PERANCANGAN SISTEM PENDETEKSI ALAT PELINDUNG DIRI (APD) UNTUK PENGAWASAN K3 DI LINGKUNGAN KERJA SECARA *REALTIME* BERBASIS RASPBERRY PI 4 DENGAN NOTIFIKASI TELEGRAM

SKRIPSI

Disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T)

Disusun Oleh:

PANDU AKBAR MAULANA

NPM. 3332160026

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS SULTAN AGENG TIRTAYASA

2023

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya sebagai penulis Skripsi berikut:

Judul	: Perancangan Sistem Pendeteksi Alat Pelindung Diri (APD) Untuk Pengawasan K3 Di Lingkungan Kerja Secara *Realtime* Berbasis Raspberry Pi Dengan Notifikasi Telegram
Nama Mahasiswa	: Pandu Akbar Maulana
NPM	: 3332160026
Fakultas/Jurusan	: Teknik/Teknik Elektro

Menyatakan dengan sesungguhnya bahwa Skripsi tersebut diatas adalah benar-benar hasil karya asli saya dan tidak memuat hasil karya orang lain, kecuali dinyatakan melalui rujukan yang benar dan dapat dipertanggungjawabkan. Apabila dikemudian hari ditemukan hal-hal yang menunjukkan bahwa sebagian atau seluruh karya ini bukan karya saya, maka saya bersedia dituntut melalui hukum yang berlaku. Saya juga bersedia menanggung segala akibat hukum yang timbul dari pernyataan yang secara sadar dan sengaja saya nyatakan melalui lembar ini.

Cilegon, … …… 2023

Pandu Akbar Maulana

NPM. 3332160026

LEMBAR PENGESAHAN

Dengan ini ditetapkan bahwa Skripsi berikut:

Judul	: Perancangan Sistem Pendeteksi Alat Pelindung Diri (APD) Untuk Pengawasan K3 Di Lingkungan Kerja Secara *Realtime* Berbasis Raspberry Pi Dengan Notifikasi Telegram
Nama Mahasiswa	: Pandu Akbar Maulana
NPM	: 3332160026
Fakultas/Jurusan	: Teknik/Teknik Elektro

Telah diuji dan dipertahankan pada tanggal ….. …… 2023 melalui Sidang Skripsi di Fakultas Teknik Universitas Sultan Ageng Tirtayasa Cilegon dan dinyatakan LULUS.

Dewan Penguji

		Tanda Tangan
Pembimbing I	: Dr. Ing. M. Iman Santoso, M.Sc.	……
Penguji I	:	……
Penguji II	:	……

Mengetahui,
Ketua Jurusan

Dr. Romi Wiryadinata, S.T., M.Eng.
NIP. 198307032009121006

PRAKATA

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, Karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Sultan Ageng Tirtayasa. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Kedua orang tua tercinta serta seluruh keluarga yang telah memberikan nasehat, semangat, doa, dan materi yang tak terhingga nilainya,
2. Bapak Dr. Romi Wiryadinata, S.T., M.Eng., selaku Ketua Jurusan Teknik Elektro,
3. Bapak Dr. -Ing. M. Iman Santoso, S.T., M.Sc., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini,
4. Bapak Prof. Dr. Alimuddin, S.T., M.M., M.T., selaku dosen pembimbing akademik yang telah memberikan bimbingan kepada saya selama masa perkuliahan ini,
5. Seluruh dosen Teknik Elektro Universitas Sultan Ageng Tirtayasa beserta sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu

Cilegon, … …… 2023

Penulis

ABSTRAK

Pandu Akbar Maulana
Teknik Elektro

Perancangan Sistem Pendeteksi Alat Pelindung Diri (APD) Untuk Pengawasan K3 Di Lingkungan Kerja Secara *Realtime* Berbasis Raspberry Pi Dengan Notifikasi Telegram

Penelitian ini merancang sistem APD untuk pengawasan K3 di lingkungan kerja berbasis Raspberry Pi 4B dan aplikasi Telegram sebagai penerima notifikasi secara *realtime*. Sistem ini akan melakukan pendeteksian dan pemantauan para pekerja yang akan memasuki area wajib APD menggunakan Webcam sebagai alat penangkap gambar yang informasinya akan dikirimkan kepada pengawas K3 melalui aplikasi Telegram yang ter*install* pada *smartphone*-nya guna pengawasan jarak jauh. Dari pendeteksian objek yang memanfaatkan metode SSD-MobilenetV2, sistem ini mendapatkan nilai akurasi sebesar 93% pada siang hari, 90% pada sore hari, 43% pada malam hari dan 76% dari total keseluruhan pengujian 3 kondisi waktu tersebut dengan rata-rata waktu pengiriman notifikasi ke aplikasi Telegram sebesar 3,53 detik.

Kata Kunci:
Raspberry P, K3, *realtime*, SSD-MobilenetV2, Telegram.

ABSTRACT

Pandu Akbar Maulana
Electrical Engineering

Design of Personal Protective Equipment (PPE) Detection System for Realtime Occupational Safety and Health Monitoring Using Raspberry Pi with Telegram Notifications

This research designs a PPE detection system for occupational safety and health (OSH) monitoring in the workplace, based on Raspberry Pi 4B and the Telegram application for real-time notifications. The system performs detection and monitoring of workers entering designated PPE areas using a webcam as the image capture device, with the information sent to the OSH supervisor through the Telegram application installed on their smartphone for remote monitoring. By utilizing the SSD-MobilenetV2 object detection method, the system achieves an accuracy rate of 93% during daytime, 90% during evening, 43% during nighttime, and 76% of the total testing of the 3 time conditions with an average notification delivery time to the Telegram application of 3.53 seconds.

Keywords:
Raspberry Pi, OSH, real-time, SSD-MobilenetV2, Telegram.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kecelakaan kerja adalah sebuah kecelakaan yang terjadi akibat pekerjaan dan juga termasuk penyakit yang timbul karena hubungan kerja, demikian pula kecelakaan yang terjadi dalam perjalanan ke dan dari tempat kerja. Kecelakaan kerja merupakan kejadian tidak terduga dan tidak diinginkan baik kecelakaan akibat langsung pekerjaan maupun kecelakaan yang terjadi pada saat pekerjaan[1]. Kejadian tersebut biasanya bersifat fisik dan merugikan. Menurut Rowlinson, kecelakaan adalah kejadian yang tidak direncanakan, tak terkontrol, yang dapat menyebabkan atau mengakibatkan luka-luka pada pekerja, kerusakan pada peralatan dan kerugian lainnya[2].

Heinrich menyatakan lima urutan kejadian kecelakaan berdasar teori domino, bahwa: Kecelakaan kerja terjadi karena faktor bawaan, kurangnya pengetahuan dan keahlian dalam melakukan pekerjaan, lingkungan sosial dan lingkungan kerja yang tidak tepat. Enam puluh persen kecelakaan kerja disebabkan oleh kesalahan manusia hal ini antara lain karena keterbatasan pengetahuan pekerja, lalai dan ceroboh dalam bekerja, tidak melaksanakan prosedur kerja yang diberikan dan tidak disiplin melaksanakan peraturan keselamatan kerja termasuk penggunaan alat pelindung diri[3]. Berdasarkan catatan *International Labour Organization* pada tahun 2013, 1 pekerja di dunia meninggal setiap 15 detik karena kecelakaan kerja dan 160 pekerja mengalami sakit akibat kerja. Tahun sebelumnya ILO mencatatat angka kematian dikarenakan kecelakaan dan penyakit akibat kerja sebanyak 2 juta kasus setiap tahun[4]. Beberapa hasil penelitian menunjukkan bahwa faktor manusia memegang peranan penting dalam munculnya kecelakaan kerja. Kecelakaan dalam bekerja dapat disebabkan berbagai macam hal, mulai dari kurang disiplinnya pekerja, kecerobohan, dan kelalaian pekerja dengan tidak menggunakan perlengkapan APD (Alat pelindung Diri)[5]. Oleh sebab itu penggunaan Alat Pelindung Diri (APD) merupakan tahap akhir dari pengendalian kecelakaan maupun penyakit akibat kerja[6]. Alat Pelindung Diri (APD) berfungsi untuk

melindungi tubuh terhadap bahaya-bahaya kecelakaan kerja dan mengurangi dampak dari kecelakaan kerja yang terjadi[7].

Kedisiplinan para pekerja dalam penggunaan alat pelindung diri terbilang masih rendah, sehingga resiko yang disebabkan kecelakaan kerja yang dapat membahayakan pekerja terbilang dalam jumlah yang besar. Berdasarkan angka kecelakaan kerja di Indonesia tahun 2011, angka kecelakaan mencapai 99.491 kasus. Jumlah tersebut meningkat jika dibandingkan dengan tahun sebelumnya. Pada tahun 2007 sebanyak 83.714 kasus, tahun 2008 sebanyak 94.736 kasus, tahun 2009 sebanyak 96.314, dan tahun 2010 sebanyak 98.711 kasus. Masih lemahnya kedisiplinan dan kesadaran masyarakat pada khususnya pekerjalah yang menyebabkan angka kecelakaan kerja di Indonesia tergolong cukup tinggi[8]. Berkaitan dengan sistem deteksi untuk para pekerja mengenai APD, banyak perusahaan yang masih mengadopsi sistem manual dalam pengecekan APD oleh manusia. Secara umum sistem tersebut tidak layak karena keterlibatan manusia yang efisiensinya menurun dalam durasi yang lama[9]. Otomatisasi menjadi peran penting untuk memantau para pekerja yang tidak menggunakan APD. Dengan adanya otomatisasi mampu mengurangi beban kerja *supervisor* terhadap pekerja yang tidak memakai APD. Solusi yang tepat untuk menghadapi masalah ini adalah menggunakan object detection pada video kamera pengintai di zona wajib alat pelindung diri[10].

Berdasarkan paparan tersebut penulis akan melakukan penelitian dengan judul “Perancangan Sistem Pendeteksi Alat Pelindung Diri (APD) Untuk Pengawasan K3 Di Lingkungan Kerja Secara *Realtime* Berbasis Raspberry Pi 4 Dengan Notifikasi Telegram”. Pada penelitian ini sistem akan menggunakan kamera yang diletakan pada sudut-sudut atau tempat yang strategis untuk memantau zona wajib alat pelindung diri dilingkungan aktifitas kerja, yang mana kamera ini digunakan untuk merekam objek secara *real time* sehingga akan memudahkannya kegiatan pengawasan K3 dan mendisiplinkan pekerja terhadap penggunaan APD.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah dikemukakan diatas, maka dapat dirumuskan permasalahan yang melatar belakangi penelitian ini adalah sebagai berikut :

1. Membuat sistem yang dapat mendeteksi APD secara *realtime* di zona wajib APD pada lingkungan kerja demi meminimalisir adanya kecelakaan kerja.
2. Mencari tau mengenai kinerja dari sistem pendeteksi APD menggunakan Raspberry Pi 4.
3. Menganalisis proses pengiriman hasil data ke Telegram.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang sebuah sistem yang dapat memberikan peringatan di lapangan kepada pekerja terhadap kelengkapan penggunaan APD serta mengirim notifikasi berupa tangkapan gambar dari kamera webcam kepada pengawas K3 untuk pengawas jarak jauh dan meminimalisir adanya kecelakaan kerja.
2. Mengetahui kinerja dan akurasi yang didapatkan dari sistem pendeteksi APD menggunakan Raspberry Pi 4.
3. Mengetahui proses pengiriman hasil data ke Telegram.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi akademisi, hasil penelitian ini diharapkan sangat membantu para akademisi yang melakukan penelitian dalam bidang sistem deteksi APD (Alat Pelindung Diri) berbasis Raspberry Pi dilapangan kerja.
2. Bagi pendidikan, hasil penelitian ini diharapkan menambah informasi dan pengetahuan mengenai sistem deteksi APD (Alat Pelindung Diri) berbasis Raspberry Pi dilapangan kerja dengan memanfaatkan aplikasi Telegram dan model SSD-MobilenetV2 sebagai proses pengembangannya.
3. Bagi keberlanjutan pendidikan, hasil penelitian ini diharapkan membantu dan mendorong penelitian mengenai revolusi teknologi dan modernisasi pengawasan K3 di dunia kerja.

4. Bagi masyarakat dan perusahaan, hasil penelitian ini diharapkan dapat membantu mengurangi tingginya tingkat kecelakaan didunia kerja akibat rendahnya kesadaran masyarakat terhadap pentingnya aspek K3 dengan memanfaatkan sistem deteksi APD (Alat Pelindung Diri) berbasis Raspberry Pi dilapangan kerja.

1.5 Batasan Masalah

Penelitian ini dibatasi dengan hal-hal berikut:

1. Menggunakan Raspberry Pi 4B sebagai pusat kendali.
2. Menggunakan bahasa pemrograman *Python*.
3. Menggunakan *Webcam* untuk pengambilan data latih dan pengujian.
4. Menggunakan WiFi sebagai koneksi internetnya..
5. Menggunakan metode CNN dengan model SSD-MobilenetV2.
6. Hanya melakukan pendeteksian dari 6 macam kondisi penggunaan APD yang sudah ditentukan dan dilatih.
7. Sistem yang dibuat hanya untuk kondisi cuaca baik atau tidak hujan.

1.6 Sistematika Penulisan

Peneliti membagi penggambaran tetang susunan materi yang akan dibuat, berikut ini sistematika penulisan skripsi ini terdiri:

BAB I: PENDAHULUAN

Bab ini membahas uraian singkat mengenai segala sesuatu dalam penelitian ini, pada bab ini juga dijelaskan tentang latar belakang, tujuan penelitian, rumusan masalah, manfaat penelitian, batasan masalah, serta sistematika penelitian.

BAB II: TINJAUAN PUSTAKA

Bab ini berisi tentang penjelasan mengenai tinjauan pustaka dan dasar teori, didalamnya memuat kajian-kajian dari hasil artikel publikasi terkait dengan topik penelitian, dan landasan - landasan teori yang menunjang penyelesaian penelitian.

BAB III: METODOLOGI PENELITIAN

Bab ini berisi tentang metodologi penelitian yang digunakan untuk mendapatkan hasil yang diinginkan. Selain itu pada bab ini berisi tentang proses pengambilan data yang diperlukan dan juga berisi perangkat-perangkat yang

digunakan pada penelitian ini, baik perangkat lunak maupun perangkat keras. Terdapat pula perancangan penelitian yang berupa flowchart beserta penjelasan langkah-langkahnya, dan yang terakhir pada bab ini berisi lokasi dan waktu penelitian.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi tentang analisis dan data simulasi hasil percobaan yang telah dilakukan.

BAB V: PENUTUP

Bab ini berisi uraian hasil penelitian yang ditulis secara singkat, padat, dan jelas serta berkolerasi dengan tujuan yang dibuat. Terdapat pula saran yang dapat memperbaiki penelitian yang telah dilakukan.

BAB II
TINJAUAN PUSTAKA

2.1 Alat Pelindung Diri (APD)

Menurut Tarwaka, Alat Pelindung Diri (APD) merupakan seperangkat alat keselamatan yang digunakan oleh pekerja untuk melindungi seluruh atau sebagian tubuhnya dari kemungkinan adanya pemaparan potensi bahaya lingkungan kerja terhadap kecelakaan dan penyakit akibat kerja. Sedangkan menurut Budiono, Alat Pelindung Diri (APD) adalah seperangkat alat yang digunakan tenaga kerja untuk melindungi sebagian atau seluruh tubuhnya dari potensi kecelakaan kerja. APD tidak secara sempurna dapat melindungi tubuhnya, tetapi dapat mengurangi tingkat keparahan yang mungkin terjadi. Pengendalian ini sebaiknya tetap dipadukan dan sebagai pelengkap pengendalian teknis atau pengendalian administratif[11].

Berdasarkan surat dari Menteri Tenaga Kerja dan Trasmihrasi nomor PER. 08/MENVIII/2010, jenis-jenis alat perlindungan diri antara lain[12]:

1. Alat pelindung kepala, yang terdiri dari helm pengaman (*safety helmet*), topi atau tudung kepala, penutup atau pengaman rambut, dan lain-lain;
2. Alat pelindung mata dan muka, yakni kacamata pengaman (*spectacles*), *goggles*, tameng muka (*face shield*), masker selam, tameng muka dan kacamata pengaman dalam kesatuan (*full face masker*);
3. Alat pelindung telinga, yakni sumbat telinga (*earplug*) dan penutup telinga (*earmuff*);
4. Alat pelindung pernapasan beserta perlengkapannya, yang terdiri dari masker, respirator, katrit, canister, *re-breather*, *airline respirator*, *Continues Air Supply Machine*, *Air Hose Mask Respirator*, tangka selam dan regulator (*Self-Contained Underwater Breathing Apparatus/SCUBA*), *Self-Contained Breathing Apparatus (SCBA), dan emergency breathing apparatus*.

2.2 *System Real Time*

Suatu sistem yang dapat mengendalikan sistem fisik dikenal sebagai sistem *real time.* Sistem *real time* merupakan bentuk khusus dari sistem *online*. Suatu sistem dikatakan *real time* jika tidak hanya mengutamakan ketepatan dalam proses,

tapi juga interval waktu proses tersebut dilakukan. Dengan kata lain, sistem *real time* adalah sistem yang menggunakan *deadline*, yaitu pekerjaan harus selesai jangka waktu tertentu. Pada sistem *real time*, digunakan batasan waktu. Sistem dinyatakan gagal jika melewati batasan yang ada. Sistem *real time* banyak digunakan dalam berbagai macam aplikasi. Sistem tersebut ditanam di dalam alat khusus seperti di kamera, *mp3 players*, serta di pesawat dan mobil[13].

Terdapat dua bentuk sistem *real time*, yaitu sistem *hard real time* dan sistem *soft real time*. Sistem *hard real time* menjamin tugas kritis diselesaikan tepat waktu. Pada sistem ini penyimpan sekunder terbatas atau tidak digunakan, data langsung dikirim ke *memory* atau *read-only memory* (ROM) dalam waktu singkat. Pada sistem *hard real time* terjadi konflik pada sistem *time sharing* dan tidak didukung oleh sistem operasi tujuan umum. Bentuk lainnya adalah *soft real time* dimana tugas kritis mendapatkan prioritas lebih tinggi dari tugas lain dan setelah satu *task* selesai maka *task* berprioritas ini akan diselesaikan. Sistem ini terbatas pada industri pengontrol robot. Sangat berguna pada aplikasi multimedia dan *virtual rality* yang membutuhkan fitur sistem operasi tertentu[13].

Suatu sistem dikatakan *real time* jika sistem tersebut dapat mendukung eksekusi program atau aplikasi dengan waktu yang memiliki batasan. Kelebihan suatu sistem *real time*, yaitu :

a) Memiliki batasan waktu dan memenuhi deadline.
b) Memiliki waktu yang dapat diprediksi.
c) Terfokus pada hal-hal yang penting saja, yang tidak penting tidak dikerjakan sehingga dapat menemukan tingkat effisiensi waktu.

Selain kelebihan, adapun beberapa kekurangan *real time system*, yaitu :

a) Hanya memiliki satu tujuan, seperti mentransfer sebuah lagu dari komputer ke *music player*.
b) Kebanyakan sistem banyak yang ada memiliki *physical space* yang terbatas.
c) Sistem harus memenuhi persyaratan waktu yang

2.3 Pendeteksian Objek

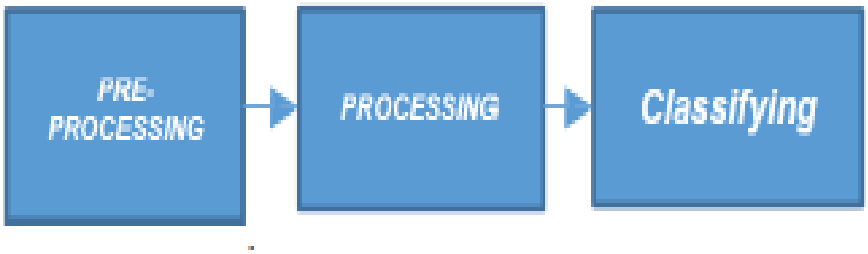
Deteksi objek (Object Detection) adalah teknik visi komputer untuk menemukan contoh objek dalam gambar atau video. Algoritma deteksi objek

biasanya memanfaatkan pembelajaran mesin atau pembelajaran mendalam untuk menghasilkan hasil yang bermakna. Algoritma deteksi objek yang cepat dan akurat akan memungkinkan komputer dapat melakukan hal serupa hingga berpotensi untuk menyelesaikan tugas secara umum. Ketika manusia melihat gambar atau video, manusia dapat mengenali dan menemukan objek dalam beberapa saat berbeda dengan komputer yang memerlukan komputasi yang kompleks. Tujuan deteksi objek adalah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer. Cara kerja deteksi objek yaitu menempatkan keberadaan objek dalam gambar dan menggambar kotak pembatas di sekitar objek itu. Ini biasanya melibatkan dua proses, yaitu mengklasifikasikan jenis objek dan kemudian menggambar kotak di sekitar objek itu[14].

Pendeteksian objek tersebut dapat dilakukan dengan berbagai macam metode yang umumnya melakukan pembacaan fitur-fitur dari seluruh objek pada citra input. Fitur dari objek pada citra input itu sendiri akan dibandingkan dengan fitur dari model yang digunakan atau *template*. Oleh karena itu, nantinya perbandingan tersebut dapat digunakan untuk menentukan apakah suatu objek terdeteksi sebagai template yang dimaksud atau tidak. Sistem pendeteksian objek perlu melatih dan menguji dataset dengan *bounding box* dan diberi label untuk kelas per setiap objek untuk proses pengenalan dari objek itu sendiri. Dan demi mencapai tujuan tersebut, dimana terdapat banyak dataset untuk menghasilkan model *Deep Learning*.

2.3.1 *Convolution Neural Network*

Convolutional Neural Network (CNN) adalah salah satu jenis *Neural Network* (NN) yang umum digunakan untuk mengenali sebuah objek pada suatu *image* maupun video. CNN adalah bagian dari macam-macam jenis *Multilayer Perceptron* yang terinspirasi dan mengadopsi dari jaringan syaraf manusia. Secara garis besar metode ini tidak berbeda jika dibandingkan dengan *Neural Network* pada umumnya. CNN terdiri dari neuron yang memiliki *weight*, bias dan *activation function*. CNN memiliki proses yang umum, proses ini memiliki 3 lagkah tahapan, yaitu *pre-processing*, *processing*, dan *classifying*[5].



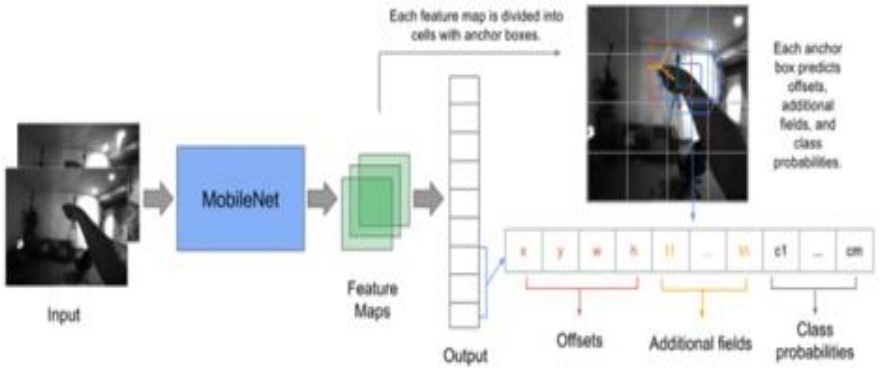
Gambar 2.1 Proses Umum CNN

Proses CNN diawali dengan tahapan *pre-processing* pada Metode *Convolutional Neural Netwok* (CNN) terdiri dari dua tahapan, yaitu pembentukan atau pembuatan dataset dan mengkonversi dataset RGB menjadi *grayscale*. Kemudian proses selanjutnya adalah processing yang tersusun oleh beberapa proses yaitu konvolusi citra, *max pooling*, *training* NN. Untuk proses yang terakhir adalah *classifying* (pengklasifikasian) yang memiliki satu proses yaitu penentuan output[5].

2.3.2 MobileNet

MobileNet merupakan model arsitektur *Convolutional Neural Network* (CNN) untuk klasifikasi gambar dan *Mobile Vision*. Terdapat pula model yang lainnya, namun yang membuat MobileNet istimewa yaitu daya komputasi yang sangat kecil untuk menjalankan dan menerapkan pembelajaran (*learning*)[15]. MobileNet disini dapat berperan sebagai *Network Model* dan akan bekerja untuk mengekstrak fitur yang nantinya diklasifikasi. Jadi MobileNet sangat dibutuhkan untuk membantu mengklasifikasi objek yang terdapat dalam suatu gambar[14].

Tedapat 2 jenis konvolusi MobileNet, yaitu *depthwise convolution* dan *pointwise convolution*. Arsitektur yang memanfaatkan kedua bagian tersebut yaitu *Batch Normalization* (BN) dan *Rectified-Linear unit* (ReLU). MobileNet itu sendiri dibangun di atas arsitektur jaringan yang efisien dengan menggunakan konvolusi, dimana hal tersebut dapat dipisahkan secara mendalam (*Depthwise Separable Convolution*) untuk menghasilkan *Deep Neural Network* yang ringan. Alur kerja MobileNet dapat dilihat pada Gambar 2.2 berikut.

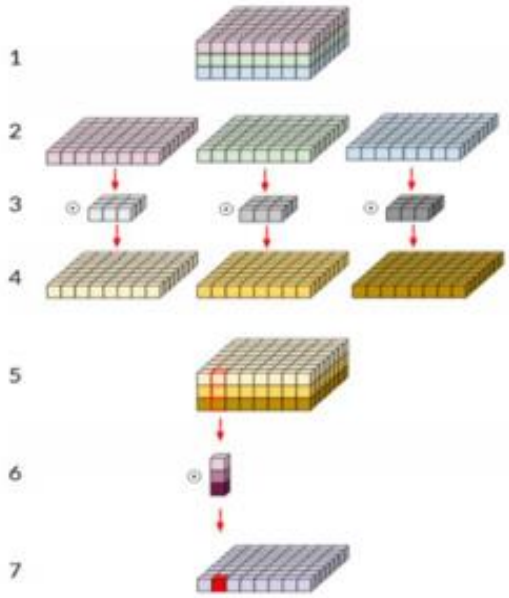


Gambar 2.2 Alur Kerja MobileNet

DSC atau *Depthwise Separable Convolution* ini menggantikan konvolusi standar dengan 2 tahap pengoperasian, yaitu :

1. *Depthwise Convolution*, dimana setiap *filter* D_F *x* D_F hanya melakukan proses *filter* terhadap sebuah *feature map input* secara mendalam.
2. *Pointwise Convolution*, dimana 1x1 *convolution layer* digunakan untuk menggabungkan jalur informasi dari *depthwise layer*.

Pemisahan secara mendalam ini menyebabkan jalur konvolusi menjadi jauh lebih efisien dengan menggunakan parameter yang jauh lebih sedikit. Ilustrasi DSC dapat dilihat pada Gambar 2.3 berikut.

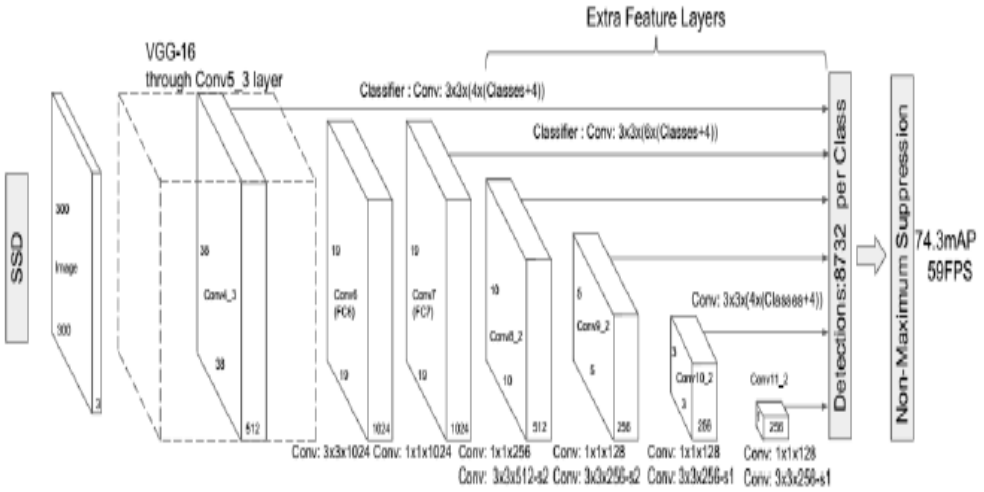


Gambar 2.3 Konvolusi *Depthwise* (2-5) dan *Pointwise* (6-7)

2.3.3 *Single Shot Detector* (SSD)

Single Shot Detector (SSD) merupakan salah satu metode yang digunakan untuk mengenali atau mendeteksi sebuah objek pada suatu gambar dengan menggunakan *single deep neural network*, metode ini juga merupakan salah satu algoritma deteksi objek yang populer karena kemudahannya dalam implementasi dan pengakurasian yang baik untuk rasio yang dibutuhkan komputasi. *Single Shot Detector* (SSD) pun hanya perlu mengambil satu bidikan tunggal untuk mendeteksi beberapa objek didalam gambar dan juga memiliki karakteristik performa FPS yang cukup cepat serta cenderung lebih akurat daripada *framework* yang lain[16].

Metode *Single Shot Detector* (SSD) ini termasuk kedalam deteksi object secara *real time*, meskipun lebih intuitif daripada rekan-rekannya seperti R-CNN, Fast R-CNN *Faster* R-CNN dan *You Only Look Once* (YOLO). *Single Shot Detector* (SSD) adalah algoritma yang sangat kuat, dimana dalam desain yang sederhana implementasinya langsung dari GPU dan sudut pandang kerangka kerja pembelajaran yang dalam dan dengan demikian melakukan pengangkatan berat deteksi dengan kecepatan kilat.



Gambar 2.4 Proses Pelatihan dengan *Single Shot MultiBox Detector*

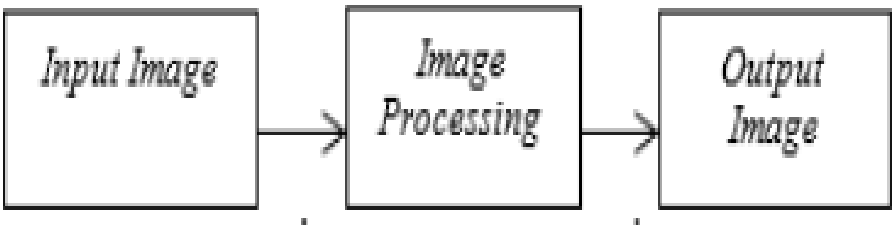
Pada arsitekturnya SSD termasuk ke dalam jenis *Convolutional Neural Network* (CNN) yang merupakan salah satu jenis *Neural Network* yang biasa digunakan pada data *image*. CNN juga bisa digunakan untuk mendeteksi dan mengenali objek dalam gambar. Arsitektur dari CNN dibagi menjadi 2 bagian besar, yaitu[17]:

1. *Feature Extraction Layer* merupakan bagian yang melakukan *encoding* dari sebuah *image* menjadi *features*, dimana *features* nya berupa angka-angka yang merepresentasikan gambar tersebut.
2. *Convolutional Layer*, terdiri dari *neuron* yang disusun demikian rupa hingga membentuk sebuah *filter* dengan panjang dan tinggi (*pixels*). Secara matematis, *Convolutional Layer* atau konvolusi merupakan integral yang mencerminkan jumlah lingkaran dari sebuah sudut fungsi F yang digeser atas fungsi G sehingga menghasilkan fungi H.

2.4 *Image Processing*

Pengolahan citra digital (*Image Processing*) memiliki pengertian sederhana yaitu manipulasi dan analisis suatu gambar oleh komputer. Sedangkan yang dimaksud dengan informasi gambar disini adalah gambar-gambar visual dalam dua dimensi. Segala operasi untuk memperbaiki, menganalisis, atau mengubah suatu gambar disebut *image processing*[18].

Sistem *image processing* memiliki konsep dasar yang mana diambil dari kemampuan indera penglihatan manusia dan dihubungkan dengan kemampuan otak manusia. *Image processing* merupakan gabungan cabang ilmu, seperti optik, matematika, elektronik, fotografi, dan teknologi komputer. *Image processing* memiliki tujuan yaitu menganalisis suatu gambar sehingga informasi baru tentang gambar dibuat menjadi lebih jelas. Gambaran proses dari *image processing* dapat dilihat pada Gambar 2.5 berikut.



Gambar 2.5 *Image Processing*

2.5 *OpenCV*

OpenCV (Open Source Computer Vision Library) adalah *library* untuk *computer vision* dan *machine learning* berkode sumber terbuka. *OpenCV* dibuat

untuk menyediakan infrastruktur umum untuk pengaplikasian *computer vision* dan mempercepat pembangunan projek yang menggunakan algoritma *machine learning* baik di bidang komersial maupun penggunaan pribadi. OpenCV dibuat dengan menggunakan bahasa C dan C++ pada tahun 1999 yang bisa berjalan dibawah sistem operasi Linux, Windows, Mac OS. Pengembangan *library* ini aktif untuk antarmuka di berbagai macam bahasa pemrograman seperti Python, Ruby, Matlab dan bahasa pemrograman lainnya[19].

OpenCV yang sekarang bisa disebut sebagai *standalone library* karena sudah tidak memerlukan *library* dari Intel *Image Processing Library*. *OpenCV* mendukung *multiplatform*, dapat mendukung baik windows ataupun linux, dan sekarang telah mendukung MacOSX dan android. *OpenCV* mempunyai banyak fitur yang dapat dimanfaatkan, diantaranya yaitu[20]:

1. *Image and video* I/O
2. *Modul computer vision high level*
3. *Computer vision* secara umum dan pengolahan citra digital (untuk *low* dan *mid level Application Programming Interface*)
4. Sampling gambar dan transformasi
5. Metode untuk AI dan *machine learning*
6. Metode untuk menciptakan dan menganalisa gambar biner serta untuk memperhitungkan permodelan 3D.

2.6 Bahasa Pemrograman Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang dianggap mudah dan berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar[21]. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan diberbagai platfrom sistem operasi seperti Linux, Windows, Mac OS, JVM, OS/2, Amiga, Palm, dan Symbian. Berdasarkan penjelasan singkat, python memiliki beberapa fitur dan kelebihannya sendiri, diantaranya[22]:

1) Memiliki koleksi kepusatakaan yang banyak. Itu artinya, telah tersedia modul-modul siap pakai untuk berbagai keperluan.
2) Memiliki struktur bahasa yang jelas, sederhana dan mudah dipelajari.
3) Memiliki sistem pengelolaan memori otomatis (*garbage collection*) seperti halnya bahasa *java*.
4) Berorientasi objek.
5) Bersifat modular, sehingga mudah dikembangkan dengan menciptakan modul-modul baru, baik dibangun dengan menggunakan bahasa *Python* maupun C/C++.

2.7 Tensorflow *Lite*

Tensorflow merupakan sebuah *framework machine learning* yang dikembangkan oleh google, tensorflow merupakan sebuah *machine learning* yang ditulis dalam bahasa python. Paket Tensorflow Object Detection API adalah proses untuk menyelesaikan masalah deteksi objek. Teknik ini dapat mendeteksi objek secara *real-time* dalam suatu gambar. TensorFlow juga mampu untuk mengakurasikan kecepatan dan memori pada aplikasi, sehingga bisa memodifikasi model untuk memenuhi kebutuhan[23].

Tensorflow lite sendiri merupakan *framework deep learning open source* untuk inferensi *on-device*. Tensorflow lite dikembangkan oleh tensorflow sendiri guna untuk mendukung sistem pembelajaran mesin ke perangkat. Jadi tensorflow lite menyediakan *framework* untuk model tensorflow yang telah dilatih untuk dikompresi dan diterapkan ke dalam perangkat atau sejenisnya[24].

2.8 Google Colaboratory

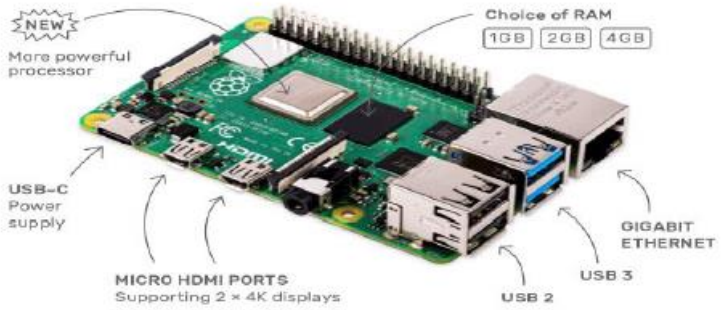
Google Colaboratory merupakan layanan *cloud* yang didasarkan dari *environment Jupyter Notebooks* dengan tujuan untuk menyebarkan pendidikan dan penelitian di bidang *machine learning* berbasis bahasa pemrograman *python*. Layanan ini dapat menyediakan *runtime* yang sepenuhnya dikonfigurasi untuk akses gratis ke GPU yang kuat. Colab dapat dikonfigurasi dengan *library machine learning* dan kecerdasan buatan seperti TensorFlow, Matplotlib, dan Keras.

Infrastruktur Google Colaboratory dihosting di *platform* Google Cloud[25]. Google Colaboratory memiliki beberapa kelebihan diantaranya:

1) Berbasis *cloud*, sehingga tidak memakan *resource* yang ada pada komputer yang kita gunakan.
2) Terintegrasi dengan Google Drive, dimana semua data bisa disimpan, *compile-run* secara *cloud*.
3) *Free* GPU, Google Colab menawarkan layanan *free* GPU yang tentunya sangat bermanfaat dalam proses membangun sebuah *deep learning model* atau *machine learning* yang sejatinya membutuhkan *resource* yang tinggi untuk menjalankannya.
4) *Built-in-library machine learning* yang lengkap.
5) Bisa diakses menggunakan berbagai macam *browser*, seperti *Google Chrome, Edge,* Firefox.
6) *No-Configuration-Needed*, tidak membutuhkan konfigurasi pada saat ingin menggunakannya.

2.9 Raspberry Pi

Raspberry Pi adalah salah satu peralatan yang umum digunakan dalam membangun sistem Internet of Things. Raspberry Pi merupakan pemroses mikro yang berukuran kompak, minim pemakaian daya, walaupun memiliki kemampuan komputasi yang rendah. Raspberry Pi memungkinkan pengguna dapat melakukan apapun selayaknya yang dapat dilakukan oleh PC atau laptop, dimana pada sirkuit raspberry pi dilengkapi dengan komponen *proccessor*, RAM dan *port hardware* yang biasa ditemui pada komputer/laptop. Raspberry Pi cocok digunakan sebagai pemroses lokal seperti agregator, preprocessor, dan gateway[26].



Gambar 2.6 Raspberry Pi 4

2.10 Webcam

Webcam atau *Web Camera* merupakan perangkat yang berupa sebuah kamera digital yang dapat mengirimkan video/gambar secara *realtime* pada suatu program pengolah pesan cepat, atau aplikasi pemanggil video dengan bantuan sambungan internet. Kamera web bisa diartikan juga sebagai sebuah kamera video digital berukuran kecil yang dapat dihubungkan ke komputer/laptop melalui *port* USB, *port* COM atau dengan jaringan Wi-Fi atau *Ethernet.*

Gambar 2.7 Webcam

2.11 *Speaker*

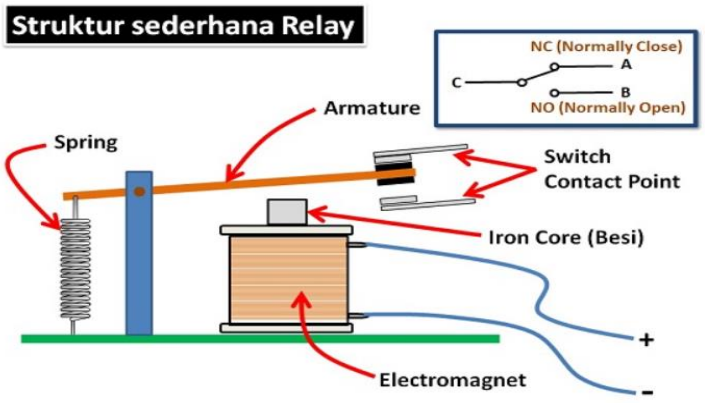
Speaker merupakan Transduser yang secara khusus dapat mengubah sinyal listrik menjadi frekuensi audio dalam bentuk sinyal suara yang dapat didengan oleh telinga manusia dengan cara mengetarkan membran pada *Speaker* tersebut sehingga terjadinya gelombang suara. Komponen utama pada *Speaker* ini adalah *Cone Suspension*, Magnet Permanen, *Voice Coil* dan juga kerangka *speaker*. Dalam rangka menterjemahkan sinyal listrik menjadi suara, *speaker* memiliki komponen elektromagnetik yang terdiri dari kumparan yang disebut sebagai *Voice Coil* untuk membangkitkan medan magnet dan dapat berinteraksi dengan Magnet Permanen sehingga bisa menggerakan *cone speaker* maju dan mundur.

Gambar 2.8 *Speaker*

2.12 Relay

Relay merupakan saklar (*switch*) yang dioperasikan secara listrik dan terdiri dari 2 bagian utama, yakni elektromagnet (*coil*) dan mekanikal (seperangkat kontak saklar/*switch*). Pada dasarnya, relay terdiri dari 4 komponen dasar, yakni elektromagnet, *armature*, *switch contact point* (saklar), dan *spring*. Relay pun memiliki beberapa fungsi, diantaranya:

1. Digunakan untuk menjalankan fungsi logika (*Logic Function*)
2. Digunakan untuk memberikan fungsi penundaan waktu (*Time Delay Function*)
3. Digunakan untuk mengendalikan sirkuit tegangan tinggi dengan bantuan dari sinyal tegangan rendah
4. Berfungsi untuk melindungi motor atau komponen lainnya dari hubung singkat (*Short*).



Gambar 2.9 Struktur Sederhana *Relay*

2.13 Lampu *Rotary* Peringatan

Lampu *rotary* peringatan adalah lampu yang memberikan tanda peringatan akan bahaya suatu tempat dengan tujuan agar orang yang melintas atau lewat pada area tersebut akan lebih berhati-hati. Dengan model lampu yang berputar (rotary) yang mempunyai reflektor pada sisi lampu tersebut menarik perhatian orang untuk melihat, sering juga kita temui lampu tanda peringatan ini dipasang pada tempat keluar masuk kendaraan besar pada proyek yang sedang dibangun dan area-area kerja yang berbahaya.

Gambar 2.10 Lampu *Rotary* Peringatan

2.14 Telegram

Telegram adalah layanan perpesanan multiplatform yang didirikan oleh pengusaha Rusia Pavel Durov. Telegram diluncurkan untuk Android di Alpha pada 20 oktober 2013, dan sekarang memiliki sekitar 200 juta pengguna. Secara fungsionalitas Telegram sama dengan kebanyakan aplikasi perpesanan lainnya, pengguna dapat membuat percakapan grup, menelpon kontak, mengirim file dan stiker. Fitur utama dari aplikasi ini adalah privasi dengan menggunakan enkripsi *end-to-end*, sehingga mempersulit peretas untuk melihat pesan yang telah dikirm. Akan tetapi ada banyak perbedaan spesifik antara platform telegram dengan para aplikasi perpesanan lainnya, seperti contohnya:

a. *Secret chats* yang mengasilkan pesan terenkripsi *end-to-end*, fitur ini sangat bermanfaat untuk mengamankan privasi dari tindak kejahatan/penyadapan.
b. *Self-destruct timers* merupakan fitur penghapus pesan secara otomatis berdasarkan waktu yang telah pengguna tentukan sebelumnya.

c. *Global message deletion* merupakan salah satu fitur yang berfungsi untuk menghapus pesan secara global, untuk meminimalisir kesalahan pengiriman pesan maupun tulisan.

d. *Large file size limit* memungkinkan pengguna dapat mengirim file dengan ukuran 1.5GB yang dapat dikatakan cukup besar jika dibandingkan dengan kompetitor lainnya.

e. *Customizations,* jika dibandingkan dengan kompetitor lainnya telegram lebih mudah untuk melakukan kostumisasi tampilan seperti *template*, animasi dan UI.

`Telegram juga menyediakan sebuah API (*Application Programming Interface* atau Antarmuka Pemrograman Aplikasi) berupa Telegram Bot API yang memungkinkan siapa saja untuk membuat bot mereka sendiri, dimana Botnya nanti yang akan membalas pesan sesuai dengan program yang mereka buat.

2.15 *Confusion Matrix*

Confusion Matrix merupakan matriks yang menampilkan klasifikasi prediksi dan klasifikasi aktual. Pada dasarnya matriks ini berukuran L x L, dimana L merupakan jumlah nilai label yang berbeda[27]. Tabel matriksnya dapat dilihat seperti pada Tabel 2.1 berikut.

Tabel 2. 1 Confusion matrix

Prediksi / Aktual	Positif	Negatif
Positif	*True* +	*False* -
Negatif	*False* +	*True* -

Dengan menggunakan Tabel 2.1, menghasilkan nilai *Accuracy, Precision, Recall* atau Sensitivity, dan F-1 Score. Dimana keempat nilai tersebut memiliki pengertian sebagai berikut:

a. *Accuracy* menggambarkan seberapa akurat model dalam mengklasifikasi dengan benar, dan dapat diketahui dengan persamaan berikut.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (2.1)$$

b. *Precisiong* menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model, dapat diketahui dengan persamaan berikut.

$$Precision = \frac{TP}{TP+FP} \quad (2.2)$$

c. *Recall* atau *Sensitivity* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi, dapat diketahui dengan persamaan berikut.

$$Recall\ atau\ Sensitivity = \frac{TP}{TP+FN} \quad (2.3)$$

d. *F-Score* menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *Accuracy* tepat digunakan sebagai acuan perfomansi algoritma jika dataset yang ada memiliki jumlah data *False Negatif* dan *False Positif* yang sangat mendekati. Namun jika jumlahnya tidak mendekati, maka sebaiknya menggunakan F-1 Score.

$$F-1\ Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \quad (2.4)$$

2.16 Kajian Pustaka

Penelitian tentang Perancangan Sistem Pendeteksi Alat Pelindung Diri (APD) Untuk Pengawasan K3 Di Lingkungan Kerja Secara *Real Time* Berbasis Raspberry Pi 4 Dengan Notifikasi Telegram ini mengacu pada beberapa penelitian sebelumnya yang mana sangat perlu untuk ditinjau kembali untuk mengetahui hubungan antara penelitian terdahulu dan penelitian yang sedang dilakukan saat ini demi menghindari suatu duplikasi dalam penelitian yang dilakukan. Berikut ini merupakan tabel (2.1) referensi pada penelitian sebelumnya.

Tabel 2. 2 Daftar Judul Penelitian Terdahulu

No.	Judul Penelian	Penulis	Tahun
1.	Deteksi Alat Pelindung Diri Menggunakan Metode YOLO dan *Faster* R-CNN	Jonathan Adiwibowo, Kartika Gunadi, Endang Setyadi	2020

No.	Judul Penelian	Penulis	Tahun
2.	*Smart Face-shield* Berteknologi *Internet of Things* sebagai Alat Pelindung Diri di Era Pandemi Covid-19	Berlilana, Bagus Adhi Kusuma, Fikri Ramadhan	2021
3.	Rancang Sistem Pendeteksi Alat Pelindung Diri (APD) Berbasis *Image Prosessing* (2021	Miftachul Ulum, Muhammad Zakariya, Achmad Fiqhi I, Haryanto	2021
4.	Deteksi Penggunaan Alat Pelindung Diri (APD) Untuk Keselamatan dan Kesehatan Kerja Menggunakan Metode Mask Region Convolutional Neural Network (Mask R-CNN)	Milzamah Elvi Laily, Fathorazi Nur Fajri, Gulpi Qorik Oktagalu Pratamasunu	2022
5.	Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN	Rescky Marthen Mailoa, Leo Willyanto Santoso	2021

Berdasarkan (Tabel 2.1) dapat diuraikan sebagai berikut:

1. Penelitian pertama ini membahas tentang Deteksi Alat Pelindung Diri Menggunakan Metode YOLO Dan *Faster* R-CNN, yang mana pada penelitian ini menggunakan 2 (dua) buah metode yaitu YOLO (*You Only Look Once*) dan *Faster* R-CNN. Dimana pada metode YOLO yang diterapkan untuk bagian kepala saja mendapatkan tingkat akurasi sebesar 73.83% , sedangkan pada metode *Faster* R-CNN mendapatkan nilai rata-rata sebesar 93.61% untuk *accuracy*, 88.07% untuk precision, 82.37% untuk *Recall* dan 84.67% untuk F1-Score nya[11].
2. Penelitian kedua membahas tentang *Smart Face-shield* Berteknologi *Internet of Things* sebagai Alat Pelindung Diri di Era Pandemi Covid-19. Pada penelitian ini membahas tentang pembuatan alat yang berfungsi social

distancing reminder yang terintegrasi dalam sebuah APD yaitu *Face-Shield*, *Face-shield* ini memiliki kamera RGB, kamera termal dan sensor jarak. Alat ini akan mendeteksi seseorang yang suhu tubuhnya lebih dari 37,5 ^{0}C pada jarak 1 meter dan menghidupkan *Buzzer* dan mengirim data ke aplikasi *Blynk* untuk monitoring[28].

3. Penelitian ketiga membahas tentang Rancang Sistem Pendeteksi Alat Pelindung Diri (APD) Berbasis *Image Prosessing*. Pada penelitian ini menggunakan Arduino UNO sebagai mikrokontroller nya dan Webcam Logitech C270 sebagai perangkat pengambilan gambar nya, dimana cara kerja nya kamera akan menangkap gambar para pekerja yang tidak menggunakan APD lengkap akan mengirim data ke Arduino UNO dan kemudian mengirim sinyal ke Buzzer sebagai alarm. Penelitian ini hanya mendeteksi APD pada bagian kepala saja (seperti masker, helm *safety*, dan kaca mata), dimana penelitian ini menggunakan metode CNN dengan hasil akurasi sebesar 75%[5].
4. Penelitian keempat membahas tentang Deteksi Penggunaan Alat Pelindung Diri (APD) Untuk Keselamatan dan Kesehatan Kerja Menggunakan Metode *Mask Region Convolutional Neural Network* (*Mask* R-CNN). Pada penelitian ini dilakukan identifikasi APD diruang lingkup kerja terhadap helm dan rompi dengan menggunakan metode *Mask* R-CNN, dimana pengujian dilakukan dengan mengidentifikasi gambar yang tersedia di internet. Dan hasil yang didapatkan pada pengujian ini yaitu mendapat akurasi sebesar 95%, presisi 96%, dan sensitifitas 97%[29].
5. Dan penelitian kelima membahas tentang Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN. Penelitian ini melakukan pendeteksian pada bagian kepala (helm) dan badan (Vest/Rompi) dengan menggunakan metode YOLO dan CNN. Pengujian ini menunjukan akurasi yang cukup baik untuk setiap bagian tubuh yang dideteksi, dimana akurasi yang diperoleh untuk kepala adalah 64.09% sementara badan memperoleh nilai akurasi 63.03%[30].

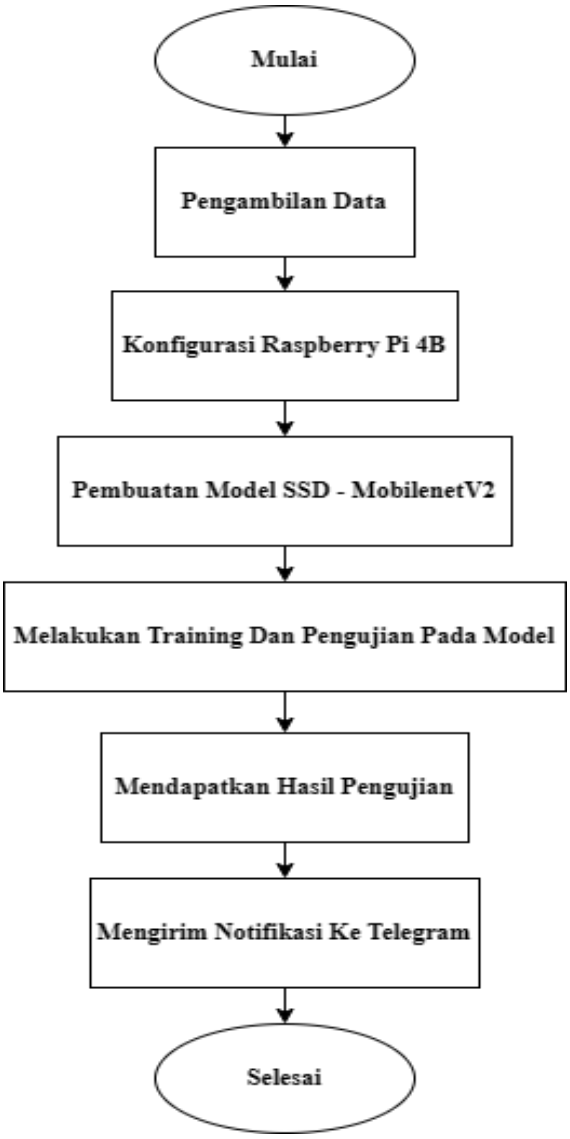
Berdasarkan beberapa penelitian terdahulu diatas, bisa disimpulkan bahwa metode yang digunakan cukup berat dijalankan pada perangkat mikro komputer.

Contohnya yaitu metode YOLO yang berat dan lama saat proses menjalankan nya di perangkat Raspberry Pi. Oleh sebab itu penelitian ini menggunakan metode yang lebih ringan namun dengan performa yang hampir sama baiknya untuk digunakan pada *device* yang tidak terlalu memumpuni.

BAB III
METODOLOGI PENELITIAN

3.1 Alur Penelitian

Penelitian ini merancang sistem *realtime* dalam mendeteksi pemakaian Alat Pelindung Diri (APD) yang dikenakan oleh pekerja di lapangan berupa helm, *wearpack*, dan sepatu *safety* kerja dengan menggunakan Raspberry Pi 4B. Data yang digunakan dalam penelitian ini berupa gambar tubuh pekerja diarea wajib Alat Pelindung Diri (APD), setiap orangnya diambil *sample* gambar dengan skenario penelitian yang dibuat. Alur penelitian dapat dilihat pada Gambar 3.1.

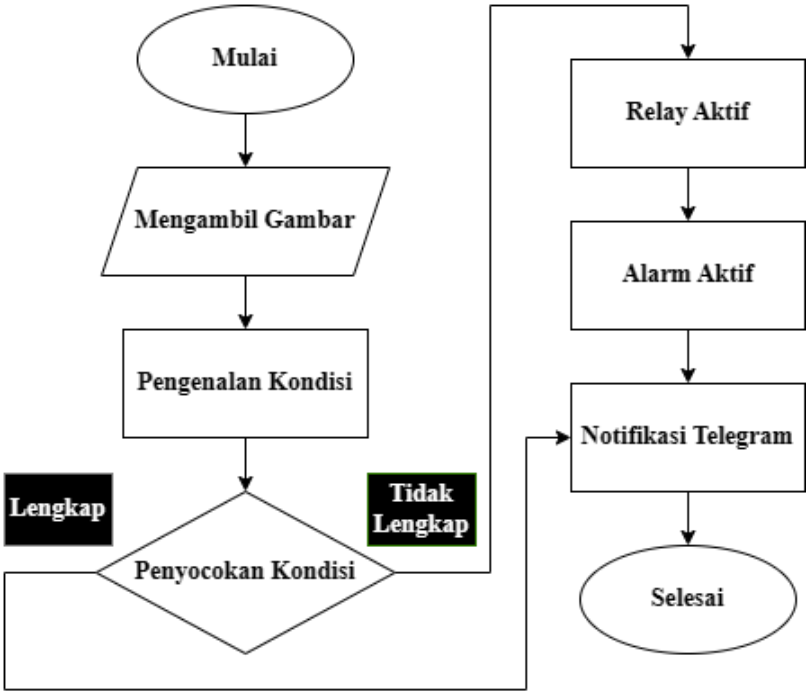


Gambar 3.1 *Flowchart* Alur Penelitian

Berdasarkan *flowchart* alur penelitian pada Gambar 3.1, maka dapat diketahui bahwa tahapan-tahapan yang digunakan untuk menyelesaikan penelitian tersebut adalah sebagai berikut:

1. Melakukan pengambilan data dengan menggunakan kamera webcam yang berupa gambar dari 6 kondisi penggunaan APD pada seseorang.
2. Melakukan konfigurasi pada Raspberry Pi 4B seperti menginstal perangkat lunak (*software*) dan *library* yang dibutuhkan agar dapat melakukan pendeteksian APD.
3. Membuat model deteksi masker dengan menggunakan metode SSD-MobileNetV2.
4. Melakukan pengujian terhadap program dan model yang telah dibuat menggunakan Raspberry Pi 4B.
5. Mendapatkan hasil pendeteksian meliputi 6 macam kondisi penggunaan APD
6. Mengirimkan notifikasi ke aplikasi Telegram berupa citra gambar yang ditangkap oleh kamera webcam dan berisi keterangan objek yang terdeteksi tersebut.

Selain terdapat alur penelitian, terdapat juga cara kerja dari perancangan alat ini sebagaimana yang dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2 *Flowchart* Cara Kerja Alat

3.2 Instrumen Penelitian

Instrumen yang digunakan dalam perancangan dan penelitian ini meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang dapat mendukung dan mempermudah kinerja dalam proses pengerjaan penelitian. Pada tabel 3.1 berikut ini adalah instrumen penelitian yang akan digunakan

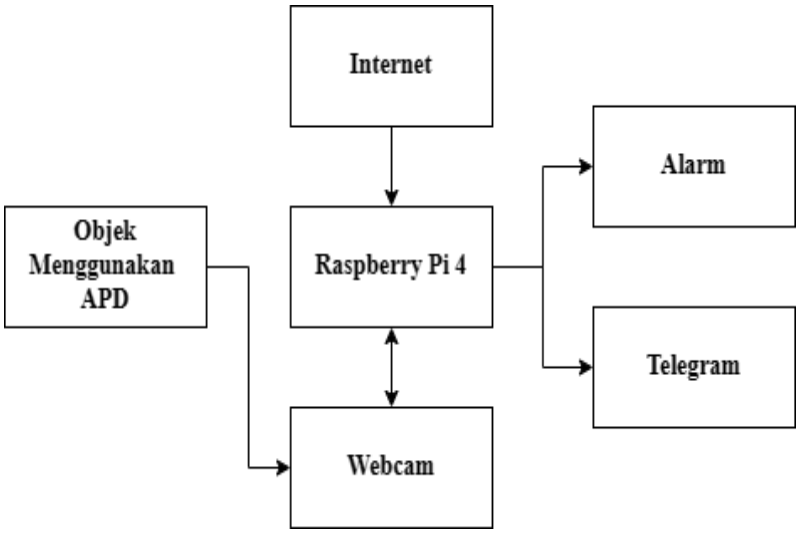
Tabel 3.1 Instrumen Penelitian

No.	Jenis	Instrumen	Aplikasi
1	*Hardware*	Raspberry Pi 4 Model B RAM 4 GB	*Microcomputer* yang berfungsi sebagai pusat kendali komponen-komponen yang digunakan, serta mengolah dan memproses gambar.
		Adapter Tipe C Output 5V 3A	Sumber tegangan Raspberry Pi
		Relay	Sebagai komponen yang mengontrol lampu rotary.
		Aki Motor dengan Output 12 V	Sebagai penyuplai daya ke lampu rotary.
		WebCam	Mengambilan gambar dan melakukan proses *image processing.*
		Speaker	Memberikan alarm/notifikasi
		Lampu rotary	Memberikan alarm

No.	Jenis	Instrumen	Aplikasi
		Monitor	Sebagai *output* tampilan dari Raspberry Pi.
2	*Software*	*Raspbian* OS 64-bit	Sistem operasi berbasis *Debian* yang digunakan untuk menjalankan Raspberry Pi 4.
		Python IDE	Membuat listing program, lalu menyusun dan mengunggah program pada Raspberry Pi 4.
		Telegram	Sebagai pusat notifikasi pendeteksi APD untuk pemantauan petugas K3.

3.3 Diagram Blok Penelitian

Diagram blok penelitian ini akan menjelaskan alur proses penelitian yang dimulai dari menangkap gambar seseorang yang memakai APD menggunakan webcam, kemudian melakukan proses pendeteksian di Raspberry Pi 4, kemudian gambar yang telah terdeteksi akan diteruskan oleh Raspberry Pi untuk mengirim perintah kepada alarm apabila tidak memenuhi syarat, dan selanjutnya akan mengirim hasil dari pendeteksian dikirimkan ke Telegram. Diagram blok ini dapat dilihat pada Gambar 3.3 berikut.



Gambar 3.3 Diagram Blok Penelitian

3.4 Menentukan Jenis Kondisi

Menentukan jenis kondisi ini bertujuan untuk melakukan proses pendeteksian. Terdapat 6 jenis kondisi yang sudah ditentukan, diantaranya yaitu 1) Memakai APD lengkap, 2) Tidak memakai helm safety, 3) Tidak memakai wearpack safety, 4) Tidak memakai sepatu safety, 5) Lebih dari satu individu (APD lengkap) dan 6) Lebih dari satu individu (salah satu APD tidak lengkap).

3.5 Menentukan Media

Media bertujuan untuk lokasi pengambilan data latih dan lokasi untuk melakukan pengujiannya. Pada penelitian ini media yang digunakan adalah gerbang masuk menuju *switch yard*. Lokasi tersebut dipilih karena area tersebut merupakan batas aman atau batas area wajib menggunakan APD di lingkungan Gardu Induk Transmisi. Lokasi pengambilan data dan pengujian alat dapat dilihat pada Gambar 3.4 berikut ini.

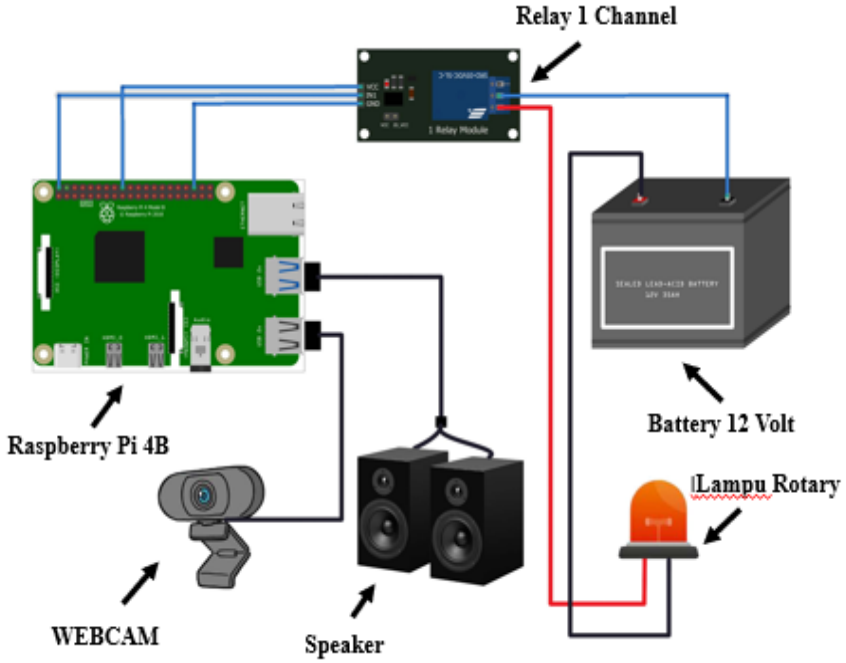
Gambar 3.4 Lokasi Penempatan Kamera Webcam

3.6 Perancangan *Hardware*

Perancangan perangkat keras berisikan tentang perancangan instrumentasi alat, implementasi komponen-komponen sampai dengan konfigurasi komponen-komponen yang mendukung.

3.6.1 Perancangan Elektronika

Perancangan ini menjelaskan mengenai pemasangan atau perakitan komponen-komponen dalam pembuatan sistem. Dalam perancangan elektronika penelitian ini berisikan mikrokomputer Raspberry Pi 4 yang terhubung dengan *Speaker* dan *Relay*, yang mana *Relay* tersebut terhubung dengan lampu rotary sebagai alarm dan aki motor sebagai pemberi tegangan ke lampu rotary tersebut. Adapun skema lengkap perancangan alat yang dibuat, agar lebih jelas dalam konfigurasi setiap komponennya yang bisa dilihat pada gambar 3.5 berikut ini.



Gambar 3.5 Skema Perencanaan Alat

Berdasarkan Gambar 3.5 pin-pin yang ada pada Raspberry Pi terhubung secara langsung dengan komponen yang digunakan, dan Raspberry Pi itu sendiri menjadi pusat control untuk mengendalikan semua komponen yang digunakan. Aturan perintah-perintah yang terdapat pada pin diatur langsung menggunakan bahasa pemrograman.

3.6.2 Konfigurasi Raspberry Pi

Raspberry Pi harus terlebih dahulu dikonfigurasi agar program yang dibuat dapat dijalankan dengan baik. Pada penelitian ini Raspberry Pi 4 model B

merupakan bintang utama, yang mana berperan sebagai pusat kendali seluruh komponen yang digunakan. Dan sebagai pusat kendali, Raspberry Pi bertugas untuk mengolah data yang didapat oleh webcam, mengaktifkan speaker dan lampu rotary sebagai alarm jika data yang diterima sudah sesuai, dan kemudian mengirimkan notifikasi ke telegram berupa tangkapan gambar dari kondisi yang sudah ditentukan. Adapun pin-pin Raspberry Pi 4 yang telah dikonfigurasi dapat dilihat pada Tabel 3.2 berikut ini.

Tabel 3.2 Konfigurasi Pin Raspberry Pi 4B

NO	PIN	FUNGSI
1.	GPIO 24	Terhubung dengan pin *signal relay*
2.	5 V	Terhubung dengan pin *power relay*
3.	GROUND	Terhubung dengan pin *ground relay*

3.6.3 Konfigurasi *WEBCAM*

Konfigurasi Webcam bertujuan supaya gambar yang tangkap oleh Webcam sesuai dengan yang diinginkan, dikarenakan webcam yang digunakan ini sudah mendukung fitur *plug and play* maka kamera yang telah terhubung bisa langsung digunakan. Letak posisi Webcam saat pengambilan data latih sama dengan posisi saat pengujian. Hasil gambar dari Webcam dapat dilihat pada Gambar

3.7 Perancangan *Software*

Apabila perancangan *hardware* telah selesai, maka dilanjutkan dengan perancangan *software* nya. Pada perancangan software ini terdiri atas perancangan *software* pada Raspberry Pi dan aplikasi telegram. Dan pada tahap ini dilakukan proses pengambilan data latih, *pre-processing training data*, dan training data.

3.7.1 Mengambil Data Latih

Data latih ini diperlukan untuk membangun model SSD-MobilenetV2 dari jenis kondisi yang sudah ditentukan. Data latih yang diambil berupa video manusia dengan 6 jenis kondisi yang telah ditentukan sebelumnya. Setelah video didapat, kemudian diambil gambarnya dengan cara *screenshoot* foto menggunakan aplikasi

pemutar video. Untuk memperbanyak data latih, dibutuhkan juga pengambilan data yang tersedia di internet, contohnya pada *website* Kaggle.

3.7.2 Menentukan Model Tensorflow Lite

Tensorflow Lite merupakan *interface* yang gunanya untuk mengeksekusi perintah dengan memanfaatkan informasi yang dimiliki seperti berupa objek yang ingin dikenali. Tensorflow Lite memiliki banyak model, sehingga harus memilih model yang sesuai dengan hardware supaya mendapatkan hasil yang diinginkan. Pada penelitian ini, model yang digunakan ialah SSDlite-MobilenetV2. Model tersebut dipilih karena dalam beberapa penelitian merupakan model yang paling sesuai dalam hal keakuratan dan kecepatan dalam mendeteksi objek.

3.7.3 *Pre-Processing Training* Data

Pre-processing training data merupakan proses manipulasi *dataset* sebelum dimasukan kedalam model, dengan kata lain tahap ini bertujuan untuk membangun model SSD-MobilenetV2. Tahap ini terbilang proses yang penting untuk persiapan sebelum melakukan training data. Tujuan lainnya yaitu agar kompatibel dengan *library* yang digunakan, seperti halnya tensorflow yang memerlukan masukan tensor. Dan tahap-tahap berikut ini menjelaskan data latih yang telah diambil kemudian diproses menjadi model yang memiliki ekstensi Tensorflow Lite.

3.7.3.1 Pelabelan Gambar

Pelabelan gambar ini bertujuan untuk mendapatkan hasil dengan format `.xml`, dan proses ini menggunakan aplikasi LabelImg. Dalam proses pelabelannya yaitu dengan membuat kotak pada wajah, lalu menamainya sesuai dengan jenis kondisi yang telah ditentukan dan kemudian disimpan. Pelabelan gambar ini kemudian akan menghasilkan file dengan format `.xml`, dimana file tersebut berisi jenis kondisi wajah yang sudah dinamai. Jenis kondisi yang digunakan diantaranya yaitu Memakai APD lengkap, Tidak memakai helm safety, Tidak memakai wearpack safety, Tidak memakai sepatu safety, Lebih dari satu individu (APD lengkap) dan Lebih dari satu individu (salah satu APD tidak lengkap).

3.7.3.2 Konversi XML ke CSV

Proses konversi XML ke CSV perlu dilakukan setelah proses pelabelan pada aplikasi LabelImg sudah selesai, hal itu bertujuan untuk menghasilkan berkas TFRecord yang mana untuk menghasilkan berkas tersebut perlu menggunakan format `.csv`. Dan perlu diketahui juga bahwa isi dari format `.csv` merupakan semua *file* berformat `.xml` yang telah dijadikan satu, didalamnya berisi rincian nama yang telah diberikan pada proses *labeling*.

3.7.3.3 Membuat *File Record*

File Record pada tensorflow menghasilkan suatu format yang bernama TFRecord, dan format TFRecord itu juga merupakan komponen pada tensorflow yang bertugas untuk men-generate data dari konversi file `.csv` yang telah dilakukan menjadi data kedalam bentuk biner. Tensorflow akan membaca data *input* dalam proses yang dinamakan *feeding* data pada saat *training* dilakukan. Proses *feeding* data tersebut secara langsung akan mengambil informasi dari *dataset* yang telah disiapkan dalam bentuk format TFrecord itu tadi.

3.7.3.4 Membuat Label Map

Label map merupakan proses pelabelan terhadap objek/citra, yang mana berguna untuk mengkonfigurasi pemetaan label dalam memberikan penamaan terhadap objek atau citra yang akan dideteksi. Didalam isi dari label map tersebut juga haruslah sesuai dengan objek/citra yang dideteksi.

3.7.3.5 Konfigurasi *Pipeline Config*

Konfigurasi *pipeline config* ini menggunakan berkas dari protobuf, dimana file konfigurasinya dibagi menjadi 5 bagian, yaitu:

a. Model: Untuk menentukan jenis model. Model yang dipilih ialah SSD (*Single Shot Detector*).
b. *Training Config*: Untuk memutuskan parameter apa yang harus digunakan untuk melatih parameter model.
c. *Train input reader*: Untuk mendefinisikan dataset *train* yang dilatih dengan menyertakan lokasi datasetnya.

d. *Eval input config*: Untuk mendefinisikan dataset *eval* yang dilatih dengan menyertakan lokasi datasetnya.
e. *Eval input reader*: Untuk mendefinisikan dataset apa yang dievaluasi modelnya.

3.7.3.6 Mengunggah ke Google Drive

Langkah pengunggahan ini dilakukan karena data untuk proses *training* menggunakan Google Colab, dan perlu dilakukan penyusunan file sebelum mengunggahnya ke Google Drive guna memudahkan proses *upload* file. Folder data latih perlu dikompresi dalam bentuk `.zip` sebelum di *upload*. Dalam folder `.zip` tersebut berisi dokumen data latih beserta file konfigurasi yang digunakan untuk mendukung proses *training* data menggunakan Google Colab.

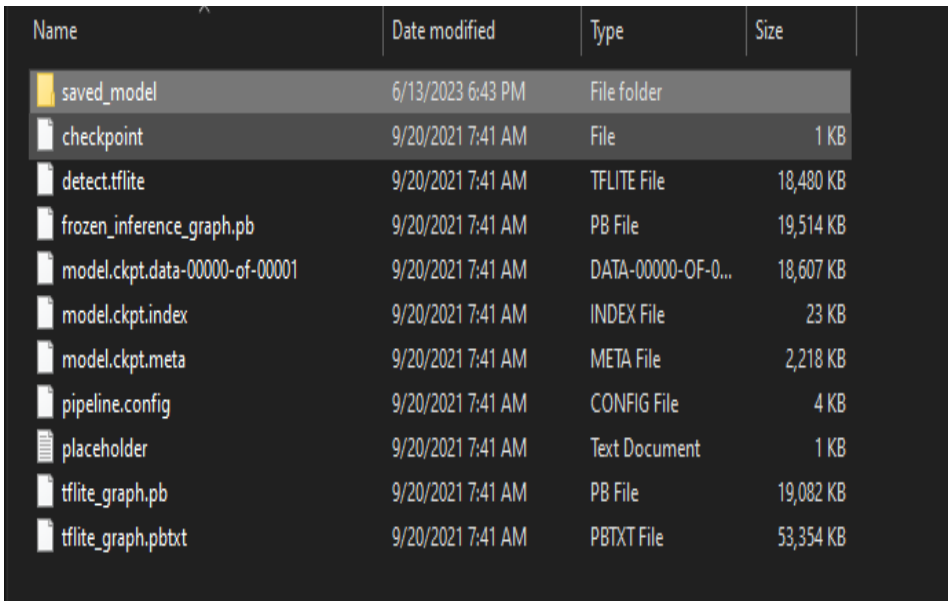
3.7.4 *Training* Data

Training data ini dilakukan setelah langkah *pre-processing training* datanya selesai. Pada proses training data ini, prosesnya akan dilakukan dengan memanfaatkan Google Colab karena terdapat super GPU (*Graphics Processing Unit*), sehingga dapat mempercepat proses *training* data. Dalam melaksanakan proses *training* data, ada beberapa tahapan diantaranya yaitu:

1. Melakukan Input dan informasi dari dataset yang telah disiapkan, dibuat dalam format file TFRecord.
2. Melakukan operasi konvolusi pada layer konvolusi dengan mengkombinasikan linier filter terhadap daerah lokal. Bentuk layernya berupa sebuah filter dengan p, l, t dimana ketiga filter tersebut bergeser keseluruh bagian gambar dengan dimasukkan ke channel image. Penggeseran tersebut menghasilkan output yang disebut sebagai feature map.
3. Proses selanjutnya yang bertujuan untuk menghasilkan non-linearitas, langkah ini disebut *Activation* ReLu . Cara kerjanya ialah dengan mengubah nilai yang kurang dari 0 menjadi 0, sehingga pada indeks gambarnya tidak ada yang bernilai minus.
4. Selanjutnya yaitu *pooling layer*, dimana *layer* ini yang menerima *output* dari konvolusi *layer*. *Pooling layer* sendiri terdiri dari *filter* dengan ukuran

tertentu dan kemudian bergeser keseluruh area *feature* map. *Output* dari *layer* ini masih berupa *multidimension array*, sehingga harus dilakukan *reshape* supaya menjadi sebuah vektor dan nantinya bisa digunakan sebagai *input* dari *fully connected layer*.

5. Setelah *pooling layer* yaitu *fully connected layer*. Langkah ini terdapat proses *pixel* yang menghubungkan konvolusi dan *pooling* dimana yang dianggap sebagai wajah menjadi sebuah output yang terdiri dari satu kelas label. Hasil proses tersebut menentukan bagian mana yang memiliki pola wajah.
6. Hasil dari proses *training* data berupa *checkpoint* yang dibuat secara otomatis oleh tensorflow dalam bentuk graph tensor. Hal tersebut bertujuan untuk menyimpan informasi proses *training* data yang dilakukan dengan format *file* .ckpt. Kemudian file tersebut di export ke dalam format *file* .pb. Setelah di ekspor, proses *training* data yang dilakukan mendapatkan hasil seperti pada Gambar 3.6 berikut ini.

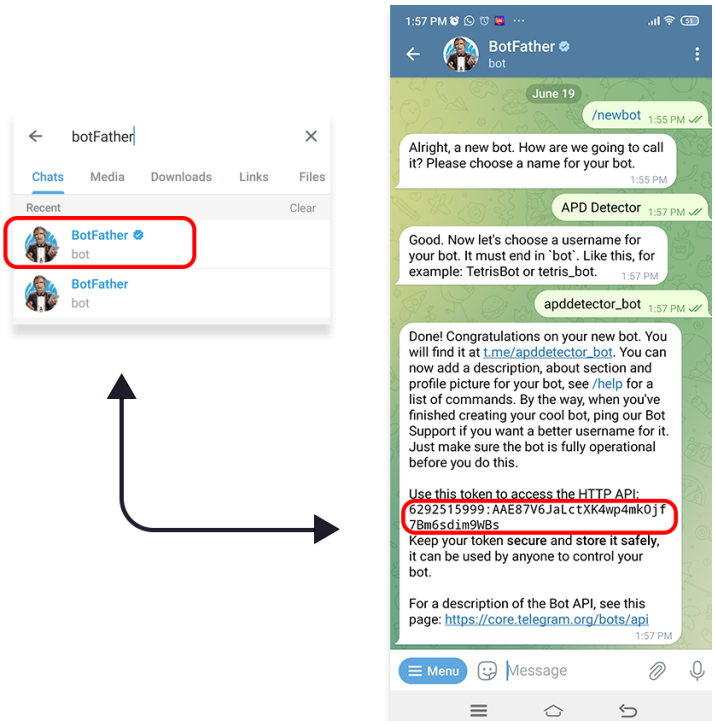


Gambar 3.6 Hasil *Training* Data

Berdassarkan Gambar 3.6 dimana yang dihasilkan setelah proses *training* data menghasilkan cukup banyak *file*, adapun *file* yang digunakan sebagai parameter pengenalan wajah memiliki format .tflite. *File* tersebut akan disimpan bersamaan dengan direktori pemrograman untuk memudahkan dalam pencarian *file* pada saat memprogram sistem.

3.7.5 Pembuatan Bot Telegram

Telegram digunakan untuk memberikan notifikasi kepada pengawas K3 ketika terdapat pekerja yang lengkap atau tidak lengkap dalam penggunaan APD saat pendeteksian dilakukan. Bentuk dari notifikasi yang didapatkan oleh pengawas K3 berupa pemberitahuan berbentuk tangkapan gambar pekerja saat hendak memasuki *Swicth Yard*. Dalam penyampaian notifikasinya, Telegram dipilih dan digunakan karena memiliki berbagai kemudahan untuk pengimplementasian sistem IoT, dimana telegram menyediakan *Application Program Interface* yang dapat diprogram dengan menggunakan bahasa pemrograman. Pembuatan bot telegram ini menggunakan bot @BotFather yang ada di aplikasi Telegram yang tampilannya seperti Gambar 3.7 berikut ini.



Gambar 3.7 Pembuatan Bot Telegram

Berdasarkan Gambar 3.7 ada beberapa langkah dalam pembuatan bot telegram nya seperti berikut:

1. Membuka aplikasi Telegram.
2. Mencari dan menambahkan bot @BotFather.
3. Ketik `/start` pada kolom *chat* botnya.

4. Ketik `/newbot` untuk membuat bot baru, setelah itu memasukkan nama dan *username* bot yang ingin digunakan.
5. Pembuatan bot selesai ketika sudah mendapatkan token dimana token tersebut digunakan untuk mengakses ke HTTP API.

3.8 Tempat Dan Waktu Penelitian

Tempat penelitian dilakukan di PT. PLN (Persero) Gardu Induk 150 KV Lengkong. Waktu Pengerjaan Skripsi ini berlangsung dari bulan Oktober 2021.

BAB IV
HASIL DAN PEMBAHASAN

Penelitian ini membahas sistem pendeteksi alat pelindung diri untuk pengawasan K3 di lingkungan kerja, guna meminimalisir dan mencegah terjadinya kecelakaan kerja yang diakibatkan oleh faktor ketidakdisiplinan manusia itu sendiri.

4.1 Hasil Pengambilan Data Latih

Data latih yang didapatkan berupa video yang direkam menggunakan webcam. Video yang telah didapat tersebut kemudian akan diambil kembali berupa potongan gambar (*screenshoot*) dari orang-orang yang menggunakan beberapa kondisi APD menggunakan *software* pemutar video. Pengambilan data latih sama dengan pengujian yaitu di gerbang PT. PLN (Persero) Gardu Induk 150 KV Lengkong, lebih tepatnya gerbang masuk *switch yard* yang mana merupakan batas aman atau batas area wajib menggunakan APD di lingkungan Gardu Induk Transmisi. Kemudian setelah proses pengambilan data latih, dilanjutkan dengan proses pengolahan gambar selanjutnya seperti labelling, augmentasi, dan serangkaian proses lainnya agar bisa diproses dengan menggunakan bahasa pemograman.

Hasil pengolahan gambar yang dilakukan, proses labeling menggunakan program bernama *labelImg* yang diinstal menggunakan *command* bahasa *python*. Proses *labelling* ini akan mengjhasilkan file dengan ekstensi `.xml`, file tersebut digunakan untuk mengelompokan deskripsi suatu citra yang nantinya akan dikonfigurasi menjadi file yang siap untuk *training* data. Adapun total jumlah data latih yang dihasilkan dapat dilihat pada (Tabel 4.1) berikut:

Tabel 4.1 Jumlah Data Latih

KONDISI	JUMLAH
APD Lengkap	268
Tidak memakai helm safety	159
Tidak memakai wearpack safety	146

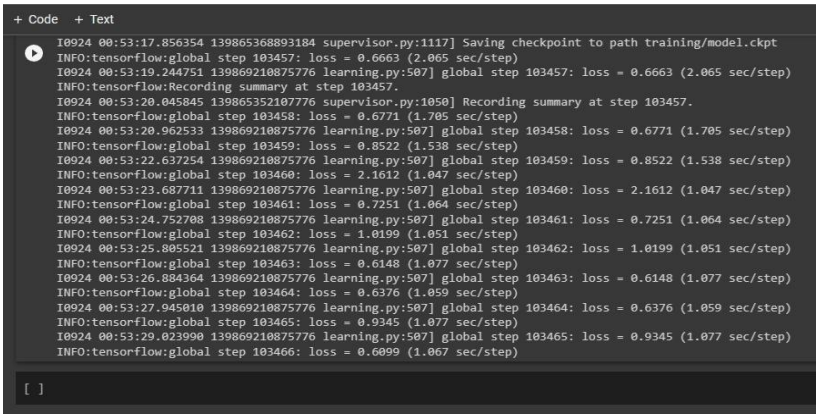
KONDISI	JUMLAH
Tidak memakai sepatu safety	138
Lebih dari satu individu (APD lengkap)	231
Lebih dari satu individu (salah satu APD tidak lengkap)	129
JUMLAH	1071

Berdasarkan (Tabel 4.1) jumlah data diambil pada tiap kondisi yang berbeda-beda berdasarkan hasil pengambilan gambar dan proses augmentasi. Jumlah data latih yang diambil mempengaruhi akurasi sistem ketika digunakan, semakin baik dan banyak data latih yang digunakan, maka akurasi sistem akan semakin baik juga.

4.2 Hasil Training Data

Training data ini bertujuan agar sistem yang dibuat dapat melakukan pendeteksian APD, meminimalisir terjadinya kesalahan saat pendeteksian dan mendapatkan hasil akurasi yang bagus. Data yang digunakan dalam proses *training* ini berasal dari data latih yang telah diolah melalui beberapa tahapan mulai dari melakukan *resizing* atau merubah ukuran resolusinya ke menjadi 300x300 *pixel*, anotasi gambar, konversi xml ke csv, membuat *file record*, dan membuat label map, setelah itu *upload* file-file tersebut ke Google Drive.

Data yang sudah diolah kemudian akan dilakukan *training* dengan menggunakan bantuan dari Google Colaboratory. Google Colaboratory digunakan karena terdapat fitur super GPU (*Graphics Processing Unit*) yang membuat proses training menjadi cepat. Dalam proses *training* data dengan model SSD-MobileNetV2 membutuhkan waktu rata-rata 1 detik setiap 1 *step*. Penelitian ini membutuhkan 103466 *step*, maka waktu yang diperlukan untuk 103466 *step* kurang lebih yaitu sekitar 29 jam. Proses *training* dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 *Training* Data Menggunakan Google Colaboratory

Berdasarkan Gambar 4.1, proses *training* menggunakan Google Colab Pro sudah mencapai 103466 *step*, terdapat parameter *loss* yang apabila nilainya semakin kecil maka proses *training* disetiap *step* semakin akurat. Durasi proses *training* data per-*step* berbeda-beda sesuai dengan kondisinya.

4.3 Pengujian Pendeteksi Alat Pelindung Diri

Pengujian pendeteksi APD ini dilakukan yaitu untuk mengetahui dan menghitung tingkat akurasi dari sistem yang telah dibuat itu sendiri. Dalam pengujian ini keberhasilan nya dipengaruhi oleh beberapa faktor seperti jumlah *dataset* yang dimiliki, pencahayaan di lokasi pengujian alat ini dilakukan, serta hasil resolusi gambar yang di *capture* oleh kamera. Dari pengujian-pengujian yang telah dilakukan, telah didapatkan hasil seperti yang dijelaskan pada point-point berikut ini.

4.3.1 Pengujian Pada Waktu Siang Hari

Pengujian yang pertama dilakukan pada waktu siang hari yaitu pada rentang waktu jam 2 sampai dengan jam 3 siang, dimana pada rentang waktu tersebut didapati lux (pencahayaan) di area pengujian yaitu sebesar 13302 lux. Dimana telah didapatkan hasil tangkapan gambar objek yang dilakukan oleh Webcam dan dirangkum pada tabel 4.2 berikut ini.

Tabel 4.2 Hasil Pengujian Pada Siang Hari

Kondisi	Jumlah Pengujian	Hasil Pengujian
APD Lengkap	1	Lengkap
	2	Lengkap
	3	Lengkap
	4	Lengkap
	5	Lengkap
Tidak Memakai Helm Safety	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Lengkap
Tidak Memakai Waerpack Safety	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Lengkap
Tidak Memakai Sepatu Safety	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Lengkap
Lebih Dari Satu Individu (APD Lengkap)	1	Lengkap
	2	Tidak Lengkap
	3	Lengkap
	4	Lengkap
	5	Lengkap
Lebih dari satu individu (salah satu APD tidak lengkap)	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Lengkap

Berdasarkan (Tabel 4.2) menunjukan hasil pengujian terhadap 6 sample kondisi berbeda yang telah ditentukan dengan melakukan pengujian sebanyak 5 kali tiap samplenya.

4.3.2 Pengujian Pada Waktu Sore Hari

Pengujian yang kedua dilakukan pada waktu sore hari yaitu pada rentang waktu jam 5 sampai dengan jam 6 sore. Dari pengujian yang dilakukan pada sore hari yang mana intensitas sinar matahari yang mulai menurun pada saat itu, didapati lux (pencahayaan) di area pengujian yaitu sebesar 95 lux. Dimana telah didapatkan hasil tangkapan gambar objek yang dilakukan oleh Webcam dan dirangkum pada pada tabel 4.3 berikut ini.

Tabel 4.3 Hasil Pengujian Pada Sore Hari

Kondisi	Jumlah Pengujian	Hasil Pengujian
APD Lengkap	1	Lengkap
	2	Lengkap
	3	Lengkap
	4	Lengkap
	5	Lengkap
Tidak Memakai Helm Safety	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Lengkap
Tidak Memakai Waerpack Safety	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Lengkap
	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap

Kondisi	Jumlah Pengujian	Hasil Pengujian
Tidak Memakai Sepatu Safety	4	Lengkap
	5	Tidak Lengkap
Lebih Dari Satu Individu (APD Lengkap)	1	Lengkap
	2	Lengkap
	3	Tidak Lengkap
	4	Lengkap
	5	Lengkap
Lebih dari satu individu (salah satu APD tidak lengkap)	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Lengkap

Berdasarkan (Tabel 4.3) menunjukan hasil pengujian terhadap 6 sample kondisi berbeda yang telah ditentukan dengan melakukan pengujian sebanyak 5 kali tiap samplenya.

4.3.3 Pengujian Pada Waktu Malam Hari

Terakhir yaitu pengujian yang ketiga dilakukan pada waktu malam hari yaitu pada rentang waktu jam 7 sampai dengan jam 8 malam. Dari pengujian yang dilakukan pada malam hari yang mana sudah tidak ada sinar matahari pada saat itu dan hanya mengandalkan cahaya yang ada di sekitar area pengujian pada saat itu, didapati lux (pencahayaan) di area pengujian yaitu hanya sebesar 1 lux. Dimana telah didapatkan hasil tangkapan gambar objek yang dilakukan oleh Webcam dan dirangkum pada tabel 4.4 berikut ini.

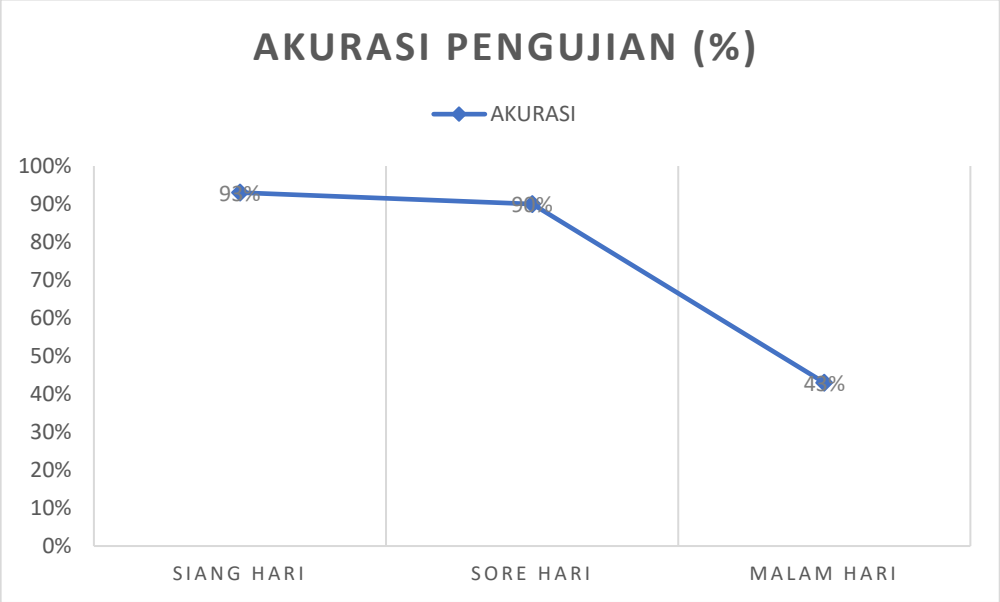
Tabel 4. 4 Hasil Pengujian Pada Malam Hari

Kondisi	Jumlah Pengujian	Hasil Pengujian
	1	Tidak Lengkap
	2	Tidak Lengkap

Kondisi	Jumlah Pengujian	Hasil Pengujian
APD Lengkap	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Lengkap
Tidak Memakai Helm Safety	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Lengkap
Tidak Memakai Waerpack Safety	1	Tidak Lengkap
	2	Tidak Terdeteksi
	3	Tidak Terdeteksi
	4	Tidak Lengkap
	5	Tidak Lengkap
Tidak Memakai Sepatu Safety	1	Tidak Terdeteksi
	2	Tidak Terdeteksi
	3	Tidak Lengkap
	4	Tidak Terdeteksi
	5	Tidak Lengkap
Lebih Dari Satu Individu (APD Lengkap)	1	Tidak Lengkap
	2	Tidak Lengkap
	3	Tidak Terdeteksi
	4	Tidak Lengkap
	5	Tidak Terdeteksi
Lebih dari satu individu (salah satu APD tidak lengkap)	1	Tidak Lengkap
	2	Tidak Terdeteksi
	3	Tidak Lengkap
	4	Tidak Lengkap
	5	Tidak Terdeteksi

Berdasarkan (Tabel 4.4) menunjukan hasil pengujian terhadap 6 sample kondisi berbeda yang telah ditentukan dengan melakukan pengujian sebanyak 5 kali tiap samplenya, dan dapat dilihat dimana pada pengujian malam hari ini ada beberapa objek yang tidak terdeteksi oleh sistem dikarenakan buruknya

pencahayaan di area pengujian. Setelah dilakukannya 3 perbedaan waktu pengujian alat tersebut dilakukan, maka dapat rangkum kedalam sebuah grafik perbandingan akurasi pendeteksian APD pada Gambar 4.2 berikut ini.



Gambar 4.2 Grafik Perbandingan Akurasi Pengujian

Berdasarkan (Gambar 4.2) hasil pengujian yang tertera pada (Tabel 4.2) (Tabel 4.3) dan (Tabel 4.4) yang telah dirangkum, dapat diketahui bahwa persentase keakuratan sistem (As) untuk objek yang telah berhasil dideteksi dapat dihitung dengan menggunakan rumus pada (Lampiran B). Dari hasil pengujian yang telah dilakukan didapatkan keakuratan sistem (As) sebesar 93 % pada siang hari, 90 % pada sore hari, dan 43 % pada malam hari (Lampiran C). Tingkat akurasi yang didapatkan cukup bervariasi pada 3 kondisi waktu pengujian dipengaruhi oleh intensitas cahaya yang berbeda, dimana bisa dilihat pada pengujian di malam hari mendapatkan hasil akurasi yang sangat buruk sekali karena sistem tidak mendeteksi objek dalam pengujian disebabkan hilangnya sumber penerangan dari matahari, dan penerangan di area pengujian dari bangunan terdekat sangat minim.

4.3.4 Hasil Akurasi Sistem Secara Keseluruhan

Pengujian akurasi sistem secara keseluruhan ini dilakukan untuk mencari garis besar dari hasil pengujian pada penelitian ini, yaitu dengan menggabungkan seluruh data pada (Tabel 4.2) (Tabel 4.3) dan (Tabel 4.4). Data yang sudah digabungkan tersebut kemudian dianalisa menggunakan metode *Confusion Matrix*,

guna mendapatkan nilai (meliputi *Accuracy, Presicion, Recall/Sensitivity,* dan F-1 Score) dari model yang dibuat itu sendiri. Hasil keseluruhan dari 3 waktu pengujian tersebut dirangkum pada Tabel 4.5 berikut ini.

Tabel 4.5 *Confusion Matrix* Pengujian Sistem Secara Keseluruhan

n = 90	Lengkap	Tidak Lengkap
Lengkap	TP = 18	FP = 12
Tidak Lengkap	FN = 10	TN = 50

Berdasarkan data dari (Tabel 4.5) yang kemudian diselesaikan dengan (Persamaan 2.1) (Persamaan 2.2) (Persamaan 2.3) dan (Persamaan 2.4) yang ada pada metode *Confusion Matrix*. Dan didapatkan nilai *Accuracy* sebesar 76 %, nilai *Precision* sebesar 60 %, nilai *Recall* atau *Sensitivity* sebesar 64 %, dan nilai F-1 Score sebesar 61 %. Perhitungan ada pada (Lampiran C)

4.4 Pengujian Pengiriman Data Ke Telegram

Pengujian pengiriman data ke telegram ini dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan Raspberry Pi mengirimkan data ke Telegram, dan dalam menghitung atau mengukur durasi proses pengiriman data nya digunakan *stopwatch* untuk mengetahui berapa lama data sampai masuk dan diterima ke Telegram. Proses perhitungan atau pengukuran durasi ini dimulai ketika kamera pertama kali mendeteksi objek sampai kemudian pesan notifikasi masuk ke Telegram. Notifikasi yang diterima oleh Bot Telegram berupa tangkapan gambar dan pesan teks seperti pada Gambar 4.3 berikut ini.

Gambar 4. 3 Hasil Notifikasi Yang Diterima Bot Telegram

Berdasarkan Gambar 4.3, notifikasi berupa pesan teks dan tangkapan gambar yang mengakses sistem. Pesan teks yang muncul menjelaskan keterangan dari objek yang dideteksi seperti (APD Lengkap apabila objek yang dideteksi memakai APD lengkap dan APD Tidak Lengkap apabila objek yang dideteksi tidak memakai APD lengkap). Waktu yang dibutuhkan sistem dalam mengirim notifikasi ke Telegram kemudian dirangkum kedalam Tabel 4.6 berikut ini.

Tabel 4. 6 Hasil Pengujian Waktu Pengiriman Data Ke Telegram

Kondisi	Pesan ke Telegram	Waktu Pengiriman (Detik)
APD Lengkap	Terkirim	3,77
	Terkirim	3,81
	Terkirim	3,04
	Terkirim	3,22
	Terkirim	3,76
Tidak Memakai Helm Safety	Terkirim	3,91
	Terkirim	3,43
	Terkirim	3,56
	Terkirim	3,64
	Terkirim	3,52
Tidak Memakai Waerpack Safety	Terkirim	3,12
	Terkirim	3,58
	Terkirim	3,41
	Terkirim	3,25
	Terkirim	3,17
Tidak Memakai Sepatu Safety	Terkirim	3,46
	Terkirim	3,19
	Terkirim	3,70
	Terkirim	3,68
	Terkirim	3,29
	Terkirim	3,91
	Terkirim	3,34

Kondisi	Pesan ke Telegram	Waktu Pengiriman (Detik)
Lebih Dari Satu Individu (APD Lengkap)	Terkirim	3,17
	Terkirim	3,80
	Terkirim	3,57
Lebih dari satu individu (Salah Satu APD lengkap)	Terkirim	3,61
	Terkirim	4,23
	Terkirim	3,15
	Terkirim	3,97
	Terkirim	3,83
Rata-rata Waktu Pengiriman		3,53

Berdasarkan Tabel 4.5, dapat diketahui bahwa waktu yang butuhkan oleh sistem dari 30 objek yang berhasil dideteksi dan diterima oleh Telegram paling cepat dengan waktu 3,04 detik dan paling lama 4,23 detik dengan rata-rata waktu sebesar 3,53 detik. Adanya *delay* waktu percobaan bisa terjadi karena adanya proses pendeteksian sistem untuk melakukan penyocokan terlebih dahulu terhadap objek untuk dikenali, sedangkan perbedaan waktu antar satu dengan yang lain bisa terjadi karena perbedaan resolusi objek yang ditangkap. Pengiriman data ke Telegram juga dipengaruhi oleh kecepatan internet yang digunakan Raspberry Pi 4, serta ukuran *file* yang dikirim.

4.5 Pembahasan Hasil Pengujian

Pembahasan ini bertujuan untuk memvalidasi hasil penelitian dengan teori-teori yang menjadi acuan dalam pembuatan penelitian ini. Dimana hasil pada penelitian sistem pendeteksi APD ini bisa dikatakan baik dan memiliki keunggulan yang tidak dimiliki oleh penelitian-penelitian sebelumnya yang dirangkum pada (Tabel 2.1), dan dapat disimpulkan penelitian ini memiliki keunggulan yang jelas mencolok yaitu mengimplementasikan sistem *Internet of Think* (IoT) dengan memanfaatkan aplikasi Telegram sebagai penerima notifikasi dari hasil pengujian APD yang dilakukan. Pengimplementasian aplikasi telegram ini berguna untuk melakukan monitoring atau pengawasan terhadap pekerja di area wajib APD atau

area yang berbahaya lainnya, sehingga pekerjaan pengawas lapangan akan lebih ringan karena pengawasan bisa berjalan secara otomatis dan dapat dilakukan dari jarak jauh.

Keunggulan lain pada penelitian ini terdapat pada konsep pendedeteksian APD yang dapat mendeteksi video pekerja yang menggunakan APD secara *realtime* saat ingin memasuki wilayah wajib APD, berbeda dari penelitian terdahulu yang hanya melakukan pendeteksian terhadap gambar baik yang ditangkap menggunakan *trigger* seperti *push bottom* ataupun gambar yang tersedia internet kemudian diidentifikasi menggunakan pendeteksian objek. Dan penelitian-penelitian sebelumnya hanya dapat melakukan pendeteksian bagian kepala saja, bukan seluruh anggota badan[5].

Akurasi yang didapatkan pada penelitian ini juga kurang lebih dapat dikatakan baik dibandingkan penelitian sebelumnya dengan menyesuaikan metode yang digunakan terhadap *device* yang dimiliki, namun dengan hasil akurasi pengujian yang tidak kalah jauh baiknya. Dimana dengan menggunakan metode *Convolutional Neural Network* (CNN) dengan model SSD-MobileNetV2 dan memanfaatkan Raspberry Pi 4B, penelitian ini masih mampu mendapatkan akurasi sebesar 76 %. Hasil tersebut bisa dikatakan baik karena bila dibandingkan dengan penelitian sebelumnya yang menggunakan metode *You Only Look Once* (YOLO) dimana metode tersebut bisa dikatakan lebih terbaru dan akurat namun disisi lain metode ini sangat berat apabila diaplikasikan pada *device* sekelas Raspberry Pi 4B, dengan metode tersebut penelitian sebelumnya hanya mendapat akurasi sebesar 73,83%[11].

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan dan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem pendeteksi APD dirancang menggunakan Raspberry Pi 4 sebagai pusat kendali sistem tersebut. Proses kerja sistem ini berlangsung secara *realtime* menggunakan model SSD-MobilenetV2 sebagai metode pengenalan objek pendeteksian, sistem ini mampu menghasilkan *output* berupa alarm dan notifikasi Telegram.
2. Sistem yang dibuat dengan memanfaatkan model SSD-MobilenetV2 sebagai pendeteksian objek ini mampu mendapatkan akurasi pendeteksian APD sebesar 93% pada siang hari, 90% pada sore hari, 43% pada malam hari dan 76% dari total keseluruhan pengujian 3 kondisi waktu tersebut.
3. Dengan memanfaatkan Telegram dalam pengiriman notifikasi hasil pendeteksian objek mendapatkan nilai *delay* sebesar 3,04 detik untuk waktu tercepat, 4,23 detik untuk waktu terlambat, dan nilai rata-rata *delay* sebesar 3,53 detik.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka terdapat saran-saran yang diuraikan sebagai berikut.

1. Penelitian ini mendapatkan hasil yang sangat buruk saat dilakukan pada malam hari dikarenakan kamera tidak cukup baik (tidak ada *infrared*) dan penerangan yang tidak memadai pada area pengujian. Disarankan menggunakan camera yang lebih baik dan menambahkan penerangan pada area pengujian.
2. Disarankan pada penelitian selanjutnya untuk menggunakan metode yang terbaru dan didukung dengan *device* yang lebih memumpuni, agar akurasi yang didapatkan juga bisa lebih baik.

3. Disarankan juga untuk menambahkan data latih yang lebih banyak lagi agar hasil lebih akurat, atau menambahkan data latih dengan kondisi tanpa APD untuk mengantisipasi adanya orang lain yang tidak berwenang dan tidak menggunakan APD sama sekali memasuki area tersebut.
4. Diharapkan pada pengembangan penelitian selanjutnya untuk menambahkan Analisa Konsumsi Daya yang dibutukan sistem ini dalam waktu 1x24 jam, hal ini dikarenakan alat tersebut bekerja secara *realtime* dan mungkin pengaplikasiannya mengikuti jam kerja suatu instansi. Dengan begitu mungkin kedepannya sistem ini bukan lagi dibuat menjadi prototipe, namun bisa dibuat menjadi produk yang dapat dijual belikan

DAFTAR PUSTAKA

[1] R. A. Zahara, “Kepatuhan Menggunakan Alat Pelindung Diri (APD),” vol. 148, pp. 148–162.

[2] J. T. Sipil and F. Teknik, “Analisis faktor-faktor penyebab kecelakaan kerja konstruksi,” no. 024, pp. 21–32, 2005.

[3] D. Cahyaningrum, H. T. Muktiana Sari, and D. Iswandari, “Faktor-Faktor yang Berhubungan dengan Kejadian Kecelakaan Kerja di Laboratorium Pendidikan,” *J. Pengelolaan Lab. Pendidik.*, vol. 1, no. 2, pp. 41–47, 2019, doi: 10.14710/jplp.1.2.41-47.

[4] P. P. Putra, “Penerapan Inspeksi Keselamatan Dan Kesehatan Kerja Sebagai Upaya Pencegahan Kecelakaan Kerja,” *Higeia J. Public Heal. Res. Dev.*, vol. 1, no. 3, pp. 84–94, 2017, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/higeia/article/view/15976.

[5] M. Ulum, M. Zakariya, and A. F. I, “Rancang Sistem Pendeteksi Alat Pelindung Diri (Apd) Berbasis Image Prosessing,” *J. Ilm. Tek. Inform. Elektron. dan Kontrol*, vol. 1, no. 1, pp. 23–30, 2021.

[6] E. E. Handayani, T. A. Wibowo, and D. Suryani, “Relationship Between Wearing Protective Equipment for Rustic Workers In PT Borneo,” *Kesmas J. UAD*, vol. 4, no. 3, pp. 208–217, 2010.

[7] R. A. Zahara, S. U. Effendi, and N. Khairani, “Kepatuhan Menggunakan Alat Pelindung Diri (APD) Ditinjau dari Pengetahuan dan Perilaku pada Petugas Instalasi Pemeliharaan Sarana Dan Prasarana Rumah Sakit (IPSRS).,” *J. Aisyah J. Ilmu Kesehat.*, vol. 2, no. 2, pp. 153–158, 2017, doi: 10.30604/jika.v2i2.60.

[8] R. D. A. ; A. Rahman, “PERANCANGAN SISTEM PENDETEKSI ALAT PELINDUNG DIRI MENGGUNAKAN TEKNOLOGI IMAGE PROCESSING,” *Inf. Serv. Use*, vol. 11, no. 1991, pp. 33–34, 2008.

[9] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, “Robust real-time unusual event detection using multiple fixed-location monitors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, 2008, doi:

10.1109/TPAMI.2007.70825.

[10] A. H. M. Rubaiyat *et al.*, "Automatic detection of helmet uses for construction safety," *Proc. - 2016 IEEE/WIC/ACM Int. Conf. Web Intell. Work. WIW 2016*, pp. 135–142, 2017, doi: 10.1109/WIW.2016.10.

[11] J. Adiwibowo, K. Gunadi, and E. Setyati, "Deteksi Alat Pelindung Diri Menggunakan Metode YOLO dan Faster R-CNN," *J. Infra*, vol. 18, no. 2, pp. 106–112, 2020.

[12] Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia, *Peraturan menteri tenaga kerja dan transmigrasi republik indonesia nomor PER.08/MEN/VII/2010 tentang alat pelindung diri*. 2008, pp. 1–69.

[13] P. M. D. Wisnu Jatmiko, *Real Time Operating System*. Depok, 2005.

[14] J. I. Komputasi, V. No, M. Ssd, V. Mobilenet, and S. Model, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih," *J. Ilm. Komputasi*, vol. 19, no. 3, pp. 421–430, 2020, doi: 10.32409/jikstik.19.3.68.

[15] R. F. Muharram, P. Studi, T. Informatika, K. Gedong, P. Rebo, and J. Timur, "IMPLEMENTASI ARTIFICIAL INTELLIGENCE UNTUK DETEKSI," vol. 01, no. 03, pp. 115–122, 2021.

[16] D. Diamanta and H. Toba, "Pendeteksian Citra Pengunjung Menggunakan Single Shot Detector untuk Analisis dan Prediksi Seasonality," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 125–141, 2021, doi: 10.28932/jutisi.v7i1.3329.

[17] A. N. A. Thohari and R. Adhitama, "Real-Time Object Detection For Wayang Punakawan Identification Using Deep Learning," *J. Infotel*, vol. 11, no. 4, pp. 127–132, 2019, doi: 10.20895/infotel.v11i4.455.

[18] D. R. H. Putra, F. Marisa, and I. D. Wijaya, "Identifikasi Wajah Berbasis Segmentasi Warna Kulit Wajah Menggunakan Naive Bayes Classifier," *J. Teknol. Inf.*, vol. 9, no. 2, pp. 99–106, 2018.

[19] A. Zelinsky, *Learning OpenCV*, vol. 16, no. 3. 2009.

[20] M. M. Hanugra Aulia Sidharta, S.T., "INTRODUCTION TO OPEN CV," [Online]. Available: https://binus.ac.id/malang/2017/10/introduction-to-

open-cv/.

[21] T. K. Akbar Nur Syahrudin, “INPUT DAN OUTPUT PADA BAHASA PEMROGRAMAN PYTHON,” pp. 1–7, 2018.

[22] J. Enterprise, *Otodidak Pemrograman Python*. Jakarta: PT Elex Media Komputindo, 2017.

[23] F. Al-Azzo, A. M. Taqi, and M. Milanova, “Human related-health actions detection using Android Camera based on TensorFlow Object Detection API,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 9–23, 2018, doi: 10.14569/IJACSA.2018.091002.

[24] A. Taufiq, M. Pratama, and A. R. Pratama, “Rancang Bangun Aplikasi Android ‘Kuliah Apa?’ Berbasis Flutter dan TensorFlow Lite,” *Automata*, vol. 2, no. 1, 2021.

[25] T. Carneiro, R. V. M. Da Nobrega, T. Nepomuceno, G. Bin Bian, V. H. C. De Albuquerque, and P. P. R. Filho, “Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications,” *IEEE Access*, vol. 6, pp. 61677–61685, 2018, doi: 10.1109/ACCESS.2018.2874767.

[26] I. F. Rizal, I. W. A. Arimbawa, and R. Afwani, “Rancang Bangun Digital Home Assistant dengan Perintah Suara Menggunakan Raspberry Pi dan Smartphone (Design and Built Digital Home Assistant with Voice Commands Using Raspberry Pi and Smartphone),” *J-Cosine*, vol. 2, no. 2, pp. 127–134, 2018, [Online]. Available: http://jcosine.if.unram.ac.id/.

[27] Provost F. and R. Kohavi, *“Special issue of applications of machine learning and the knowledge discovery process,” Glossary of Terms, vol. 30, no. 2/3, pp. 271-274*. 1998.

[28] B. Berlilana, B. A. Kusuma, and F. Ramadhan, “Smart Face-shield Berteknologi Internet of Things sebagai Alat Pelindung Diri di Era Pandemi Covid-19,” *J. Media Inform. Budidarma*, vol. 5, no. 4, p. 1559, 2021, doi: 10.30865/mib.v5i4.3307.

[29] T. Online, M. E. Laily, F. Nur, G. Qorik, and O. Pratamasunu, “Deteksi Penggunaan Alat Pelindung Diri (APD) Untuk Keselamatan dan Kesehatan Kerja Menggunakan Metode Mask Region Convolutional

Neural Network (Mask R-CNN),” vol. 8, no. 2, pp. 279–288, 2022.

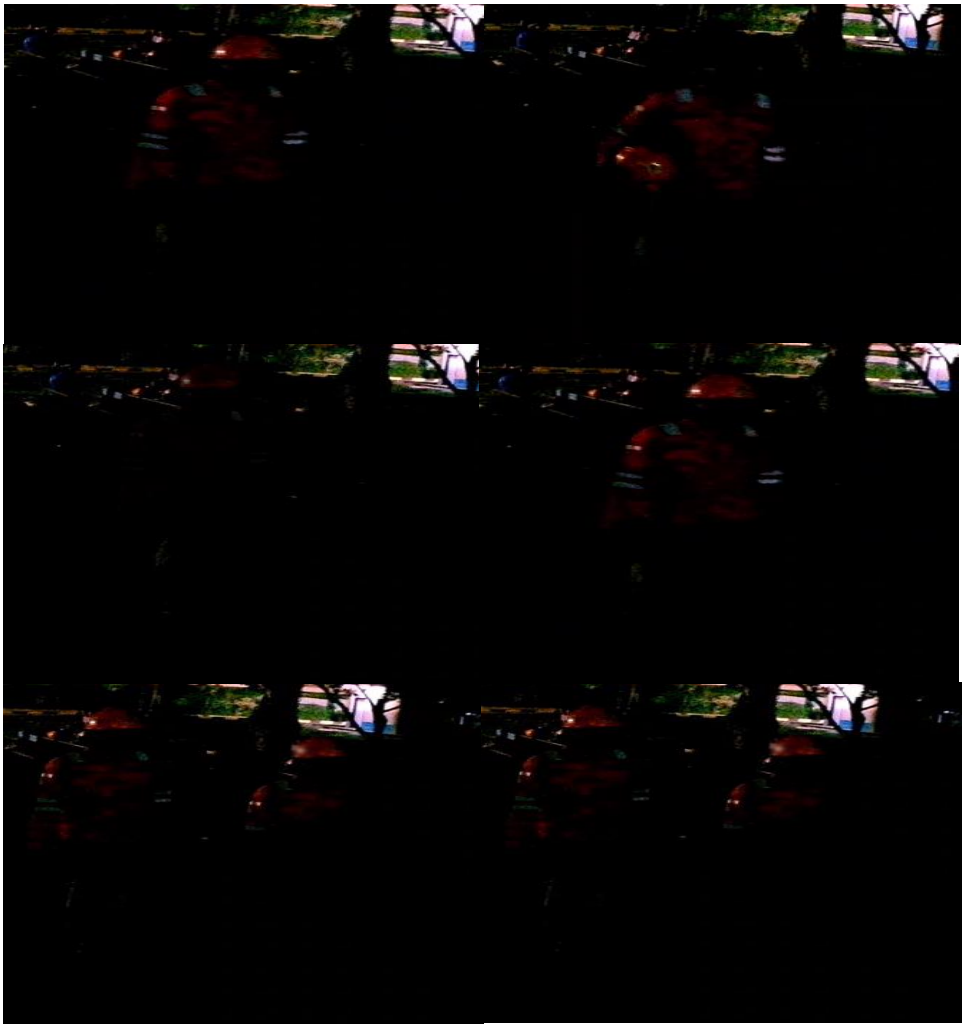
[30] R. M. Mailoa, L. W. Santoso, and J. S. Surabaya, “Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN,” 2021.

LAMPIRAN

Lampiran A Hasil Pengujian Pendeteksi APD

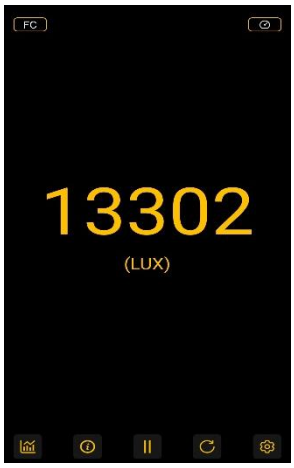
Gambar A-1 Hasil Pengujian Siang Hari

Gambar A-2 Hasil Pengujian Sore Hari

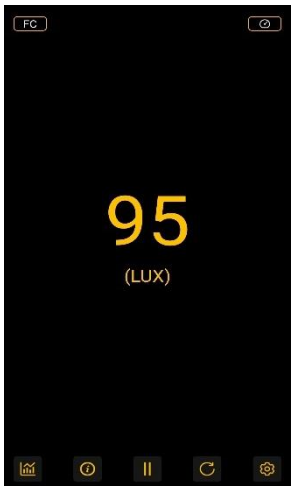


Gambar A-3 Hasil Pengujian Malam Hari

Lampiran B Intensitas Cahaya Di Area Pengujian



Gambar B-1 Lux Pada Siang Hari



Gambar B-2 Lux Pada Sore Hari

Gamabar B-3 Lux Pada Malam Hari

Lampiran C Perhitungan

A. Perhitungan Akurasi 3 Waktu Percobaan

Rumus Akurasi Dasar:

$$\text{As} = \frac{(Total\ Data\ Benar)}{(Total\ Seluruh\ data)} \times 100\%$$

1. Pehitungan Akurasi Pada Pengujian Siang Hari

$$\text{As} = \frac{28}{30} \times 100\% = 93\%$$

2. Pehitungan Akurasi Pada Pengujian Sore Hari

$$\text{As} = \frac{27}{30} \times 100\% = 90\%$$

3. Pehitungan Akurasi Pada Pengujian Malam Hari

$$\text{As} = \frac{13}{30} \times 100\% = 43\%$$

B. Perhitungan Akurasi Keseluruhan Data

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% = \frac{18 + 50}{18 + 10 + 12 + 50} \times 100\%$$
$$= \frac{68}{90} \times 100\% = 76\%$$

$$Precision = \frac{TP}{TP + FP} \times 100\% = \frac{18}{18 + 12} \times 100\% = \frac{18}{30} \times 100\% = 60\%$$

$$Recall = \frac{TP}{TP + FN} \times 100\% = \frac{18}{18 + 10} \times 100\% = \frac{18}{28} \times 100\% = 64\%$$

$$F - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \times 100\% = \frac{2 \times (0{,}6 \times 0{,}64)}{0{,}6 + 0{,}64} \times 100\%$$
$$= \frac{0{,}768}{1{,}24} \times 100\% = 76\%$$

Lampiran D Listing Kode Program

```
import cv2
import numpy as np
import math
import time
import sys
import telepot
import RPi.GPIO as GPIO
import telegram_botp
from playsound import playsound
try:
    from tflite_runtime.interpreter import Interpreter
except:
    from tensorflow.lite.python.interpreter import
Interpreter

fps = ""
detectfps = ""
framecount = 0
detectframecount = 0
time1 = 0
time2 = 0

LABELS = ['Lengkap', 'Tdklengkap']

api = '6292515999:AAE87V6JaLctXK4wp4mkOjf7Bm6sdim9WBs'
#token dari bot telegram
bot = telepot.Bot(api)

relay = 24 #masukan GPIO nya

GPIO.setup(relay, GPIO.OUT)

cam = cv2.VideoCapture(0)
```

```
cv2.namedWindow("APD Detector")

while True:
    ret, frame = cam.read()
    k = cv2.waitKey(1)



    if k%256 == 27:
        print("Escape hit, closing...")
        GPIO.output(relay, GPIO.HIGH)
        time.sleep(4)
        GPIO.output(relay, GPIO.LOW)

    elif button_state == False:
        #pushbutton
        cv2.imwrite("orang.jpg", frame)
        print("berhasil menyimpan")

        image = cv2.imread('orang.jpg')

        interpreter =
Interpreter(model_path='detect.tflite')
        try:
            interpreter.set_num_threads(4)
        except:
            pass
        interpreter.allocate_tensors()
        input_details = interpreter.get_input_details()
        output_details = interpreter.get_output_details()
        start_time = time.perf_counter()
        image_height = image.shape[0]
        image_width  = image.shape[1]

        # Resize and normalize image for network input
        t3 = time.perf_counter()
        frames = cv2.resize(image, (300, 300
```

```
        frames = cv2.cvtColor(frames, cv2.COLOR_BGR2RGB)
        frames = np.expand_dims(frames, axis=0)
        frames = frames.astype(np.float32)
        frames = frames - 127.5
        frames = frames * 0.007843
        t4 = time.perf_counter()
        print("resize and normalize time: ", t4 - t3)

        # run model
        t5 = time.perf_counter()
        interpreter.set_tensor(input_details[0]['index'],
frames)
        interpreter.invoke()
        t6 = time.perf_counter()
        print("inference + postprocess time: ", t6 - t5)

        # get results
        boxes =
interpreter.get_tensor(output_details[0]['index'])[0]
        classes =
interpreter.get_tensor(output_details[1]['index'])[0]
        scores =
interpreter.get_tensor(output_details[2]['index'])[0]
        count =
interpreter.get_tensor(output_details[3]['index'])[0]

        for box, classidx, score in zip(boxes, classes,
scores):
            probability = score
            if probability >= 7.6:
                if (not math.isnan(box[0]) and
                    not math.isnan(box[1]) and
                    not math.isnan(box[2]) and
                    not math.isnan(box[3])):
                    pass
                else:
```

```
                        continue
                ymin = int(box[0] * image_height)
                xmin = int(box[1] * image_width)
                ymax = int(box[2] * image_height)
                xmax = int(box[3] * image_width)
                if ymin > ymax:
                    continue
                if xmin > xmax:
                    continue

                classnum = int(classidx)
                probability = score
                print('coordinates: ({}, {})-({}, {}).
class: "{}". probability: {:.2f}'.format(xmin, ymin, xmax,
ymax, classnum, probability))
                cv2.rectangle(image, (xmin, ymin), (xmax,
ymax), (0, 255, 0), 2)
                cv2.putText(image, '{}:
{:.2f}'.format(LABELS[classnum],probability), (xmin, ymin +
40), cv2.FONT_HERSHEY_COMPLEX, 0.6, (0, 255, 0), 1)

                if classnum == 0 :
                    print('APD Lengkap. Silahkan masuk!')
                    GPIO.output(relay, GPIO.HIGH)
                    time.sleep(2)
                    GPIO.output(relay, GPIO.LOW)
                    bot.sendPhoto("679837510",
photo=open("orang.jpg","rb")) #untuk mengirim gambar ke
telegram
                    bot.sendMessage("679837510", "APD
Lengkap. Silahkan masuk!")  #untuk memberikan notif ke
telegram

                elif classnum == None :
                    print('APD Tidak Lengkap. Lengkapi APD
Anda!')
```

```
                    playsound("akses_ditolak.mp3")
                    bot.sendPhoto("699866510",
photo=open("orang.jpg","rb")) #untuk mengirim gambar ke
telegram
                    bot.sendMessage("699866510", "APD Tidak
Lengkap. Lengkapi APD Anda!")  #untuk memberikan notif ke
telegram

                else :
                    pass

    #7. Resize and normalize image for network input
    cv2.imshow("test", frame)

cam.release()
GPIO.cleanup()
```