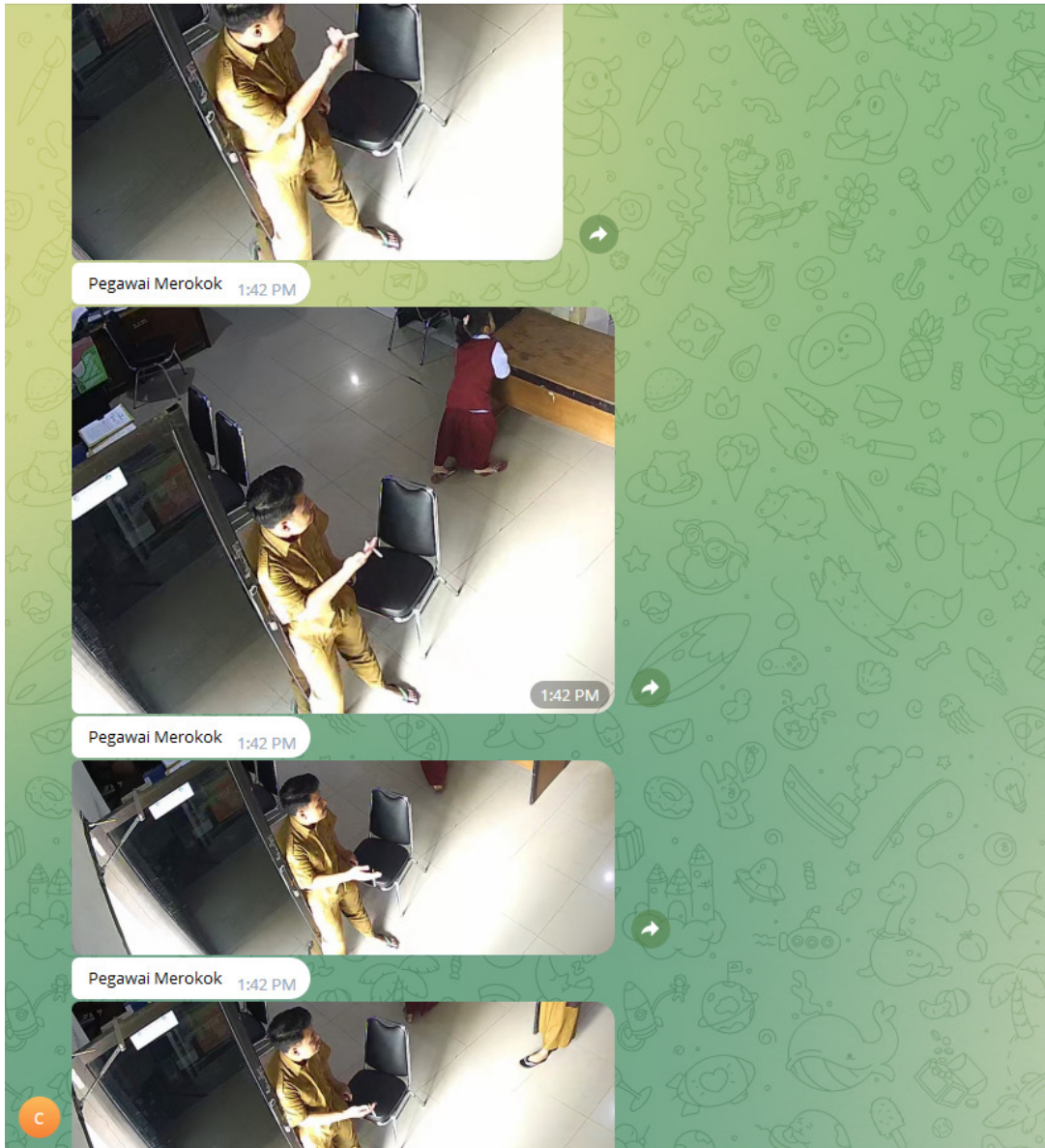


Gambar 4.37 Notifikasi bot telegram



Gambar 4.38 Notifikasi bot telegram

cctv_smoke
bot



Bukti kegiatan merokok di dalam ruangan

Gambar 4.39 Notifikasi bot telegram

Pada Gambar 4.40 database menyimpan bukti kegiatan merokok pegawai. Pada saat Yolov5 mendeteksi adanya kegiatan merokok pada ruangan bidang aset maka Yolov5 melakukan bounding box sehingga dilakukan *screenshot* pada objek yang telah dilakukan *bounding box*. Hasil dari file gambar tersebut disimpan pada database.

	id	waktu	gambar	keterangan	penyimpanan
<input type="checkbox"/>	4951	2023-09-11 13:36:04	101.jpg	Pegawai Merokok	staticplayRokok101.jpg
<input type="checkbox"/>	4952	2023-09-11 13:36:22	1012.jpg	Pegawai Merokok	staticplayRokok1012.jpg
<input type="checkbox"/>	4953	2023-09-11 13:36:27	1013.jpg	Pegawai Merokok	staticplayRokok1013.jpg
<input type="checkbox"/>	4954	2023-09-11 13:36:30	1014.jpg	Pegawai Merokok	staticplayRokok1014.jpg
<input type="checkbox"/>	4955	2023-09-11 13:36:55	1015.jpg	Pegawai Merokok	staticplayRokok1015.jpg
<input type="checkbox"/>	4956	2023-09-11 13:37:02	1016.jpg	Pegawai Merokok	staticplayRokok1016.jpg
<input type="checkbox"/>	4957	2023-09-11 13:37:09	1017.jpg	Pegawai Merokok	staticplayRokok1017.jpg
<input type="checkbox"/>	4958	2023-09-11 13:38:00	1018.jpg	Pegawai Merokok	staticplayRokok1018.jpg
<input type="checkbox"/>	4959	2023-09-11 13:38:04	1019.jpg	Pegawai Merokok	staticplayRokok1019.jpg
<input type="checkbox"/>	4960	2023-09-11 13:38:09	10110.jpg	Pegawai Merokok	staticplayRokok10110.jpg
<input type="checkbox"/>	4961	2023-09-11 13:38:12	10111.jpg	Pegawai Merokok	staticplayRokok10111.jpg
<input type="checkbox"/>	4962	2023-09-11 13:38:41	10112.jpg	Pegawai Merokok	staticplayRokok10112.jpg
<input type="checkbox"/>	4963	2023-09-11 13:38:49	10113.jpg	Pegawai Merokok	staticplayRokok10113.jpg
<input type="checkbox"/>	4964	2023-09-11 13:38:53	10114.jpg	Pegawai Merokok	staticplayRokok10114.jpg
<input type="checkbox"/>	4965	2023-09-11 13:39:18	10115.jpg	Pegawai Merokok	staticplayRokok10115.jpg
<input type="checkbox"/>	4966	2023-09-11 13:39:28	10116.jpg	Pegawai Merokok	staticplayRokok10116.jpg
<input type="checkbox"/>	4967	2023-09-11 13:41:09	10117.jpg	Pegawai Merokok	staticplayRokok10117.jpg
<input type="checkbox"/>	4968	2023-09-11 13:41:23	10118.jpg	Pegawai Merokok	staticplayRokok10118.jpg
<input type="checkbox"/>	4969	2023-09-11 13:41:43	10119.jpg	Pegawai Merokok	staticplayRokok10119.jpg
<input type="checkbox"/>	4970	2023-09-11 13:41:46	10120.jpg	Pegawai Merokok	staticplayRokok10120.jpg
<input type="checkbox"/>	4971	2023-09-11 13:42:03	10121.jpg	Pegawai Merokok	staticplayRokok10121.jpg
<input type="checkbox"/>	4972	2023-09-11 13:42:08	10122.jpg	Pegawai Merokok	staticplayRokok10122.jpg
<input type="checkbox"/>	4973	2023-09-11 13:42:27	10123.jpg	Pegawai Merokok	staticplayRokok10123.jpg
<input type="checkbox"/>	4974	2023-09-11 13:42:33	10124.jpg	Pegawai Merokok	staticplayRokok10124.jpg
<input type="checkbox"/>	4975	2023-09-11 13:42:36	10125.jpg	Pegawai Merokok	staticplayRokok10125.jpg
<input type="checkbox"/>	4976	2023-09-11 13:42:45	10126.jpg	Pegawai Merokok	staticplayRokok10126.jpg
<input type="checkbox"/>	4977	2023-09-11 13:42:49	10127.jpg	Pegawai Merokok	staticplayRokok10127.jpg
<input type="checkbox"/>	4978	2023-09-11 13:42:53	10128.jpg	Pegawai Merokok	staticplayRokok10128.jpg

Gambar 4.40 Database capture

BAB V PENUTUP

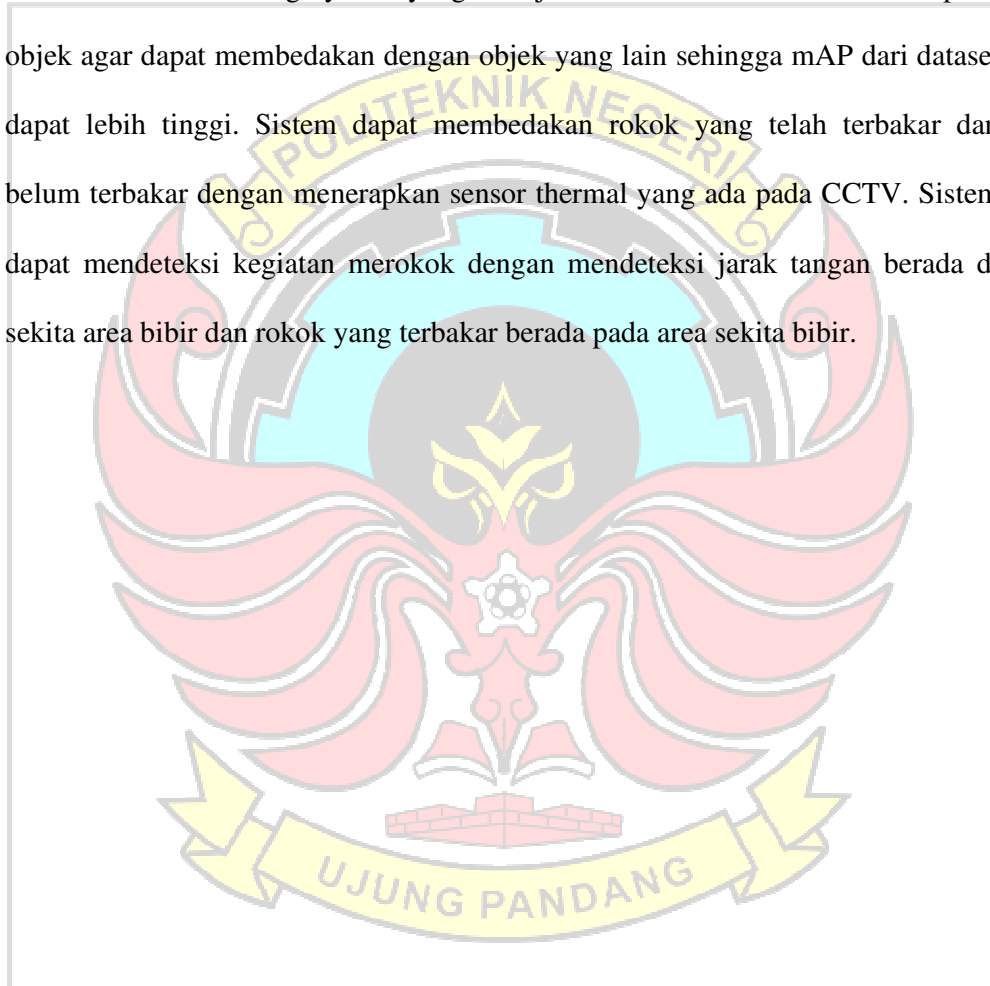
5.1 Kesimpulan

Berdasarkan pengujian Sistem implementasi RTSP pada CCTV untuk melakukan pengawasan terhadap pegawai kantor BPKAD maka dapat disimpulkan bahwa Implementasi *Real Time Stream Protocol* Pada CCTV Kawasan Tanpa Merokok berhasil dibuat dengan baik sesuai dengan perencanaan yang dilakukan sebelumnya. Berdasarkan pengujian terhadap model gambar yang telah dilatih dan diujikan sebelumnya maka dapat disimpulkan bahwa :

- a) Dengan adanya RTSP pada CCTV sistem dengan mudah melakukan pendeteksian kegiatan merokok pada ruangan. Yolov5 dapat melakukan pendeteksian kegiatan merokok pada pegawai dengan cepat. Selain itu Yolov5 dapat menentukan nilai akurasi dari gambar yang dideteksi pada objek tersebut. Hasil akurasi sistem dapat mendeteksi kegiatan merokok sebesar 66,6%.
- b) pada skenario jarak, sudut, dan intensitas cahaya sistem berhasil mendeteksi kegiatan merokok di kawasan tanpa rokok. Apabila terdapat pegawai melakukan kegiatan merokok di kawasan tanpa rokok maka sistem melakukan bounding box terhadap objek tersebut dan melakukan *screenshot* pada objek yang telah terdapat *bounding box* serta menyimpan hasil *screenshot* yang diteruskan ke notifikasi telegram sebagai bukti kegiatan merokok pegawai tersebut.

5.2 Saran

Untuk saran penelitian selanjutnya adalah untuk meningkatkan akurasi model maka memperbanyak dataset gambar dan model gambar sehingga sistem dapat mendeteksi lebih akurat pada kegiatan merokok. Pada tahap preprocessing menambahkan metode grayscale yang bertujuan memberikan warna abu abu pada objek agar dapat membedakan dengan objek yang lain sehingga mAP dari dataset dapat lebih tinggi. Sistem dapat membedakan rokok yang telah terbakar dan belum terbakar dengan menerapkan sensor thermal yang ada pada CCTV. Sistem dapat mendeteksi kegiatan merokok dengan mendeteksi jarak tangan berada di sekita area bibir dan rokok yang terbakar berada pada area sekita bibir.



DAFTAR PUSTAKA

- Angraini, Mutiara Dwi, Kusri Kusri, and Hanif Al Fatta. 2022. "Social Distancing Detection Finding Optimal Angle With Yolo V3 Deep Learning Method." *Jurnal Teknik Informatika (Jutif)* 3(5): 1449–55.
- Arsyad, Muhammad, Elsy Indah Fitria, M Rizky Farhan, and Rahmad Maulana. 2023. "Sistem Pengambil Keputusan Pemilihan Brand CCTV Terbaik Menggunakan Metode Simple Additive Weighting." *Jurnal Fusion* 3(02): 178–87.
- Aziz, Nur Chudlori, and Dimas Aryo Anggora. 2021. "Implementasi Algoritma Knn Untuk Memprediksi Potensi Penyakit Jantung Dengan Python Flask."
- Dio, Muhamad Dio Riza Pratama, Bayu Priyatna Bayu Priyatna, April Lia Hananto April Lia Hananto, and Shofa Shofiah Hilabi Shofa Shofiah Hilabi. 2022. "Deteksi Objek Kecelakaan Pada Kendaraan Roda Empat Menggunakan Algoritma YOLOv5." *Teknologi: Jurnal Ilmiah Sistem Informasi* 12(2): 15–26.
- Fitri, Kiki Reski R, Ady Rahmansyah, and Wahyuni Darwin. 2017. "Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D." In *5th Indonesian Symposium on Robotic Systems and Control*, , 22.
- Fu'adah, Yunendah Nur et al. 2022. "Glaucoma Classification Based On Fundus Images Processing With Convolutional Neural Network." *Jurnal Teknik Informatika (Jutif)* 3(3): 717–22.
- Ilham, Aldi Arista et al. 2021. "Pengujian Sistem Informasi Parkir PT KISP Berbasis Desktop Dengan Metode Black-Box." *Jurnal Informatika Universitas Pamulang* 6(1): 96–101.
- Indaryanto, Faizal, Anan Nugroho, and Alfa Faridh Suni. 2021. "Aplikasi Penghitung Jarak Dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19." *Edu Komputika Journal* 8(1): 31–38.
- Jupiyandi, Sisco et al. 2019. "Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA Dan Modified YOLO." *Jurnal Teknologi Informasi dan Ilmu Komputer* 6(4): 413–19.
- Kembuan, Olivia, Gladly Caren Rorimpandey, and Soenandar Milian Tompunu Tengker. 2020. "Convolutional Neural Network (CNN) for Image Classification of Indonesia Sign Language Using Tensorflow." In *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, IEEE, 1–5.
- Kusuma, Tsabita Al Asshifa Hadi, Koredianto Usman, and Sofia Saidah. 2021. "People Counting for Public Transportations Using You Only Look Once

Method.” *Jurnal Teknik Informatika (Jutif)* 2(1): 57–66.

Mayasari, Ulfa. 2020. “Penerapan Sistem Counter Pada Laboratorium Telekomunikasi Untuk Menghitung Jumlah Pengunjung Menggunakan Cctv Di Politeknik Negeri Sriwijaya.”

Muhaimin, Muhamad, and Wan Sen Tjong. 2021. “Real-Time Detection of Face Masked & Face Shield Using YOLO Algorithm with Pre-Trained Model and Darknet.” *Indonesian Journal of Artificial Intelligence and Data Mining* 4(2): 97–107.

Ngantung, Ronaldo Kristoforus, and M A Ineke Pakereng. 2021. “Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python.” *Jurnal Media Informatika Budidarma* 5(3): 1052–62.

Ningtyas, Dinda Fitri, and Nina Setiyawati. 2021. “Implementasi Flask Framework Pada Pembangunan Aplikasi Purchasing Approval Request.” *Jurnal Janitra Informatika dan Sistem Informasi* 1(1): 19–34.

Rahma, Lusiana, Hadi Syaputra, A Haidar Mirza, and Susan Dian Purnamasari. 2021. “Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once).” *Jurnal Nasional Ilmu Komputer* 2(3): 213–32.

Rath, Sovit. 2023. “YOLOv8 Ultralytics: State-of-the-Art YOLO Models.” *10 januari 2023*. <https://learnopencv.com/ultralytics-yolov8/> (February 23, 2023).

Razak, Angger Abdul, Adharul Muttaqin, and Erza Erza Pradipta. 2021. “Analisis Pembatasan Thread Cuda GPU Computing Pada Image Denoising.” *Jurnal EECCIS (Electrics, Electronics, Communications, Controls, Informatics, Systems)* 15(1): 17–22.

Rizki, Rahmad, Rizal Munadi, and Syahrial Syahrial. 2019. “Analisis Performansi Video Streaming Dengan Menggunakan Protokol RTSP Pada Jaringan IEEE 802.11 N.” *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)* 2(1): 9–12.

Saputra, Muhammad Cahaya, Andri Anto Tri Susilo, Bunga Intan, and Budi Santoso. 2022. “Sistem Pengenalan Masker Menggunakan Algoritma Yolo Di Masjid Agung As–Salam.” *Jusikom: Jurnal Sistem Komputer Musirawas* 7(2): 132–43.

Shamrat, F M Javed Mehedi et al. 2021. “A Deep Learning Approach for Face Detection Using Max Pooling.” In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 760–64.

Singh, Mandeep et al. 2019. “Implementation of Database Using Python Flask

Framework.” *International Journal of Engineering and Computer Science* 8(12): 24890–93.

Surbakti, Agung Wibowo Ardiyanta, and Rahmi Eka Putri. 2022. “Penghitung Pengunjung Dan Deteksi Masker Menggunakan OpenCV Dan YOLO.” *CHIPSET* 3(02): 83–93.

Sutarti, Sutarti, Sunny Samsuni, and Isnan Asseghaf. 2019. “Sistem Keamanan Rumah Melalui Pengenalan Wajah Menggunakan Webcam Dan Library Opencv Berbasis Raspberry Pi.” *Jurnal Dinamika Informatika* 8(2): 13–26.

SUTRISNO, MUHAMMAD. 2022. “Deteksi Kantuk Berbasis Jetson Nano Bagi Pengemudi Roda Empat Menggunakan Yolov5.”

Taufiq, Reny Medikawati, Yoze Rizki, and M Rizki Amanda Pratama. 2022. “Simulasi Deteksi Golongan Kendaraan Pada Gerbang Tol Menggunakan YOLOv4.” *Jurnal CoSciTech (Computer Science and Information Technology)* 3(2): 199–206.

