



**SISTEM PENGHITUNG JUMLAH PENGUNJUNG DI RESTORAN
MENGUNAKAN KAMERA BERBASIS *SINGLE SHOT DETECTOR* (SSD)**

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

**UNIVERSITAS
Dinamika**

Oleh:

Yosia Pradeska Admaja

17410200022

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2021

**SISTEM PENGHITUNG JUMLAH PENGUNJUNG DI RESTORAN
MENGUNAKAN KAMERA BERBASIS *SINGLE SHOT DETECTOR* (SSD)**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



UNIVERSITAS
Dinamika

Oleh:

Nama : Yosia Pradeska Admaja
NIM : 17410200022
Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

202

Tugas Akhir

SISTEM PENGHITUNG JUMLAH PENGUNJUNG DI RESTORAN MENGUNAKAN KAMERA BERBASIS *SINGLE SHOT DETECTOR* (SSD)

Dipersiapkan dan disusun oleh

Yosia Pradeska Admaja

NIM: 17410200022

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: Rabu, 28 Juli 2021

Susunan Dewan Pembahas

Pembimbing:


- I. Heri Pratikno, M.T., MTCNA., MTCRE.
NIDN 0716117302
- II. Ira Puspasari, S.Si., M.T.
NIDN 0710078601


DN: cn=Heri Pratikno, M.T.,
o=Universitas Dinamika, ou=Prodi
S1 Teknik Komputer,
email=heri@dinamika.ac.id, c=ID
Date: 2021.07.28 13:15:53 +0700


Digitally signed by
Universitas Dinamika
Date: 2021.07.28
11:16:36 +0700

Pembahas:

- I. Weny Indah Kusumawati, S.Kom., M.MT.
NIDN 0721047201


DN: cn=Weny Indah
Kusumawati, o=Teknologi dan
Informatika, Undika, ou=Teknik
Komputer,
email=weny@dinamika.ac.id,
c=ID
Date: 2021.07.28 13:30:49
+0700

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar Sarjana



Digitally signed by
Universitas Dinamika
Date: 2021.07.31
11:35:29 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

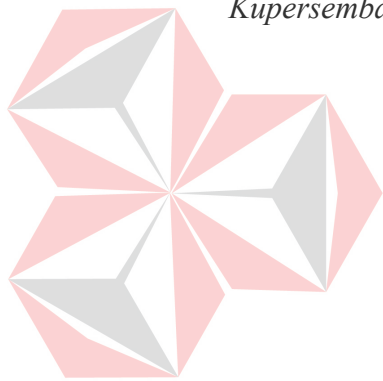
Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



*"Congratulations. You are still alive. Most people are so ungrateful to be alive.
But not you. Not anymore." -Jigsaw*

UNIVERSITAS
Dindamika



*Kupersembahkan Tugas Akhir ini kepada kedua orang tua dan teman-teman
yang selalu memberikan dukungan serta semangat.*

UNIVERSITAS
Dinamika

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Yosia Pradeska Admaja

NIM : 17410200022

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Kerja : Tugas Akhir

Judul Karya : **SISTEM PENGHITUNG JUMLAH PENGUNJUNG DI
RESTORAN MENGGUNAKAN KAMERA BERBASIS
SINGLE SHOT DETECTOR (SSD)**

Menyatakan dengan sesungguhnya bahwa:

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

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 25 Juni 2021

Yang Menyatakan,



Yosia Pradeska Admaja

NIM 17410200022

ABSTRAK

Awal tahun 2020 adalah sejarah baru bagi Indonesia pada khususnya dan dunia pada umumnya, pada bulan Maret ditemukan sebuah varietas virus baru yang bernama *Covid-19* yang kemudian disebut dengan virus Corona. Kemunculan virus ini oleh *World Health Organization* (WHO) telah dikategorikan sebagai pandemi karena telah menyebar hingga keseluruh dunia. Dalam upaya pencegahan penularan virus tersebut, pemerintah Indonesia mulai mencanangkan pola hidup *new normal* dengan membuat aturan protokol kesehatan untuk masyarakat, seperti kewajiban mencuci tangan, wajib menggunakan masker, membatasi mobilitas, menjaga jarak, dan membatasi jumlah pengunjung pada pusat keramaian, salah satunya adalah restoran. Kapasitas yang diizinkan adalah 50% dari total kapasitas jumlah pengunjung dalam satu area atau ruangan. Berdasarkan latar belakang permasalahan tersebut, maka pada Tugas Akhir ini penulis berusaha untuk memberikan solusi dalam bentuk upaya penegakan protokol kesehatan melalui layanan dukungan dan penerapan dari sisi teknologi digital, melalui pembuatan sistem yang mampu menghitung jumlah pengunjung yang masuk, keluar dan masih berada di dalam suatu ruangan secara otomatis yang diproses melalui pengolahan citra digital menggunakan metode *Single Shot Detector* (SSD) dengan bahasa pemrograman *Python*. Dari hasil pengujian yang dilakukan, aplikasi penghitung jumlah pengunjung ini mampu membedakan manusia (*person*) dengan *object* lain, sehingga memudahkan dalam proses penghitungan jumlah pengunjung yang masuk, keluar, dan masih berada di dalam suatu ruangan. Hasil penghitungan ditampilkan pada bagian kanan atas layar setelah melakukan pengujian dengan video hasil rekaman dan hasil penghitungan juga disajikan dalam bentuk file *spreadsheet* (.csv). Tingkat akurasi penghitungan yang didapatkan dari Tugas Akhir ini adalah 86% dalam mendeteksi dan menghitung pengunjung saat keadaan ramai. Disamping itu, tingkat akurasi dalam mendeteksi *object* yang sedang berjalan berada di angka 100%, akurasi deteksi *object* berjalan cepat berada di angka 90%, dan akurasi deteksi untuk *object* berlari adalah 50%.

Kata Kunci: *Single Shot Detector (SSD)*, *Object Detection*, *Python*, jumlah pengunjung, *MobileNet*

KATA PENGANTAR

Puji dan syukur atas kehadiran Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “Sistem Penghitung Jumlah Pengunjung Di Restoran Menggunakan Kamera Berbasis *Single Shot Detector (SSD)*”.

Dalam pelaksanaan Tugas Akhir dan penyelesaian laporan Tugas Akhir ini, penulis mendapatkan bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Tuhan Yesus Kristus atas segala penyertaan, kebaikan dan anugerah-Nya penulis dapat menyelesaikan Tugas Akhir ini.
2. Orang Tua dan Seluruh Keluarga tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi, sehingga penulis dapat menempuh Tugas Akhir dan dapat menyusun laporan Tugas Akhir.
3. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembimbing 1 yang telah memberikan saran dan juga wawasan bagi penulis selama pelaksanaan Tugas Akhir dan pembuatan laporan Tugas Akhir.
4. Ibu Ira Puspasari, S.Si M.T., selaku dosen pembimbing 2 yang memberikan masukan serta koreksi bagi penulis selama pelaksanaan Tugas Akhir dan penyusunan laporan Tugas Akhir.
5. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembahas yang telah memberikan saran agar Tugas Akhir ini berjalan dengan baik.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh karena itu penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk lebih baik lagi.

Surabaya, 15 Juli 2021

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1 Restoran.....	6
2.2 <i>Python</i>	6
2.3 <i>Computer Vision</i>	6
2.4 <i>OpenCV (Open Source Computer Vision)</i>	7
2.5 <i>Single Shot Detector (SSD)</i>	8
2.6 Kamera	11
BAB III METODOLOGI PENELITIAN	12
3.1 Perancangan Perangkat Keras (<i>Hardware</i>)	12
3.2 Perancangan Perangkat Lunak (<i>Software</i>)	13
3.3 Persiapan Pembuatan Program.....	16
3.4 Deteksi <i>Object</i>	17
3.5 Seleksi <i>Object</i> Manusia (<i>person</i>)	18
BAB IV HASIL DAN PEMBAHASAN.....	20
4.1 Pengujian Tingkat Akurasi <i>Person Detection</i>	20
4.1.1 Tujuan Pengujian Tingkat Akurasi <i>Person Detection</i>	20
4.1.2 Prosedur Pengujian Tingkat Akurasi <i>Person Detection</i>	20

4.1.3 Hasil Pengujian Tingkat Akurasi <i>Person Detection</i>	21
4.2 Pengujian Akurasi Penghitungan Jumlah Pengunjung	22
4.2.1 Tujuan Pengujian Akurasi Penghitungan Jumlah Pengunjung ..	22
4.2.2 Prosedur Pengujian Akurasi Penghitungan Jumlah Pengunjung	22
4.2.3 Hasil Pengujian Akurasi Penghitungan Jumlah Pengunjung	23
4.3 Pengujian Notifikasi Kapasitas Maksimal	24
4.3.1 Tujuan Pengujian Notifikasi Kapasitas Maksimal	24
4.3.2 Prosedur Pengujian Notifikasi Kapasitas Maksimal	24
4.3.3 Hasil Pengujian Notifikasi Kapasitas Maksimal	25
4.4 Pengujian Akurasi <i>Data Reporting</i>	26
4.4.1 Tujuan Pengujian Akurasi <i>Data Reporting</i>	26
4.4.2 Prosedur Pengujian Akurasi <i>Data Reporting</i>	26
4.4.3 Hasil Pengujian Akurasi <i>Data Reporting</i>	26
BAB V PENUTUP	28
5.1 Kesimpulan.....	28
5.2 Saran.....	28
LAMPIRAN	30



UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Contoh hasil dari <i>computer vision</i>	7
Gambar 2.2 Arsitektur <i>Single Shot Detector</i> (SSD).....	8
Gambar 2.3 Contoh deteksi object dengan metode <i>Single Shot Detector</i> (SSD)....	8
Gambar 2.4 Arsitektur VGG 16	9
 Gambar 3.1 Model perancangan	 12
Gambar 3.2 Algoritma sistem	13
Gambar 3.3 Tampilan aplikasi pada layar monitor TV atau PC	15
Gambar 3.4 Tampilan notifikasi ketika pengunjung mencapai batas maksimal...	15
Gambar 3.5 Hasil pengujian deteksi <i>object</i>	18
Gambar 3.6 Hasil pengujian seleksi <i>object</i>	19
 Gambar L1 Hasil pengujian notifikasi kapasitas maksimal 15 orang	 32
Gambar L2 Hasil pengujian notifikasi kapasitas maksimal 30 orang	33
Gambar L3 Hasil pengujian notifikasi kapasitas maksimal 50 orang	33
Gambar L4 Hasil penghitungan pada aplikasi saat belum ada pengunjung	34
Gambar L5 Hasil penghitungan pada file excel	34
Gambar L6 Hasil penghitungan pada aplikasi ketika sudah ada penunjung	35
Gambar L7 Hasil penghitungan pada file excel	35
Gambar L8 Hasil penghitungan pada aplikasi ketika sudah ada penunjung	36
Gambar L9 Hasil penghitungan pada file excel	36
Gambar L10 Hasil cek plagiarisme aplikasi turnitin halaman 1	37
Gambar L11 Hasil cek plagiarisme aplikasi turnitin halaman 2	38
Gambar L12 Hasil cek plagiarisme aplikasi turnitin halaman 3	39
Gambar L13 Hasil cek plagiarisme aplikasi turnitin halaman 4	40
Gambar L14 Hasil cek plagiarisme aplikasi turnitin halaman 5	41
Gambar L15 Hasil cek plagiarisme aplikasi turnitin halaman 6	41

DAFTAR TABEL

	Halaman
Tabel 4.1 Hasil pengujian akurasi <i>person detection</i>	21
Tabel 4.2 Hasil pengujian akurasi penghitungan	23
Tabel 4.3 Hasil pengujian notifikasi.....	25
Tabel 4.4 Hasil pengujian akurasi <i>data reporting</i>	27



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 <i>Code Person Detection</i>	30
Lampiran 2 <i>Code Counter Object</i>	31
Lampiran 3 <i>Code Notifikasi Kapasitas Maksimal</i>	32
Lampiran 4 <i>Code Data Reporting</i>	32
Lampiran 5 Gambar Hasil Pengujian Notifikasi Kapasitas Maksimal 15 Orang .	32
Lampiran 6 Gambar Hasil Pengujian Notifikasi Kapasitas Maksimal 30 Orang .	33
Lampiran 7 Gambar Hasil Pengujian Notifikasi Kapasitas Maksimal 50 Orang .	33
Lampiran 8 Gambar Hasil Pengujian Akurasi Data Reporting Pada Aplikasi	34
Lampiran 9 Gambar Hasil Pengujian Akurasi Data Reporting Pada File Excel..	34
Lampiran 10 Gambar Hasil Pengujian Akurasi Data Reporting Pada Aplikasi ...	35
Lampiran 11 Gambar Hasil Pengujian Akurasi Data Reporting Pada File Excel.	35
Lampiran 12 Gambar Hasil Pengujian Akurasi Data Reporting Pada Aplikasi ...	36
Lampiran 13 Gambar Hasil Pengujian Akurasi Data Reporting Pada File Excel.	36
Lampiran 14 Hasil Cek Plagiarisme Pada Aplikasi Turnitin	37



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada 2 Maret 2020, untuk pertama kalinya pemerintah mengumumkan dua kasus pasien positif Covid-19 di Indonesia (Ellyvon, 2020). Sejak itu, pemerintah mulai melakukan berbagai upaya untuk mengendalikannya. *World Health Organization (WHO)* menyatakan virus corona dapat dikategorikan sebagai sebuah pandemi karena telah menyebar semakin luas ke seluruh dunia (Ducharme, 2020).

Penerapan aturan jaga jarak atau *physical distancing* sebagai upaya pencegahan penyebaran wabah ini tidak sepenuhnya disadari dan dilakukan warga. Seperti di restoran penerapan aturan ini masih dilakukan setengah hati (Diah, 2021).

Penerapan jarak antrean misalnya tidak diimbangi pembatasan pengunjung yang masuk. Namun saat ini sudah banyak restoran sudah menyiapkan diri untuk menerapkan tatanan kehidupan baru (*new normal*) dalam menerima pengunjung dengan mengacu pada protokol kesehatan yang sebelumnya sudah ditentukan oleh satuan tugas *COVID-19* dan pemerintah. Anjuran pemerintah yang berlaku sekarang adalah wajib menggunakan masker, mencuci tangan, dan menerapkan pembatasan pengunjung dengan jumlah 50% dari total kapasitas jumlah pengunjung dalam satu area atau ruangan (Ulie, 2021). Dengan begitu diharapkan dapat meminimalisir tingkat penyebaran virus *COVID-19* dan terciptanya *cluster* penyebaran baru.

Dari permasalahan tersebut maka dalam Tugas Akhir ini dilakukan pembuatan sistem yang mampu menghitung jumlah pengunjung dengan *image processing* melalui video hasil tangkapan kamera yang terpasang diatas pintu masuk, sehingga dapat mendeteksi jumlah pengunjung masuk, keluar, dan sedang berada dalam restoran. Selain itu, pemilik usaha juga bisa mengetahui jumlah pengunjung dalam kurun waktu tertentu, serta dapat menjadi bukti bahwa pemilik usaha sudah menerapkan pembatasan jumlah kunjungan apabila suatu saat diminta data oleh pihak berwenang. Sistem ini dibuat sederhana dengan memanfaatkan komputer yang

ada dan bersifat lokal, sehingga tidak memerlukan koneksi internet atau instalasi perangkat apapun selain komputer dan kamera *external*.

Penelitian mengenai *object detection* telah banyak dilakukan oleh peneliti, salah satu contoh penelitian dalam *object detection* menggunakan metode *Convolutional Neural Network* (CNN) adalah *Single Shot Detector* (SSD), SSD merupakan salah satu pendekatan yang populer dari tiga pendekatan yaitu SSD, *You Only Look Once* (YOLO) dan *Region-proposal Convolutional Neural Network* (RCNN). RCNN memberikan hasil yang akurat namun proses yang lama sementara YOLO memberikan hasil yang cepat namun kurang akurat dari RCNN, maka dari itu SSD merupakan pilihan terbaik, yang memberikan hasil lebih akurat dari YOLO walaupun tidak lebih akurat dari RCNN dan memberikan hasil lebih cepat dari RCNN walaupun tidak lebih cepat dari YOLO, walaupun begitu SSD juga memiliki beberapa kekurangan seperti tidak terlalu baik dalam mendeteksi objek yang kecil (Chen Z., 2019).

Sampai saat ini sudah ada beberapa penelitian yang memiliki konsep hampir sama dan mengarah kepada permasalahan penghitungan *object*, salah satunya adalah (R. G. Fajri, 2021) dengan judul perancangan program pendeteksi dan pengklasifikasi jenis kendaraan dengan metode *Convolutional Neural Network* (CNN) *deep learning*. Di dalam penelitian tersebut membahas mengenai proses deteksi, klasifikasi, dan menghitung kendaraan menggunakan metode *Convolutional Neural Network* (CNN) *Deep Learning* dengan algoritma *You Only Look Once* (YOLO). Pada penelitian tersebut juga masih memiliki kekurangan yaitu akurasi belum maksimal ketika mendeteksi *object* di jalan yang padat, dan belum ada fitur untuk *reporting* hasil proses identifikasi *object* kedalam bentuk file PDF atau *spreadsheet*.

Penelitian lainnya menggunakan manusia sebagai *object image processing* adalah (Kusno Prasetya, 2020) dengan judul *implementation of tensorflow in the CCTV-based people counter application at PT Matahari Department Store, Tbk*. Pada penelitian tersebut membahas tentang digunakan untuk menghitung keluar masuknya orang-orang di toko PT. Matahari Department Store menggunakan algoritma *TensorFlow*, namun dari sudut pandang penulis, karya tersebut masih memiliki beberapa kekurangan yaitu penempatan kamera yang belum tepat, serta

akurasi pengenalan *object* masih belum akurat, penggunaan kamera yang memiliki resolusi kecil, dan belum ada fitur untuk menuliskan hasil deteksi ke dalam file dengan format lain (PDF atau *spreadsheet*).

1.2 Perumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan permasalahan dalam Tugas Akhir ini adalah:

1. Bagaimana tingkat akurasi sistem SSD dalam mendeteksi pengunjung restoran yang berupa *object* manusia ketika dalam kondisi berjalan, berjalan cepat, dan berlari?
2. Bagaimana akurasi data hasil proses penghitungan jumlah pengunjung yang keluar – masuk, sehingga dapat diketahui jumlah pengunjung yang berada dalam area restoran?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, terdapat beberapa batasan masalah, antara lain:

1. Data *input* tidak disiarkan secara langsung (*real-time*) dari hasil kamera karena hanya menggunakan hasil rekaman video.
2. Kamera berada dalam pencahayaan normal dan cukup.
3. Sistem tidak mendeteksi suhu tubuh, jarak (*physical distancing*), dan penggunaan masker.
4. Hasil *output* untuk penyampaian informasi berupa video dari kamera dengan text hasil *counter*, dan ditampilkan melalui monitor TV/PC serta notifikasi berupa *buzzer* dari komputer untuk indikator apabila kapasitas mencapai batas maksimal.
5. *Object* yang akan di deteksi tidak berhimpitan dan tidak terhalang benda lain.
6. Sistem hanya untuk jaringan lokal.

1.4 Tujuan

Berdasarkan uraian latar belakang dan rumusan masalah di atas, maka tujuan dari Tugas Akhir ini yaitu sebagai berikut:

1. Merancang sistem yang bisa mendeteksi pengunjung pada restoran menggunakan metode *Single Shot Detector* (SSD) secara cepat dan akurat.
2. Membuat sistem yang mempunyai akurasi tinggi dalam proses perhitungan keluar-masuknya pengunjung dari sebuah restoran, sehingga dapat diketahui jumlah pengunjung yang berada dalam area restoran.

1.5 Sistematika Penulisan

Tugas akhir ini dibuat secara sistematis dengan tujuan agar lebih mudah dipahami oleh pembaca tentang keseluruhan persoalan dan alur pembahasannya.

Berikut ini adalah susunansistematika penulisan:

BAB I PENDAHULUAN

Pada bab I membahas latar belakang masalah yang sedang terjadi, perumusan masalah, membuat batasan masalah, dan tujuan pembuatan Tugas Akhir serta sistematika penulisan buku.

BAB II LANDASAN TEORI

Pada bab ini lebih difokuskan untuk membahas mengenai beberapa teori dan metode yang digunakan pada Tugas Akhir ini. Teori yang dibahas adalah penggunaan bahasa pemrograman *Python*, definisi dan fungsi *Computer Vision*, *OpenCV* (*Open Source Computer Vision*), penggunaan *dataset mobilenet_SSD*, dan metode *Single Shot Detector* (SSD).

BAB III METODE PENELITIAN

Dalam bab 3 menjelaskan tentang penggunaan metode penelitian dan alasan menggunakan metode tersebut. Disamping itu, pada bab ini juga dijelaskan tentang perancangan perangkat keras (*hardware*) dengan integrasi perangkat lunak (*software*) sebagai pengolah data *input*.

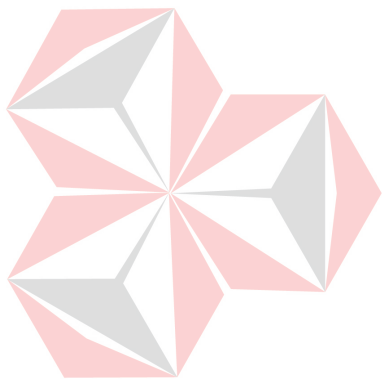
BAB IV PENGUJIAN DAN ANALISIS SISTEM

Bab ini berisi tentang tahap pengujian secara keseluruhan. Pengujian dilakukan dengan menghitung nilai akurasi pada *object* manusia yang tereteksi,

menghitung nilai akurasi penghitungan jumlah pengunjung, menghitung akurasi notifikasi ketika mencapai batas maksimal pengunjung, dan menghitung nilai akurasi perbandingan hasil penghitungan aplikasi dengan hasil yang tercatat pada file *data reporting*.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan penelitian serta saran untuk pengembangan peneliti dan pengembangan sistem kedepannya.



UNIVERSITAS
Dinamika

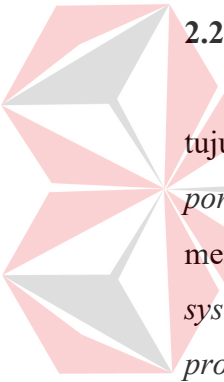
BAB II

LANDASAN TEORI

2.1 Restoran

Restoran merupakan kata serapan dari bahasa Perancis yaitu '*restaurer*' yang berarti 'memulihkan' dalam bahasa Indonesia. Pada umumnya restoran merupakan bidang usaha yang menyediakan hidangan makanan dan minuman dengan menetapkan tarif tertentu kepada masyarakat. Selain menyediakan makanan dan minuman, terdapat juga layanan *take-out dining* dan *delivery service* yang disediakan oleh beberapa restoran.

2.2 Python



Python merupakan bahasa pemrograman *open source* yang dibuat dengan tujuan mengoptimalkan *software quality*, *developer productivity*, *program portability*, dan *component integration*. *Python* telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti *internet scripting*, *systems programming*, *user interfaces*, *product customization*, *numeric programming* (Lutz, 2010).

2.3 Computer Vision

Computer vision merupakan suatu bidang ilmu komputer yang mereplikasi cara kerja mata manusia dan menerapkannya kedalam komputer, sehingga mampu untuk melihat, mengidentifikasi, dan memproses gambar dengan cara yang sama seperti yang dilakukan oleh manusia kemudian memberikan *output* yang sesuai. Hal ini seperti menanamkan naluri dan kecerdasan manusia ke dalam komputer. *Computer vision* berkaitan erat dengan kecerdasan buatan karena komputer harus menginterpretasikan apa yang dilihatnya dan kemudian melakukan analisis.

Tujuan *computer vision* tidak hanya untuk melihat, tetapi juga untuk memproses dan juga memberikan hasil yang bermanfaat berdasarkan pengamatan

misalnya, komputer dapat membuat gambar 3D dari gambar 2D. Seperti halnya pada sebuah mobil. Saat sebuah mobil dilengkapi dengan *computer vision*, maka mobil dapat mengidentifikasi dan membedakan objek yang berada di jalan seperti lampu lalu lintas, rambu lalu lintas, pejalan kaki dan sebagainya. *Computer vision* memberikan tanda kepada pengemudi atau bahkan akan membuat mobil berhenti apabila ada hambatan yang mendadak di jalan. Pada Gambar 2.1 menunjukkan hasil pengolahan *computer vision*.



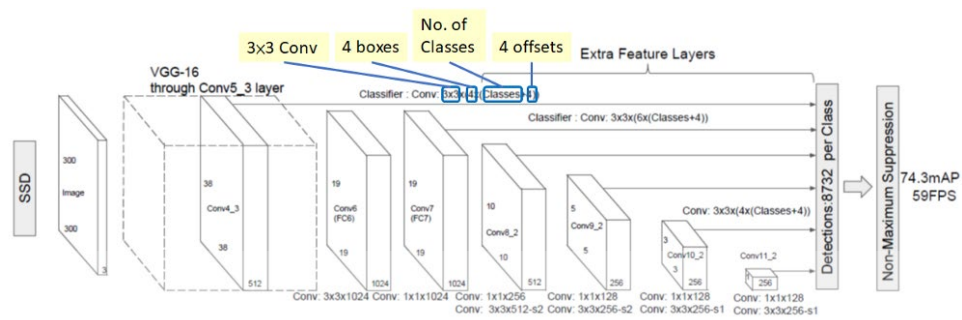
Gambar 2.1 Contoh hasil dari *computer vision*
(Sumber: www.medium.com)

2.4 OpenCV (*Open Source Computer Vision*)

OpenCV merupakan sebuah *library* yang dikembangkan oleh pusat penelitian Intel di Nizhny Novgorod, Rusia.. Penggunaan *OpenCV* ini dapat diaplikasikan di berbagai bahasa pemrograman lain seperti *Python*, *C*, *C++*, *Java*. Disamping itu, para *developer* juga bisa mengimplementasikan *OpenCV* pada berbagai macam sistem operasi yang ada seperti *Linux*, *Windows*, *IoS*, *Mac OS*, dan *Android*. Pengembangan *library OpenCV* ini sangat berpengaruh terhadap efisiensi komputasi dengan fokus yang kuat untuk diimplementasikan pada aplikasi *real-time*. Cara kerja *OpenCV* ini adalah engan berusaha menduplikasi cara kerja penglihatan atau visual mata manusia (Fajirwan, 2017).

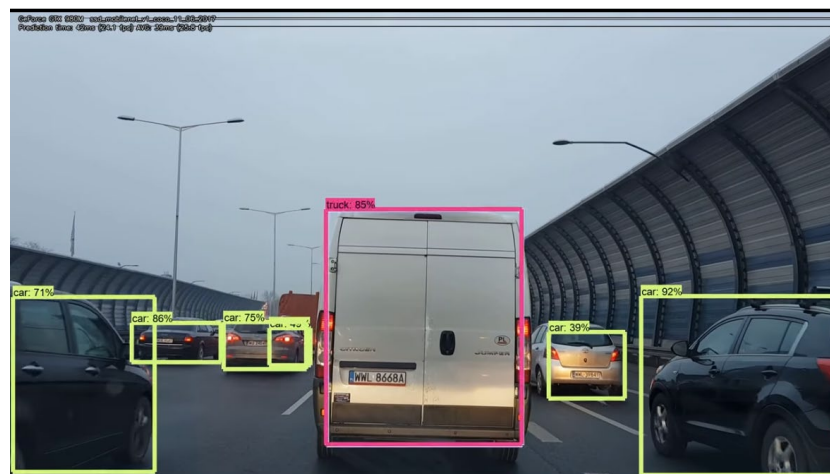
2.5 Single Shot Detector (SSD)

Single shot detector (SSD) merupakan suatu metode yang digunakan untuk mendeteksi *object* tertentu yang ada di gambar atau video dengan menggunakan *single deep neural network* dan *single shot detector* (SSD) merupakan salah satu algoritma pendeteksi *object* paling populer digunakan karena memiliki akurasi dan kecepatan pengolahan citra yang lebih baik jika dibandingkan dengan metode lainnya seperti *Faster R-CNN*, *YOLOv1* (Sik-Ho Tsang, 2018). Gambar 2.2 adalah arsitektur dari *Single Shot Detector* (SSD).



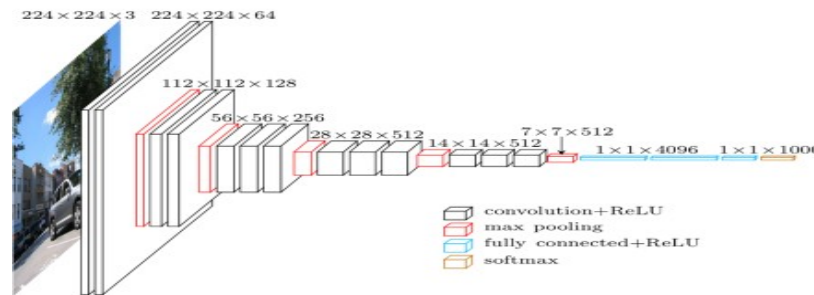
Gambar 2.2 Arsitektur *Single Shot Detector* (SSD)
(Sumber: miro.medium.com)

Single shot detector (SSD) bekerja dengan melakukan satu kali bidikan pada *object* yang ada di gambar atau video untuk menentukan area dan *object* yang berada dalam area bidikan tersebut, berbeda dengan metode *Faster R-CNN*, *YOLOv1* yang melakukan dua kali bidikan. Gambar 2.3 adalah salah satu contoh dari hasil deteksi *object* dengan metode *Single Shot Detector* (SSD).



Gambar 2.3 Contoh deteksi *object* dengan metode *Single Shot Detector* (SSD)
(Sumber: towardsdatascience.com)

Pada *Single Shot Detector (SSD)* terdapat sebuah arsitektur berbasis VGG-16 yang bertugas sebagai *feature extration* dan memiliki keunggulan dalam proses kinerja klasifikasi gambar yang beresolusi tinggi, sehingga memungkinkan untuk mengekstrak fitur di berbagai skala dan secara progresif mengurangi ukuran *input* ke setiap lapisan berikutnya. Contoh arsitektur dari VGG16 bisa dilihat pada Gambar 2.4.



Gambar 2.4 Arsitektur VGG 16
(Sumber: www.researchgate.net)

Arsitektur *single shot detector (SSD)* masuk kedalam jenis *Convolutional Neural Network (CNN)*, *Convolutional Neural Network (CNN)* adalah salah satu jenis *Neural Network* yang sering digunakan pada pemrosesan data *image*. *Convolutional Neural Network (CNN)* dapat digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image*.

Terdapat 2 bagian besar pada arsitektur *Convolutional Neural Network (CNN)* yaitu *Feature Extraction Layer* dan *Convolutional Layer*. Pada bagian *Feature Extraction Layer* dilakukan proses *encoding* dari sebuah *image* menjadi fitur yang berupa angka-angka yang merepresentasikan *image* tersebut, disamping itu pada bagian *Convolutional Layer* yang terdiri dari *neuron* yang tersusun sedemikian rupa, sehingga membentuk sebuah *filter* yang panjang dan tinggi (*pixels*). Secara matematis, *Convolutional layer* atau yang biasa disebut dengan ‘konvolusi’ adalah integral yang mencerminkan jumlah lingkaran dari sebuah sudut fungsi F yang digeser atas fungsi g , sehingga menghasilkan fungsi h .

Konvolusi dilambangkan dengan arsterik (*), yang ditunjukkan dalam persamaan [1], sehingga, $F*g=h$ berarti fungsi F dikonvolusikan dengan fungsi g menghasilkan fungsi h . konvolusi dua buah fungsi $F(x)$ dan $g(x)$ di definisikan sebagai berikut:

$$h(x) = F(x) * g(x) = \int F(a)g(x - a)$$

Konvolusi dua buah fungsi $F(x)$ dan $g(x)$ [1]

Untuk fungsi diskrit, konvolusi didefinisikan sebagai berikut dengan rumus [2]:

$$h(x) = F(x) * g(x) = \sum_{-\infty}^{\infty} F(a)g(x - a)$$

Konvolusi fungsi diskrit [2]

Pada rumus [3], $\mathbf{g(x)}$ disebut dengan kernel konvolusi (*filter*). Kernel $\mathbf{g(x)}$ merupakan jendela yang dioperasikan secara bergeser pada sinyal masukan $\mathbf{F(x)}$. Hasil konvolusi dinyatakan dengan keluaran $\mathbf{h(x)}$. Contoh, misal citra $\mathbf{F(x,y)}$ yang berukuran 5x5 sebuah kernel dengan 3x3 matriks sebagai berikut:

$$g(x, y) \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} F(x, y) \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix}$$

Citra (x,y) berukuran 5x5 dan sebuah kernel dengan 3x3 matriks [3]

Secara umum metode *Single Shot Detector* (SSD) mempunyai sebuah rumus sederhana dalam menentukan *default boxes* dan *scale default boxes*, dimana N merupakan jumlah *default boxes*, L_{conf} = *loss classification*, L_{loc} = *loss localization*, L = *prediction box* dan g = *truth ground box*. Untuk menentukan *default boxes* bisa menggunakan rumus (4) sebagai acuan:

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + aL_{loc}(x, l, g))$$

Rumus *default boxes* [4]

Sedangkan untuk menentukan *scale default boxes* bisa menggunakan rumus (5):

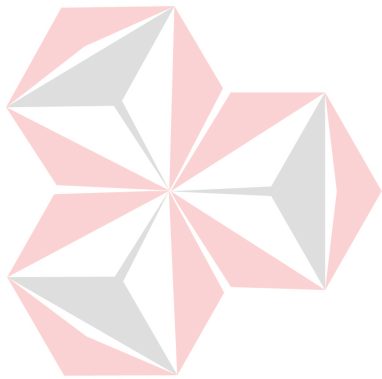
$$S_k = S_{min} \frac{S_{max} - S_{min}}{m - 1} (k - 1), k \in [1, m]$$

Rumus *scale default box* [5]

Dimana S_{min} adalah lapisan skala terendah, S_{max} adalah lapisan skala tertinggi, dan S_k adalah *input pixel*.

2.6 Kamera

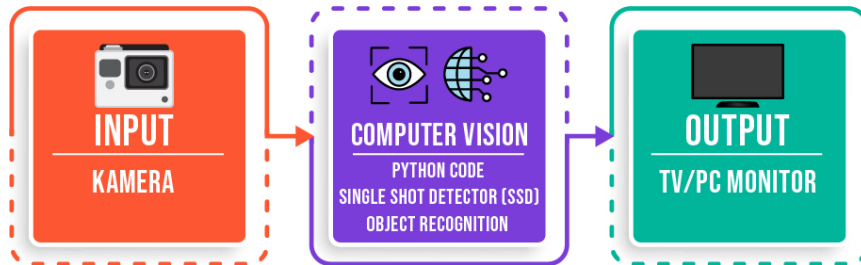
Kamera merupakan perangkat elektronik yang mampu merekam gambar atau video yang secara langsung bisa tersimpan pada kartu memori yang ada didalam. Kamera merupakan susunan dari lensa dan komponen lainnya yang memiliki cekungan tertutup dengan bukaan (*aperture*) pada salah satu ujungnya agar cahaya dapat masuk ke permukaan untuk melihat dan dapat menangkap cahaya diujung lainnya. Kamera dapat bekerja dengan cahaya spektrum yang terlihat atau bagian lain dari spektrum elektromagnetik.



UNIVERSITAS
Dinamika

BAB III METODOLOGI PENELITIAN

3.1 Perancangan Perangkat Keras (*Hardware*)



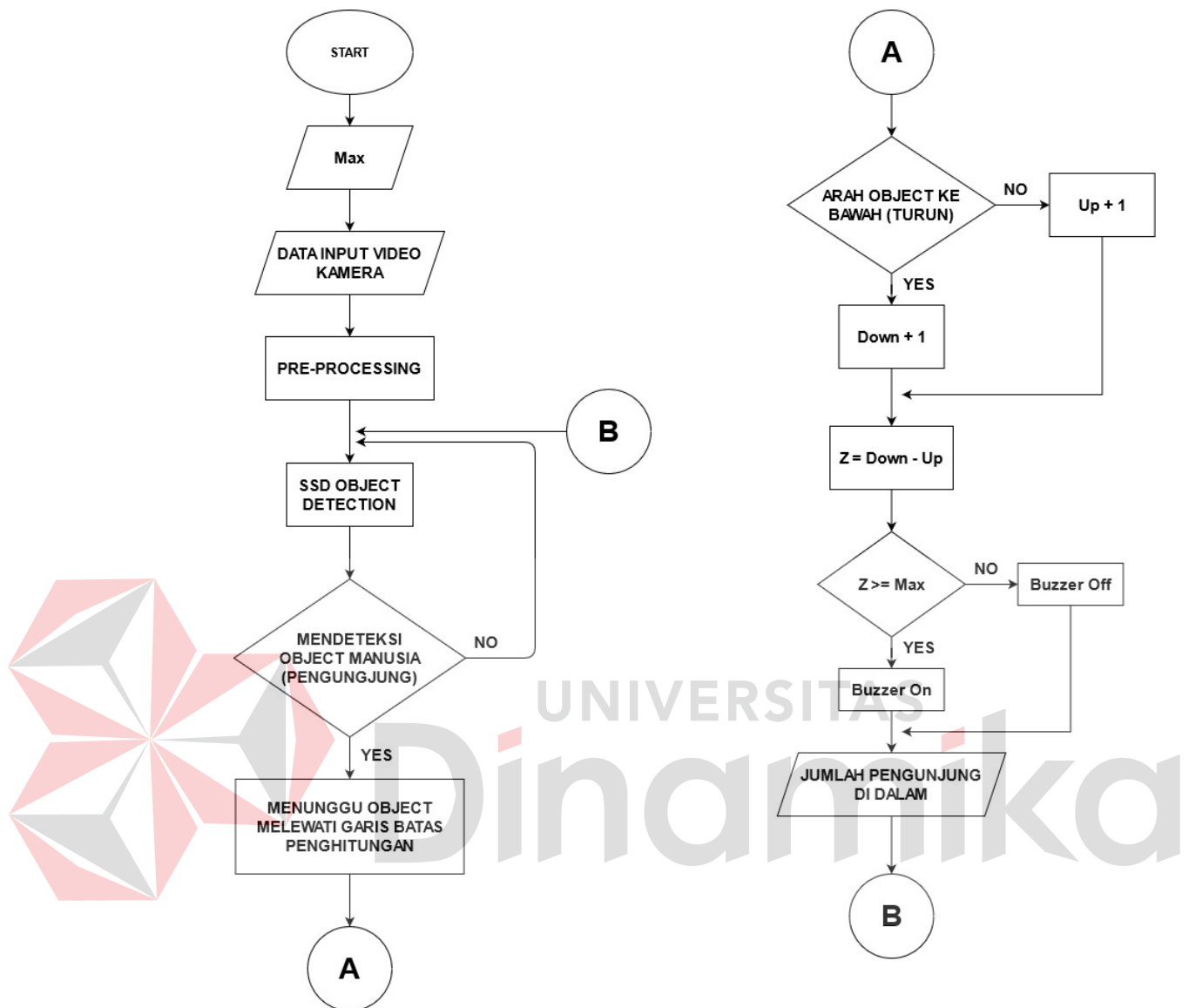
Gambar 3.1 Model perancangan

Pada Gambar 3.1, rancangan sistem diatas dapat dijelaskan bahwa data *input* berasal dari hasil rekaman kamera yang berupa video. Kemudian diproses oleh *computer vision* dengan menggunakan bahasa pemrograman *python* dan implementasi metode *Single Shot Detector (SSD)* dengan *dataset MobileNet*, sehingga didapatkan hasil dari *object* yang terbaca yaitu berupa *object* manusia.

Output berupa tampilan jam, hari, tanggal, tahun, hasil penghitungan jumlah pengunjung yang masuk, dan jumlah pengunjung yang masih ada di dalam area. Penghitungan tersebut dapat dihitung dengan rumus sederhana sebagai berikut:

$$(Jml\ org\ di\ dalam = Jml\ org\ masuk - Jml\ org\ keluar)$$

3.2 Perancangan Perangkat Lunak (*Software*)



Gambar 3.2 Algoritma sistem

Pada Gambar 3.2 menjelaskan bahwa ketika sistem dijalankan, maka pengguna atau *user* wajib memasukkan jumlah maksimal pengunjung sesuai dengan ketentuan dari restoran yang berlaku, kemudian data yang sudah dimasukkan, dijadikan variable **Max** yaitu batas maksimal jumlah pengunjung. Kemudian akan dilakukan ke proses pengambilan data *input* yang berupa video dari kamera yang sudah tersambung ke komputer atau menggunakan data video hasil rekaman, dan dilanjutkan ke proses *pre-processing* yaitu proses konversi menjadi image *greyscale* dan memperkecil nilai komputasinya, sehingga akan lebih mudah

dalam menganalisis *object* berupa manusia. Setelah itu, *object* yang sudah terdeteksi akan dilanjutkan ke proses *tracking* arah jalan dari *object* tersebut, sehingga dapat ditentukan suatu kondisi apakah *object* tersebut berjalan ke arah bawah (turun) atau berjalan ke atas (naik) dan melewati garis batas penghitungan. Ketika *object* berjalan turun dan melewati garis penghitungan, maka secara otomatis *counter* untuk variabel **Down** ditambah 1 (+1), namun ketika *object* berjalan naik dan melewati garis penghitungan, maka secara otomatis nilai dari variabel **Up** ditambah 1 (+1).

Ketika sudah didapatkan kedua data variable **Down** dan **Up**, maka dilanjutkan ke proses penghitungan jumlah pengunjung yang berada di dalam dengan rumus sederhana, yaitu nilai dari variabel **Down** akan dikurangkan dengan nilai dari variable **Up**, sehingga akan didapatkan nilai dari variable **Z** yang merepresentasikan jumlah dari pengunjung yang sedang berada di dalam restoran.

Setelah nilai dari variable **Z** didapatkan, selanjutnya akan dilakukan pengecekan dan perbandingan nilai dari variable **Z** dan variable **Max**, ketika nilai variable **Z** sama atau melebihi nilai variable **Max** maka secara otomatis *buzzer* akan berbunyi dan akan tampil *text* label yang menandakan jumlah pengunjung sudah mencapai batas maksimal. Dari semua data yang didapatkan oleh setiap *variable* akan ditampilkan melalui monitor komputer atau TV seperti seperti pada Gambar 3.3, ketika pengunjung sudah mencapai batas maksimal maka tampil *text* pemberitahuan seperti pada Gambar 3.4



Gambar 3.3 Tampilan aplikasi pada layar monitor TV atau PC



Gambar 3.4 Tampilan notifikasi ketika pengunjung mencapai batas maksimal

3.3 Persiapan Pembuatan Program

Dalam pembuatan aplikasi penghitung jumlah pengunjung di restoran diperlukan beberapa komponen pendukung agar dapat berjalan dengan lancar dan mudah dioperasikan. Berikut ini komponen pendukungnya:

1. Komputer

Spesifikasi komputer yang penulis gunakan adalah ASUS X455LJ dengan prosesor Intel(R) Core(TM) i3-5010U, RAM 4096MB, dan sistem operasi Windows 10 Pro 64-bit.

2. Layar Monitor

Spesifikasi layar monitor yang digunakan adalah monitor standard laptop yang penulis gunakan dengan pemroses grafik Intel(R) HD Graphics 5500.

3. Kamera

Pengambilan data *input* video yang digunakan dalam penelitian ini menggunakan *webcam* eksternal dari Logitech seri Stream HD Pro C922 dan seri B525 HD.

4. Aplikasi *python*

Pada penelitian ini penulis menggunakan aplikasi *python* versi 3.7.0.

5. *Python Library*

Python memiliki lebih dari 140.000 *library* tambahan yang dikembangkan dan dapat digunakan untuk keperluan tertentu, proses pengembangan *library* tersebut dapat dilakukan melalui *open source project*. Berikut ini adalah *library* yang penulis gunakan dalam pembuatan Tugas Akhir ini:

- A. *cv2 (OpenCV)*

cv2(OpenCV) adalah *library python* yang digunakan untuk pengolahan citradinamis secara *real-time* yang tefokus pada *image processing*, *video capture* dan analisis, termasuk fitur deteksi wajah dan deteksi *object*.

- B. *Dlib*

Dlib merupakan *library python* dengan bahasa *C++* dan bersifat *open source* yang mengimplementasikan berbagai algoritma *machine learning*, termasuk klasifikasi, regresi, pengelompokan, transformasi data, dan prediksi secara terstruktur.

C. *Time*

Library python yang digunakan untuk mendapatkan nilai detik, menit, jam, tanggal, bulan, dan tahun sesuai dengan waktu yang sedang berjalan.

D. *NumPy*

NumPy adalah akronim dari *Numerical Python*. Berfungsi untuk melakukan operasi vektor matriks dengan mengolah array dan array multidimensi. *NumPy* juga dapat digunakan untuk kebutuhan dalam analisis data.

E. *Imutils*

Imutils merupakan salah satu *library python* yang berfungsi untuk melakukan *image processing* seperti translasi, rotasi, perubahan ukuran, menampilkan kerangka, menampilkan gambar dari Matplotlib, klasifikasi kontur, dan deteksi tepi.

F. *TKinter*

TKinter merupakan *library python* yang digunakan untuk membuat tampilan *GUI (Graphical User Interface)*.

G. *CentroidTracker*

Merupakan *library python* yang digunakan untuk mendeteksi dan memberi titik tengah (x, y) pada *object* yang dideteksi pada setiap *frame*.

H. *TrackableObject*

Library python yang digunakan untuk meneteksi dan menghitung *object* dalam suatu video *stream* dengan memberi nomor identitas dan memprediksi arah pergerakan *object*.

I. *Sublime Text Ver. 3.2.2*

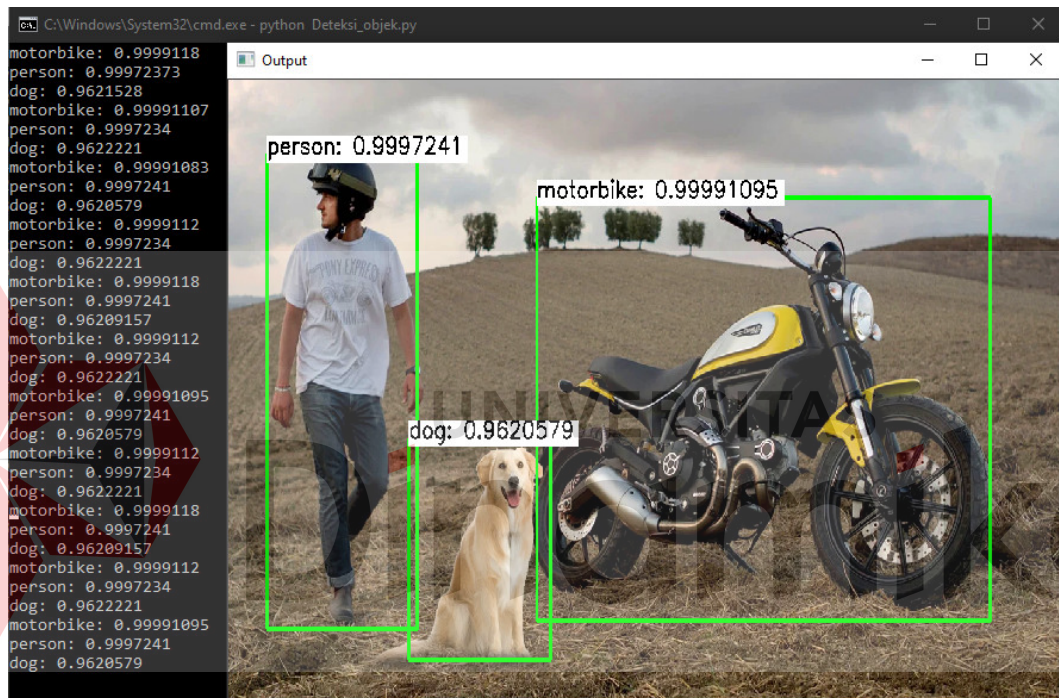
Software code editor yang dipakai penulis untuk membuat keseluruhan *code* program aplikasi.

3.4 Deteksi *Object*

Ketika pertama kali program dijalankan, maka secara otomatis program membuka file video yang berisikan 3 *object* yang berada dalam suatu *frame*. Proses

pendektesian *object* ini dilakukan dengan menggunakan metode *Single Shot Detector* (SSD) dengan *MobileNet* sebagai *dataset*.

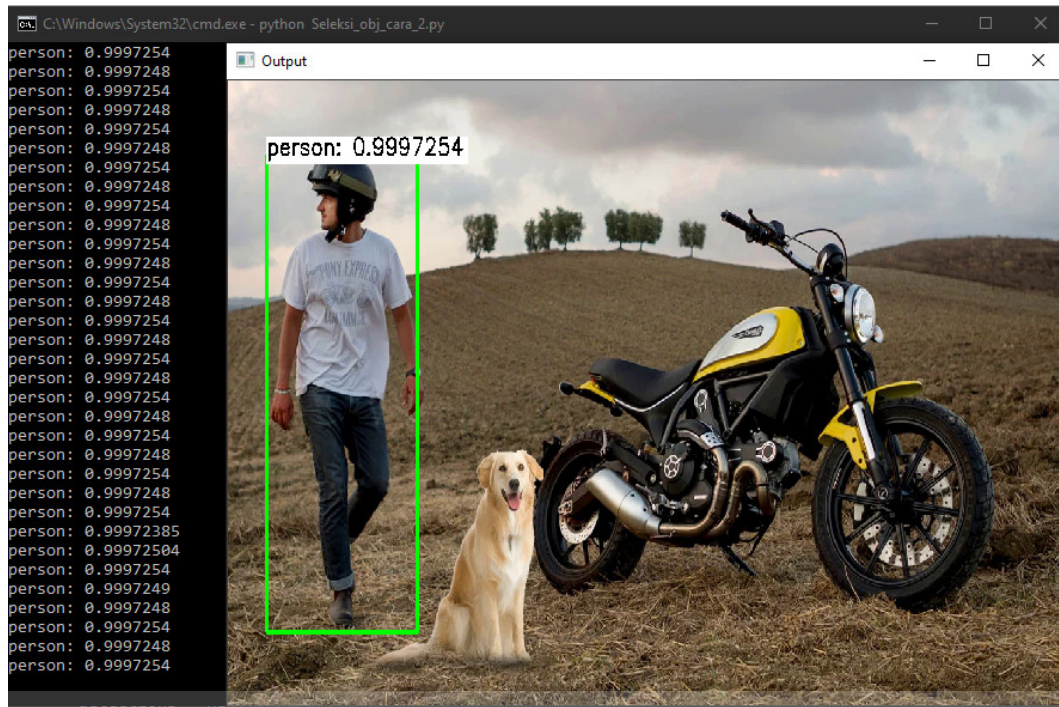
Dalam percobaan pada Gambar 3.5 diketahui data nilai akurasi dari pengenalan setiap *object* yaitu, sepeda motor (*motorbike*) dengan akurasi sebesar 100%, manusia (*person*) sebesar 100%, dan hewan (*dog*) sebesar 96%. Nilai akurasi tersebut merupakan nilai dari perbandingan antara citra dari video *input* dengan nilai citra yang ada pada *dataset MobileNet*.



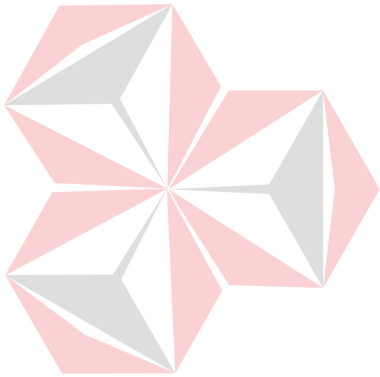
Gambar 3.5 Hasil pengujian deteksi *object*

3.5 Seleksi *Object* Manusia (*person*)

Untuk memaksimalkan kinerja aplikasi yang dibuat, maka diperlukan seleksi *object* yang bertujuan agar *object* yang dideteksi hanya manusia (*person*), sehingga dapat mempermudah proses penghitungan jumlah pengunjung yang keluar atau masuk. Dari hasil percobaan pada Gambar 3.6 dapat diketahui bahwa proses seleksi *object* manusia (*person*) sudah berjalan sesuai dengan kriteria yang diperlukan oleh aplikasi, tingkat akurasi dari pengenalan tersebut didapatkan di angka 100%.



Gambar 3.6 Hasil pengujian seleksi *object*



UNIVERSITAS
Dinamika

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang analisis dan hasil dari pengujian yang telah dilakukan. Berikut ini adalah tahapan yang dilakukan dalam pengujian pada Tugas Akhir ini:

4.1 Pengujian Tingkat Akurasi *Person Detection*

4.1.1 Tujuan Pengujian Tingkat Akurasi *Person Detection*

Tujuan pengujian ini adalah untuk mengetahui nilai akurasi sistem ketika mendeteksi *object* manusia (*person*) yang berada dalam suatu *frame* dengan kondisi *object* sedang berjalan, berjalan cepat, dan berlari.

4.1.2 Prosedur Pengujian Tingkat Akurasi *Person Detection*

1. Melakukan pengambilan data *input* berupa video dengan *object* berjalan, jalan cepat, dan berlari.
2. Membuat program pendeteksi *object* menggunakan metode *Single Shot Detector (SSD)* dengan integrasi *dataset MobileNet* menggunakan aplikasi *Sublime Text* dan file berekstensi *.py*.
3. Potongan *code* program terlampir pada Lampiran 1.

4.1.3 Hasil Pengujian Tingkat Akurasi *Person Detection*

Tabel 4.1 Tabel hasil pengujian akurasi *person detection*

Data Ke-	Kondisi <i>Object</i>	Tinggi Badan	Berat Badan	Hasil	
				Terdeteksi	Tidak
1	<i>Jalan</i>	175Cm	90Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>				✓
2	<i>Jalan</i>	178Cm	88Kg	✓	
	<i>Jalan cepat</i>				✓
	<i>Berlari</i>				✓
3	<i>Jalan</i>	174Cm	66Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>			✓	
4	<i>Jalan</i>	173Cm	60Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>				✓
5	<i>Jalan</i>	155Cm	88Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>				✓
6	<i>Jalan</i>	150Cm	62Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>			✓	
7	<i>Jalan</i>	170Cm	87Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>				✓
8	<i>Jalan</i>	176Cm	81Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>			✓	
8	<i>Jalan</i>	173Cm	69Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>			✓	
9	<i>Jalan</i>	172Cm	64Kg	✓	
	<i>Jalan cepat</i>			✓	
	<i>Berlari</i>			✓	
10	<i>Jalan</i>	157Cm	85Kg	✓	
	<i>Jalan cepat</i>			✓	

(Sumber: olahan pribadi)

Dari hasil pengujian pada Tabel 4.1, dapat diketahui bahwa hasil pengujian akurasi *person detection* masih ditemukan adanya *object* manusia (*person*) yang belum bisa terdeteksi dalam kondisi tertentu. Dengan adanya data pengujian yang

ada maka dapat dilanjutkan penghitungan tingkat akurasi menggunakan rumus [4.1] untuk setiap kondisi yaitu berjalan, jalan cepat, dan berlari.

$$\text{Akurasi Object} = \left(\frac{\text{terdeteksi}}{\text{banyakdatapercobaan}} * 100\% \right) \quad [4.1]$$

$$\begin{aligned} \text{Akurasi Object Berjalan} &= \left(\frac{10}{10} * 100\% \right) \\ &= 100\% \end{aligned}$$

Dari hasil penghitungan dengan rumus yang ada, ditemukan bahwa tingkat akurasi *person detection* ketika *object* dalam keadaan berjalan adalah 100%.

$$\begin{aligned} \text{Akurasi Object Berjalan Cepat} &= \left(\frac{9}{10} * 100\% \right) \\ &= 90\% \end{aligned}$$

Penghitungan selanjutnya adalah tingkat akurasi *person detection* ketika *object* dalam keadaan berjalan cepat. Tingkat akurasi yang diperoleh sebanyak 90%.

$$\begin{aligned} \text{Akurasi Object Berlari} &= \left(\frac{5}{10} * 100\% \right) \\ &= 50\% \end{aligned}$$

Penghitungan tingkat akurasi dilanjutkan dalam kondisi *object* sedang berlari. Hasil penghitungan ditemukan bahwa tingkat akurasi adalah 50%.

4.2 Pengujian Akurasi Penghitungan Jumlah Pengunjung

4.2.1 Tujuan Pengujian Akurasi Penghitungan Jumlah Pengunjung

Tujuan pengujian ini adalah untuk mengetahui tingkat akurasi penghitungan jumlah pengunjung yang sedang berada di dalam suatu area atau ruangan. Dengan menggunakan metode *Single Shot Detector* (SSD) dan integrasi *dataset MobileNet* maka dapat diketahui jumlah pengunjung keluar, masuk dan jumlah pengunjung yang berada di dalam suatu area, proses penghitungan pengunjung dieksekusi langsung oleh *counter*.

4.2.2 Prosedur Pengujian Akurasi Penghitungan Jumlah Pengunjung

1. Melakukan pengambilan data *input* berupa video.

2. Membuat program *python* pendeteksi *object* menggunakan metode *Single Shot Detector (SSD)* dengan integrasi *dataset MobileNet* menggunakan aplikasi *Sublime Text* dan file berekstensi *.py*.
3. Menambahkan baris *code* untuk *counter object* untuk menghitung jumlah pengunjung masuk, keluar, dan masih berada di dalam restoran.
4. Potongan *code* program *counter object* terlampir pada Lampiran 2.

4.2.3 Hasil Pengujian Akurasi Penghitungan Jumlah Pengunjung

Proses pengujian diawali pengambilan data dengan cara observasi jumlah pengunjung yang masuk, keluar, dan masih berada di dalam ruangan, kemudian mencocokkan antara hasil *counter* dari program dengan hitungan manual. Proses pengambilan data dilakukan dengan menggunakan video hasil rekaman penulis. Berikut ini adalah data dari hasil analisisnya:

Tabel 4.2 Hasil pengujian akurasi penghitungan

Percobaan Ke-	Hasil Penghitungan						Selisih Penghitungan		
	Aplikasi			Manual					
	Masuk	Keluar	Di dalam	Masuk	Keluar	Di dalam	Masuk	Keluar	Di dalam
1	7	1	6	8	1	7	1	0	1
2	13	1	12	13	2	11	0	1	1
3	21	1	20	23	1	22	2	0	2
4	29	1	28	31	1	30	2	0	2
5	36	3	33	38	3	35	2	0	2
6	46	5	41	51	4	47	5	1	4
7	51	8	43	58	4	54	7	4	11
8	58	9	49	66	4	62	8	5	13
9	62	9	53	71	5	66	9	4	13
10	67	9	58	78	5	73	11	4	15

Percobaan dilakukan dengan menghitung selisih penghitungan antara aplikasi dengan penghitungan manual, setiap percobaan dilakukan dengan durasi 1 menit dengan total akumulasi setiap menitnya dari 1 menit hingga 10 menit. Dari hasil pengujian pada Tabel 4.2, dapat disimpulkan bahwa hasil penghitungan antara *counter* program dengan penghitungan manual masih ditemukan adanya selisih. Dengan adanya data jumlah pengunjung yang ada maka dapat dilanjutkan penghitungan tingkat akurasi aplikasi dengan rumus [4.2].

$$\begin{aligned}
 \text{Akurasi Pengunjung Masuk} &= 100\% - \left(\frac{\text{selisih}}{\text{hitungmanual}} * 100\% \right) & [4.2] \\
 &= 100\% - \left(\frac{11}{78} * 100\% \right) \\
 &= 100\% - 14\% \\
 &= 86\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Akurasi Pengunjung Keluar} &= 100\% - \left(\frac{\text{selisih}}{\text{hitungmanual}} * 100\% \right) & [4.3] \\
 &= 100\% - \left(\frac{4}{9} * 100\% \right) \\
 &= 100\% - 44\% \\
 &= 66\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Akurasi Pengunjung di Dalam} &= 100\% - \left(\frac{\text{selisih}}{\text{hitungmanual}} * 100\% \right) & [4.4] \\
 &= 100\% - \left(\frac{15}{73} * 100\% \right) \\
 &= 100\% - 21\% \\
 &= 79\%
 \end{aligned}$$

4.3 Pengujian Notifikasi Kapasitas Maksimal

4.3.1 Tujuan Pengujian Notifikasi Kapasitas Maksimal

Tujuan pengujian ini adalah untuk mengetahui apakah program yang dibuat mampu memberikan pemberitahuan berupa *text* pesan yang berada di tengah layar monitor dan menyalakan *buzzer* dari komputer yang disambungkan ke speaker *external* ketika pengunjung sudah mencapai batas maksimal.

4.3.2 Prosedur Pengujian Notifikasi Kapasitas Maksimal

1. Melakukan pengambilan data *input* berupa video.
2. Membuat program pendeteksi *object* menggunakan metode *Single Shot Detector (SSD)* dengan integrasi *dataset* dari *MobileNet* menggunakan aplikasi *Sublime Text* dan file berekstensi *.py*.
3. Menambahkan baris *code* untuk menampilkan pesan *notifikasi* dan menyalakan *buzzer* yang ada pada komputer atau laptop.

4. Potongan *code* program *counter object* terlampir pada Lampiran 3.

4.3.3 Hasil Pengujian Notifikasi Kapasitas Maksimal

Setelah melakukan pengujian sesuai dengan prosedur diatas, berikut ini adalah hasil yang didapat dan disajikan pada tabel 4.3:

Tabel 4.3 Hasil pengujian notifikasi

Percobaan Ke-	Jumlah Pengunjung		Notifikasi	
	Terdeteksi	Batas Maksimal	Text Pesan	Suara <i>Buzzer</i>
1	1 Orang	1 Orang	Tampil	Aktif
2	2 Orang	2 Orang	Tampil	Aktif
3	3 Orang	3 Orang	Tampil	Aktif
4	4 Orang	4 Orang	Tampil	Aktif
5	5 Orang	5 Orang	Tampil	Aktif
6	6 Orang	6 Orang	Tampil	Aktif
7	7 Orang	7 Orang	Tampil	Aktif
8	8 Orang	8 Orang	Tampil	Aktif
9	9 Orang	9 Orang	Tampil	Aktif
10	10 Orang	10 Orang	Tampil	Aktif
11	11 Orang	11 Orang	Tampil	Aktif
12	12 Orang	12 Orang	Tampil	Aktif
13	13 Orang	13 Orang	Tampil	Aktif
14	14 Orang	14 Orang	Tampil	Aktif
15	15 Orang	15 Orang	Tampil	Aktif
16	16 Orang	16 Orang	Tampil	Aktif
17	17 Orang	17 Orang	Tampil	Aktif
18	18 Orang	18 Orang	Tampil	Aktif
19	19 Orang	19 Orang	Tampil	Aktif
20	20 Orang	20 Orang	Tampil	Aktif
21	21 Orang	21 Orang	Tampil	Aktif
22	22 Orang	22 Orang	Tampil	Aktif
23	23 Orang	23 Orang	Tampil	Aktif
24	24 Orang	24 Orang	Tampil	Aktif
25	25 Orang	25 Orang	Tampil	Aktif
26	26 Orang	26 Orang	Tampil	Aktif
27	27 Orang	27 Orang	Tampil	Aktif
28	28 Orang	28 Orang	Tampil	Aktif
29	29 Orang	29 Orang	Tampil	Aktif
30	30 Orang	30 Orang	Tampil	Aktif

Dari hasil percobaan tersebut dapat diketahui bahwa sistem notifikasi untuk menampilkan *text* pesan dan menyalakan *buzzer* mampu berjalan normal sesuai dengan yang dikehendaki. Gambar cuplikan hasil pengujian terlampir pada lampiran Gambar L5, Gambar L6, dan Gambar L7. Nilai akurasi dari notifikasi ketika kapasitas mencapai batas maksimal dapat dihitung menggunakan rumus [4.5]. Tingkat akurasi yang didapatkan adalah 100%.

$$\begin{aligned} \text{Akurasi} &= \left(\frac{\text{berhasil}}{\text{total percobaan}} * 100\% \right) & [4.5] \\ &= \left(\frac{30}{30} * 100\% \right) \\ &= 100 \% \end{aligned}$$

4.4 Pengujian Akurasi *Data Reporting*

4.4.1 Tujuan Pengujian Akurasi *Data Reporting*

Pengujian ini dilakukan dengan mencatat jumlah pengunjung yang masuk, keluar, dan masih berada di dalam ruangan dalam kurun waktu tertentu. Hasil pencatatan akan disajikan dalam bentuk file *spreadsheet*.

4.4.2 Prosedur Pengujian Akurasi *Data Reporting*

1. Melakukan pengambilan data *input* berupa video.
2. Membuat program pendeteksi *object* menggunakan metode *Single Shot Detector* (SSD) dengan integrasi *dataset* dari *MobileNet* menggunakan aplikasi *Sublime Text* dan file berekstensi *.py*
3. Menambahkan baris *code* untuk membuat file *reporting* berupa file *spreadsheet* dengan ekstensi *.csv*.
4. Potongan *code* program *counter object* terlampir pada Lampiran 4.

4.4.3 Hasil Pengujian Akurasi *Data Reporting*

Dari hasil pengujian yang sudah dilakukan, disimpulkan bahwa nilai yang tercatat pada aplikasi dan nilai yang tercatat pada *data report* file *.csv* sudah sesuai

tanpa adanya selisih. Detail hasil pengujian dituliskan pada Tabel 4.4. Hasil akurasi data yang didapatkan adalah 100%. Cuplikasn hasil pengujian terlampir pada Gambar L8, dan Gambar L9.

Tabel 4.4 Hasil pengujian akurasi *data reporting*

Percobaan Ke-	Penghitungan Aplikasi			<i>Data Report</i>		
	Masuk	Keluar	Di dalam	Masuk	Keluar	Di dalam
1	8	1	7	8	1	7
2	13	1	12	13	1	12
3	22	2	20	22	2	20
4	29	3	26	29	3	26
5	38	4	34	38	4	34
6	49	8	41	49	8	41
7	56	8	48	56	8	48
8	61	10	51	61	10	51
9	66	10	56	66	10	56
10	66	15	51	66	15	51
11	66	21	45	66	21	45
12	67	26	41	67	26	41
13	67	33	34	67	33	34
14	67	42	25	67	42	25
15	67	52	15	67	52	15
16	68	57	11	68	57	11
17	68	65	3	68	65	3
18	68	68	0	68	68	0

BAB V

PENUTUP

5.1 Kesimpulan

1. Hasil dari pengujian yang telah dilakukan, dapat disimpulkan bahwa Sistem yang telah dibuat sudah mampu mendeteksi pengunjung dengan 3 macam kondisi yaitu berjalan dengan akurasi 100%, berjalan cepat dengan akurasi 90% dan berlari dengan akurasi 50%.
2. Tingkat akurasi penghitungan dengan kondisi ramai dan pengunjung lalu lalang seperti yang didapat dari video *input* memiliki hasil 86% untuk menghitung pengunjung masuk, 66% untuk menghitung pengunjung keluar, dan 79% untuk menghitung jumlah pengunjung.
3. Sistem notifikasi mampu berjalan sesuai dengan kondisi yang ditentukan.
4. Hasil *data reporting* memiliki nilai yang sama dengan tampilan penghitungan pada aplikasi, *data reporting* mencatat setiap *frame*.

5.2 Saran

Untuk penelitian selanjutnya terdapat beberapa saran yang dapat diimplementasikan untuk mengembangkan aplikasi ini:

1. Meningkatkan efisiensi *reporting* dengan menjadikan data pada *file .csv* menjadi lebih sedikit yang mencatat data setiap menit atau jam, bukan mencatat setiap *frame* seperti saat ini.
2. Meningkatkan tampilan GUI (*Graphical User Interface*) agar lebih interaktif dan mudah dipahami oleh pengguna.
3. Posisi pemasangan kamera akan lebih maksimal ketika berada di ketinggian 3 meter sampai 4 meter dari lantai dengan sudut kamera 45° menghadap ke bawah, sehingga POV (*Point Of View*) kamera menjadi lebih luas.
4. Mengintegrasikan sistem penghitungan jumlah pengunjung dengan sistem pendeteksi suhu dan sistem pendeteksi jarak antar pengunjung (*physical distancing*) agar menjadi suatu sistem yang kompleks dan lebih efektif.

DAFTAR PUSTAKA

- Chen Z., K. R. (2019). Real Time Object Detection, Tracking, and Distance and Motion Estimation based on Deep Learning: Application to Smart Mobility. *Proceedings of the 2019 Eighth International Conference on Emerging Security Technologies (EST)*, 1–6.
- Ducharme. (2020, 03 11). *www.time.com*. Retrieved 02 23, 2021, from <https://time.com/5791661/who-coronavirus-pandemic-declaration/>
- Ellyvon. (2020, 05 11). *www.kompas.com*. Retrieved 02 23, 2021, from www.kompas.com/sains/read/2020/05/11/130600623/diumumkan-awal-maret-ahli-virus-corona-masuk-indonesia-dari-januari
- Fajirwan, D. (2017, 10). *http://derryfajirwan.blogspot.com*. Retrieved 02 20, 2021, from <http://derryfajirwan.blogspot.com/2017/10/pengenalan-opencv.html>
- Kusno Prasetya, A. A. (2020). Implementation of Tensorflow in the CCTV-Based People Counter Application at PT Matahari Department Store, Tbk. *IAIC International Conferences, Vol. 3*, 38-44.
- Lutz, M. (2010). *Programming Python. Fourth Edition ed. Sebastopo*. O'Reilly Media, Inc.
- R. G. Fajri, I. S. (2021). PERANCANGAN PROGRAM PENDETEKSI DAN PENGKLASIFIKASI JENIS KENDARAAN DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) DEEP LEARNING. *ransient: Jurnal Ilmiah Teknik Elektro, vol. 1*.
- Sik-Ho Tsang, 2. (2018, 11 03). *https://towardsdatascience.com*. Retrieved 02 20, 2021, from <https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11>
- Ulie, Y. A. (2021, 01 12). *www.money.kompas.com*. Retrieved 02 23, 2021, from <https://money.kompas.com/read/2021/01/12/183314826/pengunjung-restoran-kian-dibatasi-selama-ppkm-begini-keluh-kesah-pengusaha?page=all>