



**SISTEM DETEKSI KAPASITAS ORANG DI DALAM RUANGAN  
MENGUNAKAN METODE FASTER R-CNN**



**TUGAS AKHIR**

**Program Studi  
S1 TEKNIK KOMPUTER**

**Oleh:  
SELLINA NURIL LAILI  
18410200057**

---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA  
2022**

**SISTEM DETEKSI KAPASITAS ORANG DI DALAM RUANGAN  
MENGUNAKAN METODE FASTER R-CNN**

**TUGAS AKHIR**

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



**UNIVERSITAS**  
**Dinamika**

**Disusun Oleh:**

**Nama : Sellina Nuril Laili**

**NIM : 18410200057**

**Program Studi : S1 Teknik Komputer**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2022**

## TUGAS AKHIR

### SISTEM DETEKSI KAPASITAS ORANG DI DALAM RUANGAN MENGUNAKAN METODE FASTER R-CNN

Dipersiapkan dan disusun oleh:

**Sellina Nuril Laili**

**NIM : 18410200057**

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 14 Januari 2022

#### Susunan Dewan Pembahas

##### Pembimbing:

I. **Heri Pratikno, M.T., MTCNA., MTCRE.**

NIDN: 0716117302


II. **Hariato, S.Kom., M.Eng.**

NIDN: 0722087701

##### Pembahas:

**Dr. Jusak**

NIDN: 0708017101



Digitally signed by  
Universitas Dinamika  
Date: 2022.01.24  
22:04:09 +07'00'



DN: cn=Hariato, c=ID, o=Universitas  
Dinamika, ou=Fakultas Teknologi dan  
Informatika, email=harario@dinamika.ac.id  
Date: 2022.01.26 09:56:52 +07'00'



Digitally signed  
by Jusak Jusak  
Date: 2022.01.26  
09:56:14 +08'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana



Digitally signed by  
Universitas Dinamika  
Date: 2022.01.27  
16:06:52 +07'00'

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

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika  
UNIVERSITAS DINAMIKA

**“Cintai diriku sendiri, Cintai dirimu sendiri, Damai”**

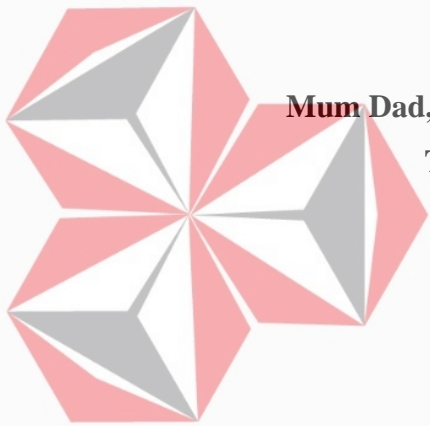
**~ Min Yoongi~**



UNIVERSITAS  
**Dinamika**

**Mum Dad, Tugas Akhir ini kupersembahkan untuk kalian dan demi kalian.**

**Terima kasih untuk segalanya. Aku menyayangi kalian.**



UNIVERSITAS  
**Dinamika**

**SURAT PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : Sellina Nuril Laili  
NIM : 18410200057  
Program Studi : SI Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Laporan Tugas Akhir  
Judul Karya : **SISTEM DETEKSI KAPASITAS ORANG DI DALAM  
RUANGAN MENGGUNAKAN METODE FASTER R-  
CNN**

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 Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas 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 diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi 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, 16 Desember 2021


Sellina Nuril Laili  
NIM : 18410200057

## ABSTRAK

Upaya pencegahan penularan virus Covid-19 dilakukan terus menerus oleh pemerintah, salah satunya adalah memberlakukan program PSBB (Pembatasan Sosial Berskala Besar). Program ini bertujuan untuk membatasi kegiatan sosial masyarakat seperti meliburkan tempat kerja. Oleh sebab itu dibuat Keputusan Menteri Kesehatan Republik Indonesia Nomor HK.01.07/MENKES/328/202. Dalam keputusan ini terdapat panduan pencegahan dan pengendalian Covid-19 di tempat kerja perkantoran dan industri dalam mendukung keberlangsungan usaha pada situasi pandemi, salah satunya adalah menerapkan pembatasan jumlah orang dalam suatu ruangan dengan menerapkan sistem *shift* WFH (*Work From Home*) dan WFO (*Work From Office*). Dari permasalahan ini, penulis memberikan solusi untuk membantu meminimalisir penyebaran Covid-19 melalui layanan dan penerapan di bidang teknologi. Pada Tugas Akhir ini, penulis membuat sistem untuk menghitung jumlah orang masuk, keluar, dan jumlah orang yang berada di dalam ruangan secara otomatis. Sistem diproses dengan pengolahan citra digital menggunakan metode Faster R-CNN dengan bahasa *Python*. Metode Faster R-CNN yang digunakan merupakan salah satu metode *deep learning* yang berfungsi mengenali objek pada suatu citra atau gambar yang ditangkap oleh kamera. Dengan menggunakan metode ini sistem hanya dapat mendeteksi *object* orang saja, hasil pengujian yang didapatkan dari penelitian ini menunjukkan bahwa sistem mampu mendeteksi tiga pergerakan keberadaan orang yang berjalan lambat, berjalan normal dan berjalan cepat yang berada di POV (*Point Of View*) kamera dengan tingkat akurasi 77%. Sedangkan tingkat akurasi perhitungan sistem untuk jumlah orang yang masuk, keluar, dan jumlah orang yang berada di dalam ruangan dengan hasil data *reporting* yang disimpan dalam bentuk file *spreadsheet* (.csv) memiliki kesesuaian data 100%, data *reporting* pada sistem ini dicatat setiap detik dalam waktu kurang lebih 20 menit. Sistem (model) yang dibuat oleh penulis memiliki nilai *Accuracy* sebesar 100%, memiliki nilai *Precision* sebesar 1, nilai *Recall* sebesar 1, nilai *F1 – Score* sebesar 1, nilai *Rate Accuracy* sebesar 79% dan *Rate Loss* sebesar 21%.

**Kata Kunci :** *Faster R-CNN, Pembatasan Jumlah Orang, Deteksi Orang, Pembatasan Sosial, Covid-19*



## KATA PENGANTAR

Syukur alhamdulillah, penulis ucapkan atas rahmad dan ridha Allah Yang Maha Esa, karena atas nikmat dan kuasanya penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul "Sistem Deteksi Kapasitas Orang Di Dalam Ruangan Menggunakan Metode Faster R-CNN" tepat pada waktunya, sebagai sebuah syarat untuk lulus dan mendapatkan gelar Sarjana Teknik (S.T.).

Pada kesempatan ini, penulis mengucapkan terima kasih pada berbagai pihak yang membimbing dan memberi dukungan selama proses mengerjakan Tugas Akhir dan menyusun laporan ini :

1. Orang Tua, yang telah memberikan dukungan besar hingga penulis dapat menyelesaikan Laporan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
4. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembimbing I yang selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
5. Bapak Harianto, S.Kom., M.Eng., selaku Dosen Pembimbing II yang juga selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
6. Miskiyanto dan Frisil, selaku kakak tingkat, yang membantu mengambil data serta memberikan semangat dan dukungannya agar penulis dapat menyelesaikan Tugas Akhir beserta laporan ini.
7. Seluruh teman-teman S1 Teknik Komputer angkatan 2018 yang memberikan semangat pantang menyerah dan selalu menemani selama proses mengerjakan Tugas Akhir ini.

Penulis menyadari, masih banyak kesalahan yang terdapat dalam penulisan laporan Tugas Akhir ini, suatu kehormatan bila pembaca dapat memberikan saran dan kritik, sehingga dapat memperbaiki kekurangan dan berusaha untuk lebih baik



lagi. Penulis juga berharap, semoga laporan ini berguna, bermanfaat, serta menambah wawasan bagi pembacanya. Terimakasih

Surabaya, 15 Desember 2021

Penulis



UNIVERSITAS  
**Dinamika**

## DAFTAR ISI

	Halaman
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xiv
DAFTAR LAMPIRAN .....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	3
BAB II LANDASAN TEORI .....	4
2.1 Webcam .....	4
2.2 <i>Deep Learning</i> .....	5
2.3 Bahasa <i>Python</i> .....	5
2.3.1 OpenCV .....	6
2.3.2 <i>Numerical Python</i> (NumPy) .....	6
2.3.3 Imutils .....	6
2.4 <i>Object Detection</i> .....	6
2.5 VGG-16 .....	7
2.6 Faster R-CNN .....	8
2.6.1 <i>Convolutional Layer (Feature Extraction)</i> .....	10

2.6.2	<i>Region Proposal Network (RPN)</i> .....	11
2.6.3	<i>Region Of Interest (RoI) Pooling</i> .....	15
2.6.4	<i>Classification</i> .....	16
2.7	<i>Confusion Matrix</i> .....	16
2.7.1	<i>Jenis Confusion Matrix</i> .....	18
2.7.2	<i>Performance Metric dari Confusion Matrix</i> .....	18
BAB III METODOLOGI PENELITIAN.....		21
3.1	<i>Rancangan Hardware</i> .....	21
3.2	<i>Flowchart Sistem</i> .....	23
BAB IV HASIL DAN PEMBAHASAN .....		25
4.1	<i>Pengujian Deteksi Orang</i> .....	25
4.1.1	<i>Tujuan Pengujian Deteksi Orang</i> .....	25
4.1.2	<i>Prosedur Pengujian Deteksi Orang</i> .....	25
4.1.3	<i>Hasil Pengujian Deteksi Orang</i> .....	26
4.2	<i>Pengujian Deteksi Object</i> .....	27
4.2.1	<i>Tujuan Pengujian Deteksi Object</i> .....	27
4.2.2	<i>Prosedur Pengujian Deteksi Object</i> .....	28
4.2.3	<i>Hasil Pengujian Deteksi Object</i> .....	28
4.3	<i>Pengujian Akurasi Perhitungan Data Report</i> .....	31
4.3.1	<i>Tujuan Pengujian Akurasi Perhitungan Data Report</i> .....	31
4.3.2	<i>Prosedur Pengujian Akurasi Data Report</i> .....	31
4.3.3	<i>Hasil Pengujian Akurasi Data Report</i> .....	31
4.4	<i>Perhitungan Confusion Matrix</i> .....	33
4.4.1	<i>Tujuan Perhitungan Confusion Matrix</i> .....	33
4.4.2	<i>Prosedur Perhitungan Confusion Matrix</i> .....	33
4.4.3	<i>Hasil Perhitungan Confusion Matrix</i> .....	33

BAB V PENUTUPAN .....	37
5.1    Kesimpulan.....	37
5.2    Saran .....	37
DAFTAR PUSTAKA .....	39
BIODATA PENULIS .....	51



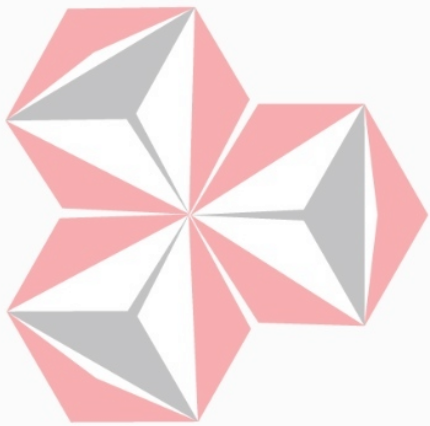
UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	Halaman
Gambar 1. 1 Grafik kasus positif Covid-19 per-hari Tahun 2021 .....	1
Gambar 2. 1 Webcam Logitech C310.....	4
Gambar 2. 2 Logo Python .....	5
Gambar 2. 3 Arsitektur VGG-16.....	7
Gambar 2. 4 Bagam Faster R-CNN .....	8
Gambar 2. 5 Struktur Diagram Faster R-CNN.....	9
Gambar 2. 6 Proses perubahan ukuran gambar .....	10
Gambar 2. 7 Struktur RPN .....	11
Gambar 2. 8 Struktur <i>anchor boxes</i> .....	11
Gambar 2. 9 Jumlah <i>anchor</i> pada gambar asli .....	12
Gambar 2. 10 Alur proses <i>softmax determine</i> .....	13
Gambar 2. 11 Regresi <i>Bounding Box</i> .....	14
Gambar 2. 12 Struktur RoI <i>Pooling Layer</i> .....	15
Gambar 2. 13 Struktur <i>Classification Layer</i> .....	16
Gambar 2. 14 Tabel <i>Confusion Matrix</i> .....	17
Gambar 3. 1 Model Perancangan <i>Hardware</i> .....	21
Gambar 3. 2 Tampilan <i>output</i> pada layar monitor .....	22
Gambar 3. 3 <i>Flowchart</i> sistem deteksi kapasitas orang .....	23

## DAFTAR TABEL

	Halaman
Tabel 4. 1 Hasil Uji Deteksi Orang .....	26
Tabel 4. 2 Hasil Uji Deteksi <i>Object</i> .....	28
Tabel 4. 3 Hasil Uji Akurasi Data <i>Report</i> .....	31
Tabel 4. 4 <i>Confusion Matrix</i> .....	33



UNIVERSITAS  
**Dinamika**

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 <i>Source Code</i> Program.....	41
Lampiran 2 Bukti Cek Plagiasi Buku TA .....	46
Lampiran 3 Bukti Cek Plagiasi Jurnal .....	49



UNIVERSITAS  
**Dinamika**

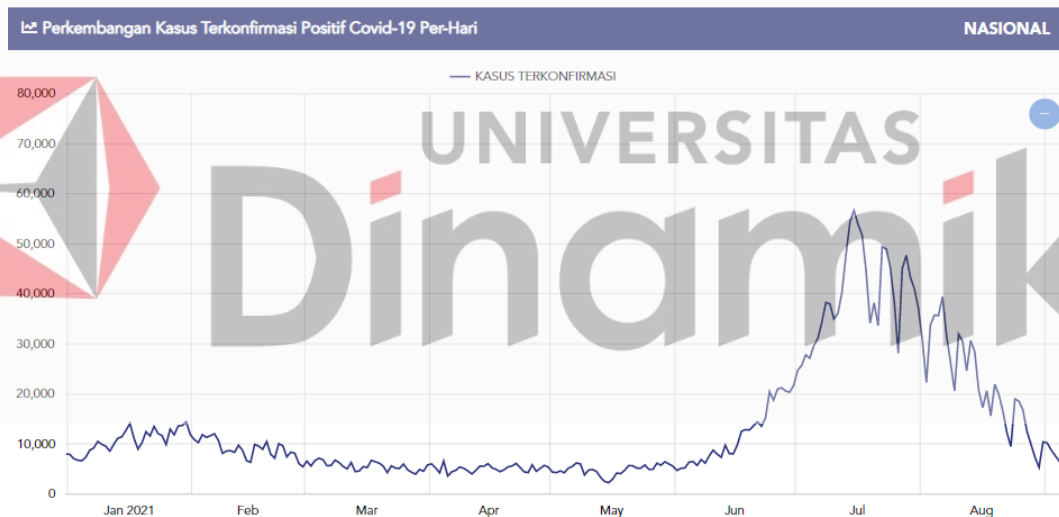


# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pada tahun 2021 di pertengahan bulan Juni hingga Agustus, Indonesia mengalami kelonjakan kasus meninggal perharinya yang diakibatkan oleh Covid-19 (Kemenkes, 2020). Hal ini disebabkan oleh kurangnya kesadaran masyarakat akan pentingnya untuk menerapkan protokol kesehatan. Dari awal pandemic melanda Indonesia sampai saat ini, banyak masyarakat yang belum mempercayai adanya Covid-19, sehingga banyak yang tidak mengikuti aturan pemerintah yang dibuat untuk menekan penyebaran Covid-19 di Indonesia.



Gambar 1. 1 Grafik kasus positif Covid-19 per-hari Tahun 2021

(Sumber : <https://covid19.go.id/peta-sebaran>)

Pembatasan Sosial Berskala Besar (PSBB) merupakan upaya pemerintah untuk memutus penyebaran Covid-19, tujuannya untuk membatasi kegiatan sosial yang dilakukan oleh masyarakat seperti meliburkan tempat kerja. Tetapi dunia kerja selamanya tidak dapat dibatasi kegiatannya, karena roda perekonomian harus tetap berjalan meskipun pandemi masih berlangsung. Oleh sebab itu Keputusan Menteri Kesehatan Republik Indonesia Nomor HK.01.07/MENKES/328/2020 membuat panduan pencegahan dan pengendalian Covid-19 di tempat kerja perkantoran dan

industri dalam mendukung keberlangsungan usaha pada situasi pandemi. Dalam panduan tersebut disampaikan bahwa mitigasi serta kesiapan tempat kerja harus dilakukan seoptimal agar dapat beradaptasi dengan perubahan pola hidup pada masa pandemi. Salah satunya adalah dengan penerapan pembatasan jumlah orang dalam suatu ruangan yang dilakukan dengan menetapkan WFH (*work from home*) yaitu menentukan pekerja yang tetap datang ke kantor serta pekerja yang dapat bekerja di rumah (Kemenkes RI, 2020).

Berdasarkan permasalahan diatas perlu dibuat deteksi jumlah orang pada ruangan kantor untuk membatasi jumlah orang pada ruangan tersebut dengan harapan resiko penyebaran Covid-19 pada dunia kerja dapat diminimalisir sekecil mungkin. Pada penelitian sebelumnya dengan judul “Sistem Penghitung Jumlah Pengunjung Di Restoran Menggunakan Kamera Berbasis Single Shot Detector (SSD)” mampu membedakan *object* orang (*person*) dengan *object* lain, sehingga dengan mudah menghitung jumlah pengunjung suatu restoran secara otomatis.

Penelitian ini menggunakan proses pengolahan citra digital dengan metode SSD dalam bahasa pemrograman python. Hasil perhitungannya ditampilkan di layar bagian kanan atas serta disimpan dalam bentuk file *spreadsheet* (.csv) (Admaja, 2021). Penelitian ini memiliki kekurangan yaitu sistem dapat mendeteksi *object* orang jika *object* tersebut *full body*, sehingga jika ada salah satu orang yang berhimpitan, tidak dapat terhitung, karena harus tampak *full body*.

Oleh sebab itu, penelitian Tugas Akhir ini dilakukan dengan tujuan untuk menyempurnakan penelitian sebelumnya, yaitu dengan mendeteksi orang menggunakan metode Faster R-CNN. Metode Faster R-CNN ini mampu melakukan proses deteksi lebih cepat dengan hasil yang lebih akurat. Proses deteksi orang pada Tugas Akhir ini dilakukan secara langsung melalui kamera atau menggunakan file video secara *realtime* menggunakan *deep learning*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut :

1. Bagaimana mendeteksi *object* orang di dalam ruangan dengan menggunakan metode Faster R-CNN ?
2. Bagaimana menghitung jumlah orang masuk, keluar dan jumlah orang dalam ruangan ?

### 1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapa hal berikut :

1. Pencahayaan adalah pencahayaan ruangan atau yang merata.
2. Reporting data dicatat setiap 4 jam sekali dalam 8 jam kerja dan disimpan dalam bentuk *file* .csv
3. Kondisi awal ruangan harus kosong.

### 1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas, tujuan pada Tugas Akhir ini adalah :

1. Sistem mampu mendeteksi *object* orang di dalam ruangan dengan menggunakan metode Faster R-CNN.
2. Sistem mampu menghitung jumlah orang masuk, keluar dan jumlah orang dalam ruangan.

### 1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah sebagai berikut :

- 1 Menambah ilmu pengetahuan terkait penerapan sistem deteksi orang pada ruangan menggunakan model Faster R-CNN.
- 2 Dapat menjadi referensi bagi mahasiswa yang membutuhkannya.
- 3 Membantu menekan penyebaran virus Covid-19.

## BAB II

### LANDASAN TEORI

#### 2.1 Webcam

Webcam berasal dari gabungan kata web dan *camera*. Webcam merupakan sebutan untuk kamera *real-time*, dimana gambarnya dapat diakses dan dilihat melalui internet. Webcam adalah jenis kamera yang digunakan untuk kebutuhan layanan berbasis web seperti untuk keperluan konferensi jarak jauh, terkadang juga digunakan sebagai kamera pemantau. Webcam merupakan perangkat yang mengambil gambar/video dan dapat dihubungkan ke komputer atau laptop. Cara kerja webcam yaitu menangkap cahaya melalui lensa yang berukuran kecil yang dibantu detector cahaya mikroskopik yang terpasang di *microchip* penerima gambar, umumnya berteknologi *Charge-Couple Device* (CCD) atau *CMOS image sensor* (Elekomp, 2018). Di bawah ini adalah contoh gambar webcam yang tampak pada Gambar 2.1.



Gambar 2. 1 Webcam Logitech C310

(Sumber : <https://www.plazakamera.com/shop/logitech-c310-hd-webcam/>)

Pada penelitian tugas akhir ini, penulis menggunakan webcam Logitech C310, yang memiliki fitur teknologi *RightLight 2 built-in* sehingga dapat merekam video dan mengambil gambar sebaik mungkin (Plazakamera.com, n.d.).

## 2.2 *Deep Learning*

*Deep Learning* merupakan metode pembelajaran mesin yang membuat computer mampu mempelajari tugas-tugas yang ditugaskan mengikuti sifat orang. *Deep Learning* adalah teknologi yang mendukung kecerdasan buatan (AI) dengan kemajuannya di berbagai bidang. *Deep Learning* memiliki kemampuan untuk mengenali tanda-tanda berhenti, membedakan suatu objek dengan objek lainnya, serta berperan penting dalam mengenali suara seperti suara yang terdapat di Ponsel, Televisi, dan sebagainya (Sena, 2018).

## 2.3 *Bahasa Python*



*Gambar 2. 2 Logo Python*

(Sumber : <https://logos-world.net/wp-content/uploads/2021/10/Python-Symbol.png>)

Bahasa *Python* merupakan salah satu bahasa pemrograman yang melakukan eksekusi dari beberapa instruksi secara langsung dengan menggunakan metode orientasi objek (*Object Orientation Programming*) serta *semantic dynamic* untuk meningkatkan kemampuan membaca *syntax*. Dalam tugas akhir ini, penulis menggunakan bahasa *Python*, dengan memanfaatkan beberapa library yang disediakan oleh *Python*.

### 2.3.1 OpenCV

OpenCV merupakan *library* yang berfungsi untuk mengelola gambar dan video, bersifat *opensource* yang artinya gratis, tidak berbayar, dan dapat di download oleh siapapun. Kepanjangan dari “CV” adalah *Computer Vision*, artinya computer yang digunakan untuk mengelola citra/gambar yang ditangkap oleh kamera/webcam kemudian melakukan konversi *analog to digital* dan diolah di dalam computer. Salah satu tujuan pengolahan citra/gambar ini adalah memperbaiki kualitas gambar serta mengidentifikasi gambar (Kandir, 2016).

### 2.3.2 Numerical Python (NumPy)

*Numerical Python* merupakan salah satu *library python* yang terarah pada *scientific computic*, aljabar linear, operasi vector (1-d array), dan matrix (2-d array). NumPy memiliki sebuah kemampuan untuk membentuk objek N-dimensional array. Pada *Python* selain NumPy, terdapat juga *library list* dengan kemampuan yang mirip dengan NumPy. Tetapi NumPy lebih unggul dari *List*, salah satu keunggulannya adalah NumPy mengkonsumsi *memory* lebih kecil, dan *runtime* yang lebih cepat.

### 2.3.3 Imutils

Imutils merupakan salah satu alternatif *library* dari OpenCV. Umumnya *library* imutils ini digunakan untuk menampilkan video. Selain itu *library* imutils juga memiliki berbagai macam kegunaan diantaranya yaitu , melakukan *translation, rotation, resizing* dan *skeletonization*.

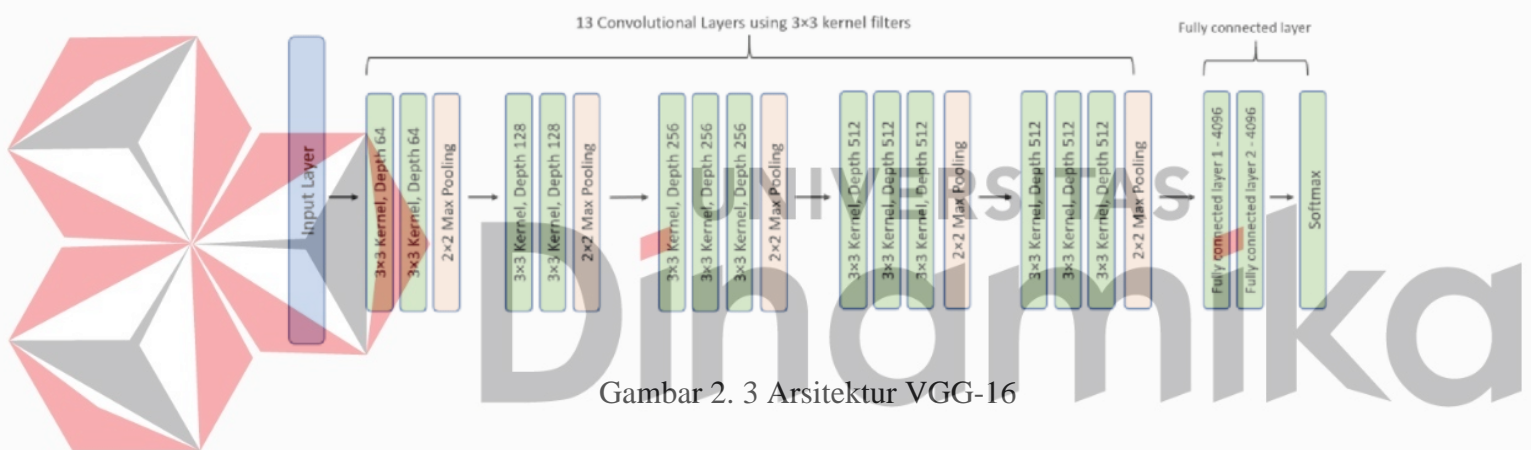
## 2.4 Object Detection

*Object Detection* merupakan sebuah metode yang digunakan untuk mendeteksi *object* pada suatu gambar atau video. Pada penelitian ini, penulis menggunakan metode ini untuk mendeteksi *object* orang, penulis menggunakan

metode ini dengan model yang sudah dilatih sehingga program yang disajikan hanya mampu mendeteksi *object* orang (Anggara et al., n.d.).

## 2.5 VGG-16

VGG16 merupakan Arsitektur Jaringan Saraf Konvolusi (CNN) sederhana dan banyak digunakan untuk ImageNet, proyek basis data visual besar dan digunakan dalam penelitian perangkat lunak pengenalan objek visual. “VGG” adalah singkatan dari *Visual Geometry Group*, dan ‘16’ menyiratkan bahwa arsitektur ini memiliki 16 *layer* yang terdiri dari 13 *layer* konvolusi, 2 *layer* *fully connected* dan 1 *layer* *classifier*.



Gambar 2. 3 Arsitektur VGG-16

(Sumber:

<https://www.researchgate.net/publication/350115831/figure/fig1/AS:1002342325428227@1615988441584/Gambar-4-Arsitektur-VGG16-9.png>)

Dari gambar 2.2 Arsitektur VGG-16 di atas dapat diketahui bahwa semua *layer* konvolusi memiliki ukuran kernel 3x3. Jumlah *filter* pada setiap *layer* konvolusi menjadi perbedaan utama dari arsitektur VGG-16. 2 *layer* konvolusi pertama mempunyai jumlah *filter* 64, *layer* 3 dan 4 mempunyai jumlah kernel 128 dan terdapat juga *layer* konvolusi yang mempunyai jumlah *filter* berbeda-beda, yaitu 256 (*layer* 4,5,6) dan 512 (*layer* 7,8,9,10,11,12). 2x2 *max pooling* dilakukan setelah *layer* konvolusi 2, 4, 7, 10 dan 13. *Output* dari *pooling* terakhir akan dihubungkan ke *fully connected layer* yang nantinya akan terhubung ke *classifier*

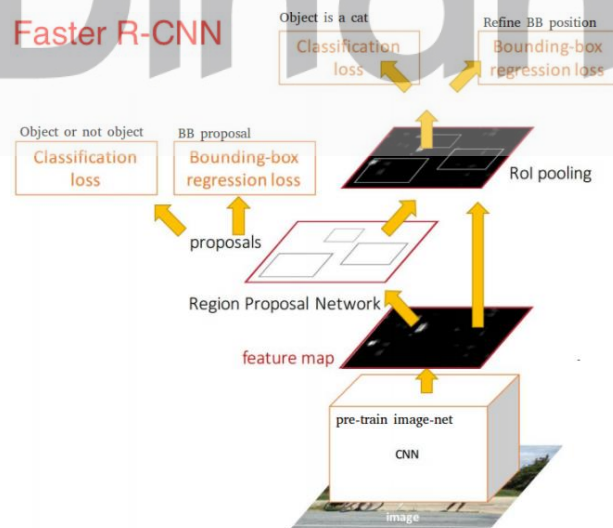


untuk menentukan kelas dari sebuah citra (Rismiyati & Luthfiarta, 2021). Pada penelitian ini, penulis menggunakan Arsitektur VGG-16 untuk proses konvolusi dari sebuah gambar, yang kemudian *output* akan diteruskan untuk proses pendeteksian *object* orang dengan menggunakan metode Faster R-CNN.

## 2.6 Faster R-CNN

Faster R-CNN adalah metode *deep learning* yang berfungsi mengenali objek pada suatu gambar. Objek dapat dikenali dengan menelusuri ciri-ciri objek pada suatu gambar, melalui beberapa *layer* proses konvolusi atau lebih dikenal dengan *Convolutional Neural Network* (CNN) (Alamsyah & Pratama, 2019).

Faster R-CNN merupakan sebuah algoritma deteksi objek dengan 2 modul utama yaitu *Deep Fully Convolutional Network* yang terdapat RPN (*Region Proposed Network*) di dalamnya dan modul yang kedua yaitu detektor dari Faster R-CNN yang berasal dari Fast R-CNN. Metode ini merupakan pengembangan dari Fast R-CNN yang merubah bagian *selective search* menjadi RPN. (Sunario Megawan & Wulan Sri Lestari, 2020).

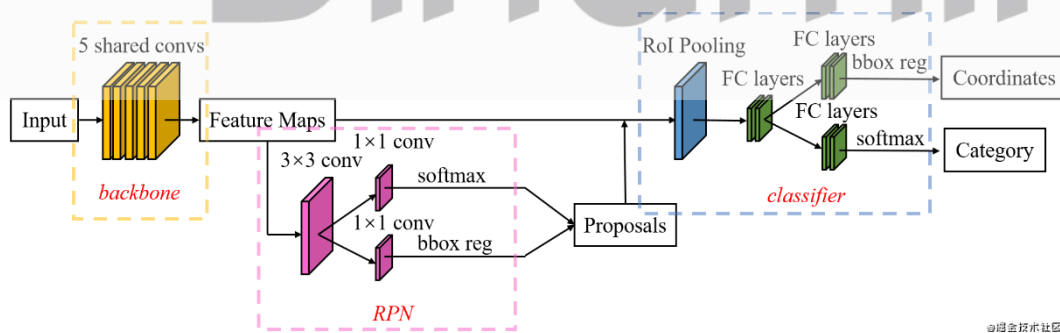


Gambar 2. 4 Bagam Faster R-CNN

(Sumber: <https://medium.com/nodelflux/convolutional-neural-net-untuk-deteksi-objek-f14d72f11ba6>)

Dari Gambar 2.3 Ilustrasi Faster R-CNN diatas, dapat dijelaskan oleh penulis terkait pengertian alur proses yang dilakukan oleh Faster R-CNN :

Saat program dijalankan, program akan mengambil data *input* berupa gambar atau video, data input tersebut akan melalui proses konvolusi dan *Pre-train image* menggunakan arsitektur VGG-16, pada proses ini juga *feature map* dihasilkan. *Feature map* yang dihasilkan ini adalah sebuah *map* yang bertugas untuk mengumpulkan seluruh informasi terkait representasi *vector* dari gambar tersebut (data *input*). Setelah semua informasi terkumpul pada *feature map*, informasi tersebut akan diolah oleh *Region Proposed Network* (RPN) untuk memprediksi area gambar yang dianggap sebagai sebuah *object* orang dan memprediksi *bounding-box* pada area yang dianggap sebagai *object* orang. Setelah itu, informasi awal yang terkumpul pada *feature map* dan informasi *feature map* yang sudah diolah di RPN akan diteruskan ke ROI Pooling (*Region Of Interest*). Pada tahapan RoI Pooling ini terjadi proses untuk mengekstraksi semua informasi yang didapat dari *feature map* dan RPN agar dapat diproses lebih lanjut agar dapat mengklasifikasi *object* yang didapat dan memberikan *bounding-box* pada *object* yang terdeteksi. Pada penelitian ini, proses konvolusi menggunakan model *transfer learning* dengan arsitektur VGG-16, sehingga *object* yang di deteksi hanya *object* orang.



Gambar 2. 5 Struktur Diagram Faster R-CNN

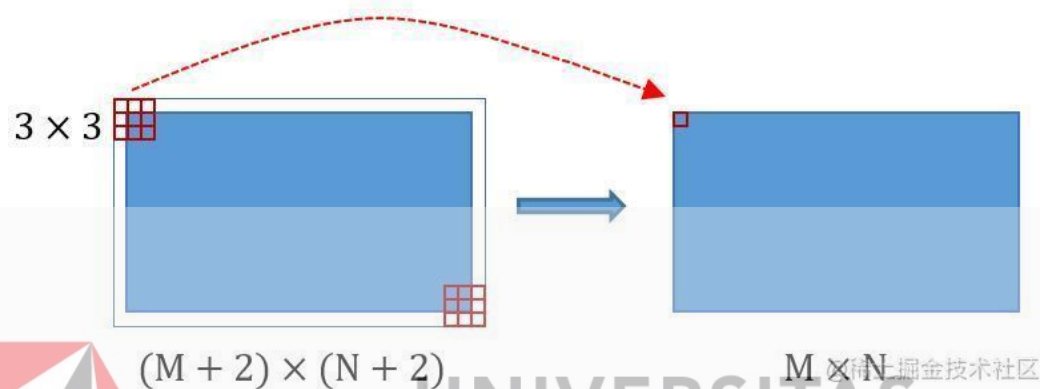
(Sumber : [https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0cee9f27ada24a9db4c806b94c1414c8~tplv-k3u1fbpfcp-](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0cee9f27ada24a9db4c806b94c1414c8~tplv-k3u1fbpfcp-watermark.image)

[watermark.image](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0cee9f27ada24a9db4c806b94c1414c8~tplv-k3u1fbpfcp-watermark.image))

Berdasarkan Gambar 2.4 Struktur Diagram Faster R-CNN di atas, berikut ini penjelasan lebih detail dari setiap proses yang dilakukan oleh Faster R-CNN :

### 2.6.1 Convolutional Layer (Feature Extraction)

Layer ini merupakan bagian dari *convolutional layer*, aktivasi *function* dan *pooling layer*. Semua proses konvolusi yang terdapat di Faster R-CNN akan memperluas gambar asli dengan rumus  $(M+2)*(N+2)$ . Perluasan gambar asli ini terjadi di *layer 3x3* konvolusi, dimana *layer* ini tidak mengubah ukuran matriks antara *input* dan *output*, sehingga menghasilkan sebuah *output*  $M*N$ .



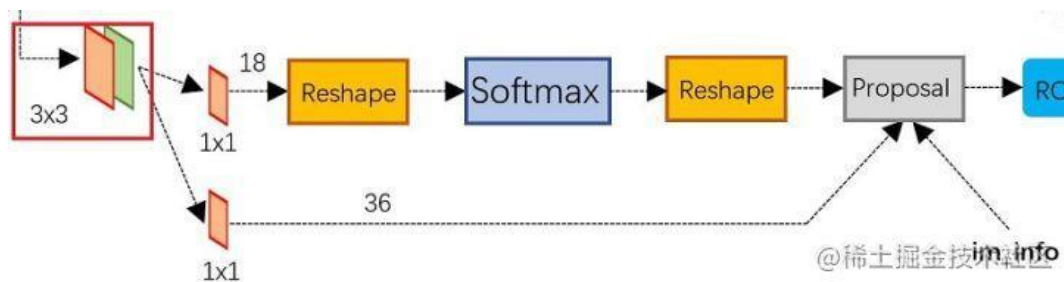
Gambar 2. 6 Prose perubahan ukuran gambar

(Sumber : [https://p6-juejin.byteimg.com/tos-cn-i-](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/4b964dc9becf4249998aa859cbef92f2~tplv-k3u1fbpfcp-watermark.image)

[k3u1fbpfcp/4b964dc9becf4249998aa859cbef92f2~tplv-k3u1fbpfcp-watermark.image](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/4b964dc9becf4249998aa859cbef92f2~tplv-k3u1fbpfcp-watermark.image))

Ukuran kernel dari setiap *pooling layer* di *convolutional layer* adalah  $2 \times 2$  dan *stride* 2. Pengaturan ini mengurangi setengah dimensi data *input* dan mengurangi komputasi hingga 75%, jadi matriks  $M*N$  yang melewati *pooling layer* ukurannya akan berkurang menjadi  $(M/2)*(N/2)$ . Oleh sebab itu, pada *Convolutional Layer (Feature Extraction)*, hanya *pooling layer* saja melakukan perubahan ukuran, dengan panjang dan lebar *output* masing-masing menjadi  $1/2$  dari dimensi *input* yang asli. Sehingga matriks  $M * N$  pada akhirnya dikurangi ukurannya menggunakan rumus  $(M/16)*(N/16)$  dan peta fitur yang dihasilkan oleh *convolutional layer* dapat dikaitkan dengan gambar asli seperti yang ditunjukkan pada Gambar 2.5 di atas.

### 2.6.2 Region Proposal Network (RPN)

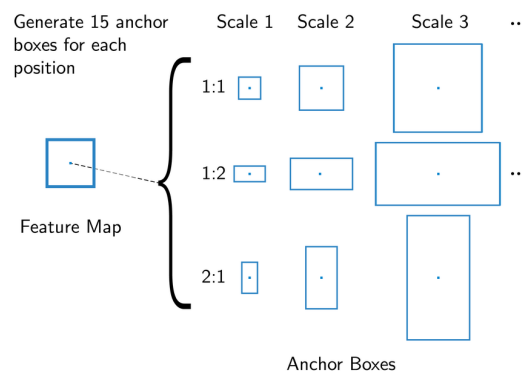


Gambar 2. 7 Struktur RPN

(Sumber : <https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0c4f988d73d446b0b481495396273cec~tplv-k3u1fbpfcp-watermark.image>)

Dari Gambar 2.6 di atas dapat dilihat bahwa jaringan RPN dibagi menjadi 2 bagian. Bagian pertama untuk mengadopsi *softmax* klasifikasi *anchor* untuk mendapatkan klasifikasi positif dan negatif. Bagian terakhir adalah proses perhitungan untuk *anchor* dari regresi *bounding box offset*, untuk mendapatkan titik akurat proposal. dan *layer* proposal terakhir adalah *layer* yang bertanggung jawab untuk integrasi positif *anchor* dan regresi *bounding box offset* yang sesuai dengan akuisisi proposal (Zhang et al., 2016).

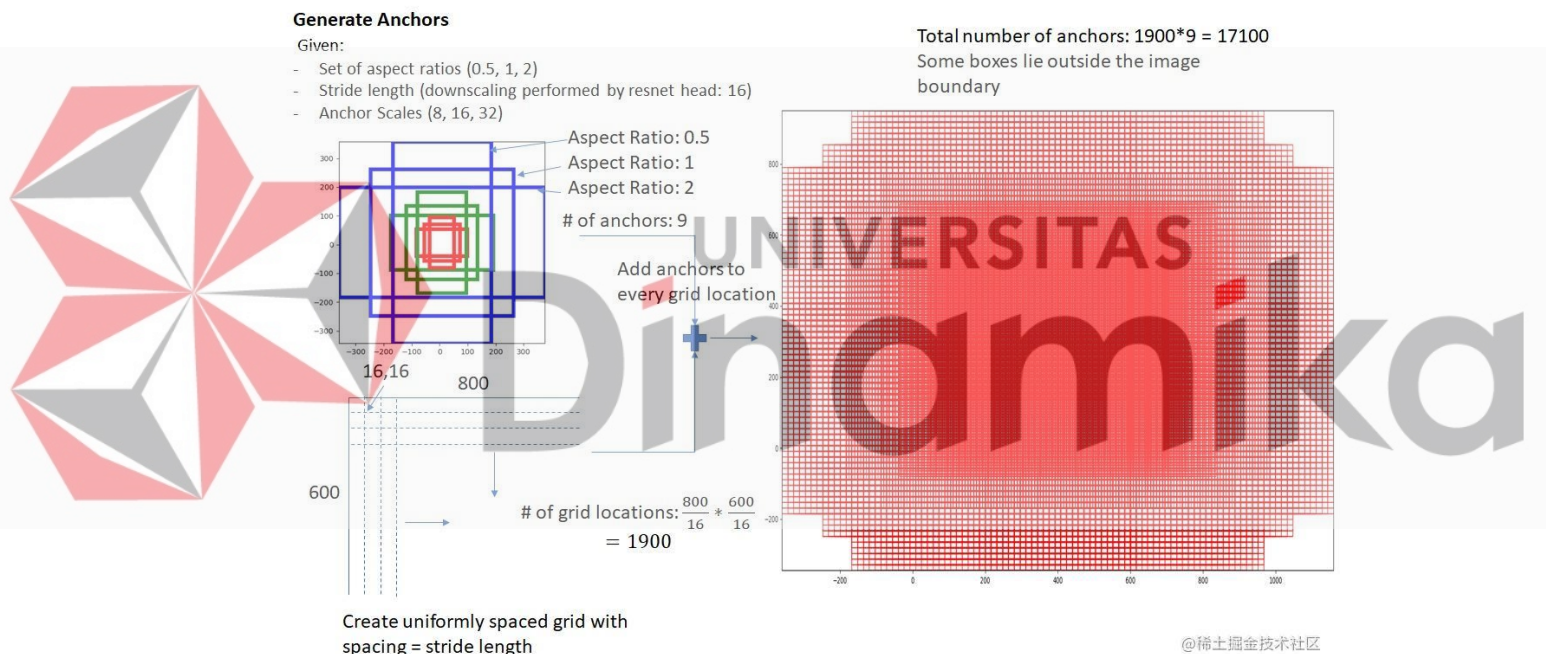
#### A. Anchor



Gambar 2. 8 Struktur *anchor boxes*

(Sumber : <https://www.researchgate.net/profile/Max-Ferguson/publication/327392506/figure/fig8/AS:666613162450944@1535944371721/Anchor-Boxes-at-a-certain-position-in-the-feature-map.png>)

Dari Gambar 2.7 Struktur *anchor boxes* di atas, titik tengah kernel *layer* konvolusi 3x3 disesuaikan dengan posisi pada gambar asli, dan titik tengah ini digunakan sebagai titik pusat dari setiap *anchor*. Berdasarkan hal ini, multiscale *anchor* dengan berbagai rasio aspek diatur pada citra aslinya. Oleh karena itu, *anchor* tidak mengacu pada *feature map*. *Anchor* dihasilkan oleh *convolutional layer* tetapi untuk citra asli.



Gambar 2. 9 Jumlah *anchor* pada gambar asli

(Sumber : <https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/7601f62fce3c47a3bcf91d0fbef2126d~tplv-k3u1fbpfcp-watermark.image>)

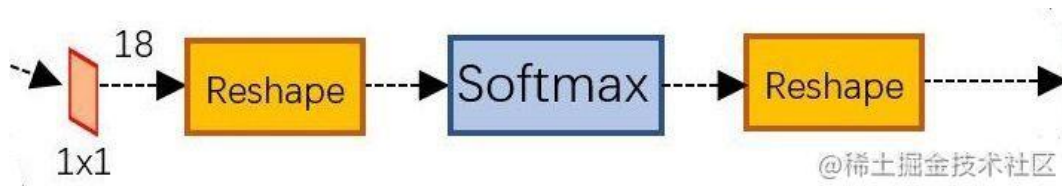
Dari Gambar 2.9 di atas, dapat dilihat bahwa jumlah *anchor* pada gambar asli sebanyak 17100 *anchor*. Jumlah ini didapatkan dari gambar asli 800x600,



VGG *Down Sampling* 16 kali dan 9 titik *feature map* di setiap *anchor*. Semua itu diperhitungkan secara manual dengan cara :

$$\text{Ceil} (800/16) * \text{Ceil} (600/16) * 9 = 50 * 38 * 9 = 17100 \quad (1)$$

### B. *Softmax Determines The Anchors*



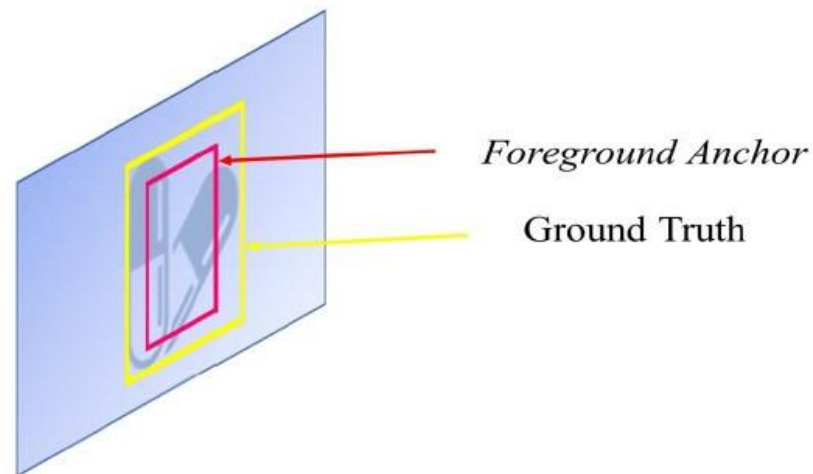
Gambar 2. 10 Alur proses *softmax determine*

(Sumber : [https://p6-juejin.byteimg.com/tos-cn-i-](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/001dbf641ed242689602c9b19df5b35e~tplv-k3u1fbpfcp-watermark.image)

[k3u1fbpfcp/001dbf641ed242689602c9b19df5b35e~tplv-k3u1fbpfcp-watermark.image](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/001dbf641ed242689602c9b19df5b35e~tplv-k3u1fbpfcp-watermark.image))

Setelah menggunakan Backbone untuk mengekstrak *feature map*, layer konvolusi 1x1 digunakan untuk mendapatkan matriks  $H \times W \times 18$  untuk menunjukkan apakah 9 *anchor* pada setiap piksel bernilai positif atau negatif. Kemudian *softmax* digunakan untuk menentukan apakah *object* diikutsertakan, umumnya apabila positif maka *object* diikutsertakan. Sebelum *softmax* digunakan untuk membentuk kembali *input*  $B \times H \times W \times 18$  menjadi  $B \times 2 \times 9 \times H \times W$  dan dimensi kedua digunakan untuk menyimpan hasil klasifikasi *softmax* dari 9 *anchor* di setiap piksel, *reshape layer* digunakan untuk menyesuaikan kembali ukuran gambar ke  $B \times 18 \times H \times W$  (Li, 2021).

### C. *Bounding Box*



Gambar 2. 11 Regresi *Bounding Box*

(Sumber : <https://www.researchgate.net/profile/Liang-Bi-Chen/publication/331980242/figure/fig14/AS:743367646380032@1554244066133/Bounding-box-regression.jpg>)

Berdasarkan Gambar 2.10 Regresi *Bounding Box* dapat dilihat, *Frame* kuning mewakili dasar target *object* yang benar dan *frame* berwarna merah menunjukkan *anchor foreground* yang telah diekstraksi. Meskipun *frame* merah diakui sebagai target *object* oleh *classifier*, jika *frame* merah tidak berada di posisi yang akurat, hasil deteksi dianggap salah dan *object* keluar dari *frame*. Karena itu dibutuhkan sebuah metode untuk menyempurnakan *frame* merah untuk membawa *anchor* lebih dekat dengan *frame* kuning (*Ground Truth*).

### D. *Proposal Layer*

*Proposal Layer* berfungsi untuk menghitung proposal yang akurat berdasarkan matriks regresi koordinat dari *bounding box* ukuran  $H \times W \times 4k$  dan akan dikirim ke *RoI Pooling Layer*. Untuk menerapkan *proposal layer*, terdapat beberapa langkah spesifik yang harus dilakukan :

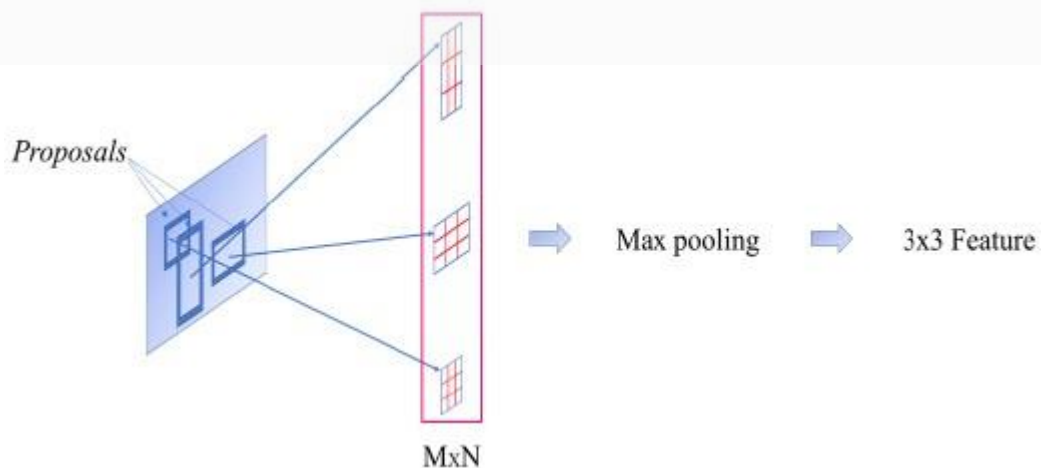
1. Hasilkan *proposal* berdasarkan RPN yang diprediksi *offset* dan positif *bounding box*.



2. Memproses *bounding box* yang melebihi batas gambar, dan memperlakukan batas gambar sebagai batas *bounding box* yang melebihi batas.
3. Menghapus *bounding box* yang kurang dari ambang batas.
4. Mengurutkan semua (*proposal*, *score*) berpasangan dari tinggi ke rendah sesuai dengan nilai *score*.
5. Mengambil nilai pertama *pre\_nms\_topN* (6000)
6. Menampilkan NMS pada *proposal* positif dengan ambang batas = 0.7 .
7. Mengambil nilai terata dari *post\_nms\_topN* (300) setelah NMS.
8. Kembali ke *proposal* yang tersisa.

### 2.6.3 Region Of Interest (RoI) Pooling

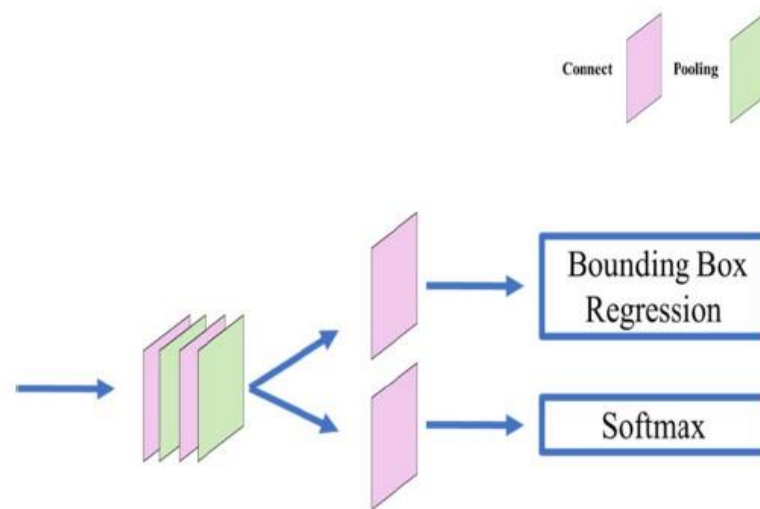
RoI pooling layer memiliki dua fungsi utama yaitu sejajar dengan *proposal object* dalam gambar dengan *patch* yang sesuai di *feature map* dan mengumpulkan sampel yang sesuai dengan *feature map patch* kemudian mentransfernya ke *fully connected layer*. RoI pooling layer secara merata membagi setiap area kandidat ke blok  $M \times N$  dan melakukan *max pooling* pada setiap blok. Sehingga, *proposal* dengan ukuran yang berbeda pada *feature map* agar dikonversi menjadi data ukuran seragam dan dikirim ke layer berikutnya.



Gambar 2. 12 Struktur RoI Pooling Layer

(Sumber : <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8679954>)

### 2.6.4 Classification



Gambar 2. 13 Struktur *Classification Layer*

(Sumber : <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8679954>)

Berdasarkan Gambar 2.13 Struktur *Classification Layer* di atas, dapat dijelaskan lasifikasi dilakukan dengan menggunakan *feature map region* yang diperoleh untuk setiap *proposal* agar dapat menentukan kategori dimana setiap *proposal* yang dimiliki melalui *fully connected layer* dan *softmax layer*. Regresi *bounding box* digunakan untuk memperbaiki setiap *box anchor* ke posisi yang lebih tepat.

### 2.7 Confusion Matrix

*Confusion matrix* bisa juga disebut sebagai *error matrix*, artinya memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji dimana nilai sebenarnya diketahui.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<b>TP</b> (True Positive)	<b>FP</b> (False Positive) <i>Type I Error</i>
	0 (Negative)	<b>FN</b> (False Negative) <i>Type II Error</i>	<b>TN</b> (True Negative)

Gambar 2. 14 Tabel *Confusion Matrix*(Sumber : [https://miro.medium.com/max/1400/1\\*](https://miro.medium.com/max/1400/1*IzN36IDL95ASZcV7g_KRUg.jpeg)[IzN36IDL95ASZcV7g\\_KRUg.jpeg](https://miro.medium.com/max/1400/1*IzN36IDL95ASZcV7g_KRUg.jpeg))

Dari Gambar 2.14 Tabel *Confusion Matrix* di atas terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Berikut ini penjelasan 4 istilah dari tabel *confusion matrix* dengan contoh yang sesuai dengan penelitian Tugas Akhir ini :

- True Positive* (TP), merupakan data positif yang diprediksi benar oleh sistem (model). Contohnya terdapat *object* orang dan sistem (model) mendeteksi bahwa *object* tersebut adalah *object* orang.
- True Negative* (TN), merupakan data negatif yang diprediksi benar oleh sistem (model). Contohnya terdapat *object* barang dan sistem (model) tidak mendeteksi *object* barang tersebut.
- False Positive* (FP), merupakan data negatif akan tetapi sistem (model) mendeteksi sebagai data positif. Contohnya terdapat *object* barang yang terdeteksi oleh sistem (model) sebagai *object* orang.

- d. *False Negative* (FN), merupakan data positif akan tetapi sistem (model) mendeteksi sebagai data negatif. Contohnya terdapat *object* orang yang terdeteksi oleh sistem (model) sebagai *object* barang.

Berikut ini adalah tujuan dan manfaat dari *Confusion Matrix* :

- a. Menunjukkan bagaimana sistem (model) yang dibuat membuat suatu prediksi.
- b. Memberi informasi tentang kesalahan yang dibuat dan jenis kesalahan yang dibuat oleh sistem (model).
- c. Setiap kolom dari *confusion matrix* merepresentasikan *instance* dari kelas prediksi.
- d. Setiap baris dari *confusion matrix* mewakili *instance* dari kelas aktual.

### 2.7.1 Jenis *Confusion Matrix*

- a. Klasifikasi *Binary*

Data *input* dikelompokkan menjadi dua kelas. Jenis klasifikasi ini merupakan klasifikasi yang paling sederhana, dimana hanya terdapat nilai 0 dan 1.

- b. Klasifikasi *Multi-Class*

Data *input* diklasifikasikan menjadi beberapa kelas. Contoh sistem dapat mengklasifikasikan jenis kendaraan seperti sepeda, sepeda motor, mobil, bus, truk, dan sebagainya.

- c. Klasifikasi *Multi-Label*

Data *input* dikelompokkan menjadi beberapa kelas, namun pada klasifikasi *multi-label*, data dapat dimasukkan dalam beberapa kelas sekaligus.

- d. Klasifikasi *Hierarchical*

Data *input* dikelompokkan menjadi beberapa kelas, namun kelas tersebut dapat dikelompokkan kembali menjadi kelas-kelas yang lebih sederhana secara hirarkis.

### 2.7.2 *Performance Metric* dari *Confusion Matrix*

- a. *Accuracy*

*Accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Sehingga *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Nilai *accuracy* dapat diperoleh dengan rumus (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$Accuracy = \frac{\text{Jumlah object diprediksi benar (Orang + Barang)}}{\text{Jumlah object orang keseluruhan}} \times 100\%$$

Rumus mencari nilai *accuracy* (1)

b. *Precision*

*Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. *Precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah di prediksi dengan benar, berapa banyak data yang benar-benar positif. Nilai *precision* dapat diperoleh dengan menggunakan rumus (2).

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{\text{Jumlah object orang diprediksi benar}}{\text{Jumlah object (Orang atau Barang) diprediksi benar}}$$

Rumus mencari nilai *precision* (2)

c. *Recall*

*Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* dapat diperoleh dengan rumus (3).

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{\text{Jumlah object orang diprediksi benar}}{\text{Jumlah object orang}}$$

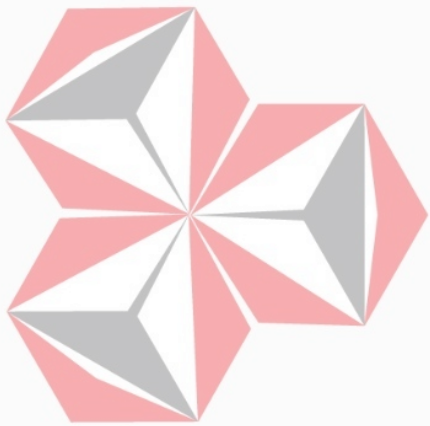
Rumus mencari nilai *recall* (3)

d. F1 - Score

F1 - Score adalah *harmonic mean* (rata-rata harmonik) dari precision dan recall, yang secara matematik nilai F1 - Score didapatkan dengan rumus (4). Nilai F1 - Score berada diantara 0 hingga 1 sehingga, Nilai terbaik F1 - Score adalah 1.0 dan nilai terburuknya adalah 0.

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

Rumus mencari nilai F1 - Score (4)

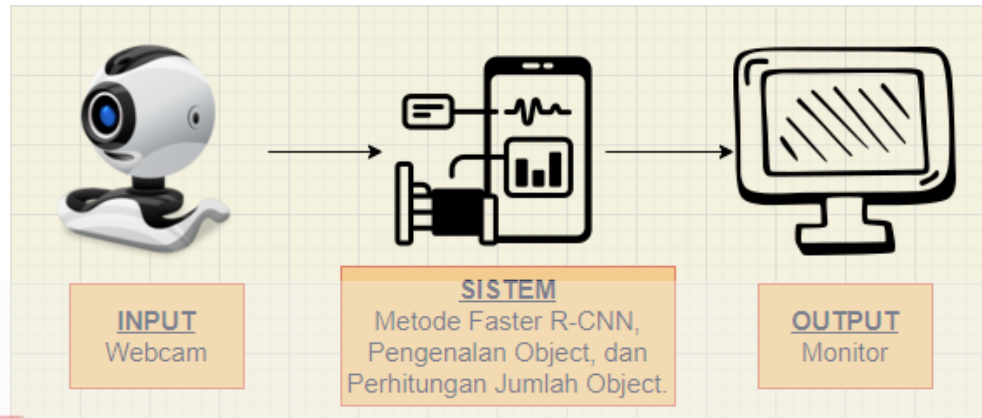


UNIVERSITAS  
Dinamika

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Rancangan *Hardware*



Gambar 3. 1 Model Perancangan *Hardware*

(Sumber : Olahan Penulis)

Berdasarkan Gambar 3.1 Model Perancangan *Hardware* di atas dapat diketahui bahwa *input* data berasal langsung dari kamera. *Input* tersebut akan diproses oleh sistem dengan menggunakan metode Faster R-CNN agar dapat mengenali *object* dan menghasilkan *object* berupa orang. *Output* yang ditampilkan pada monitor adalah hari, tanggal dan waktu, jumlah orang masuk, keluar, dan total jumlah orang di dalam ruangan.



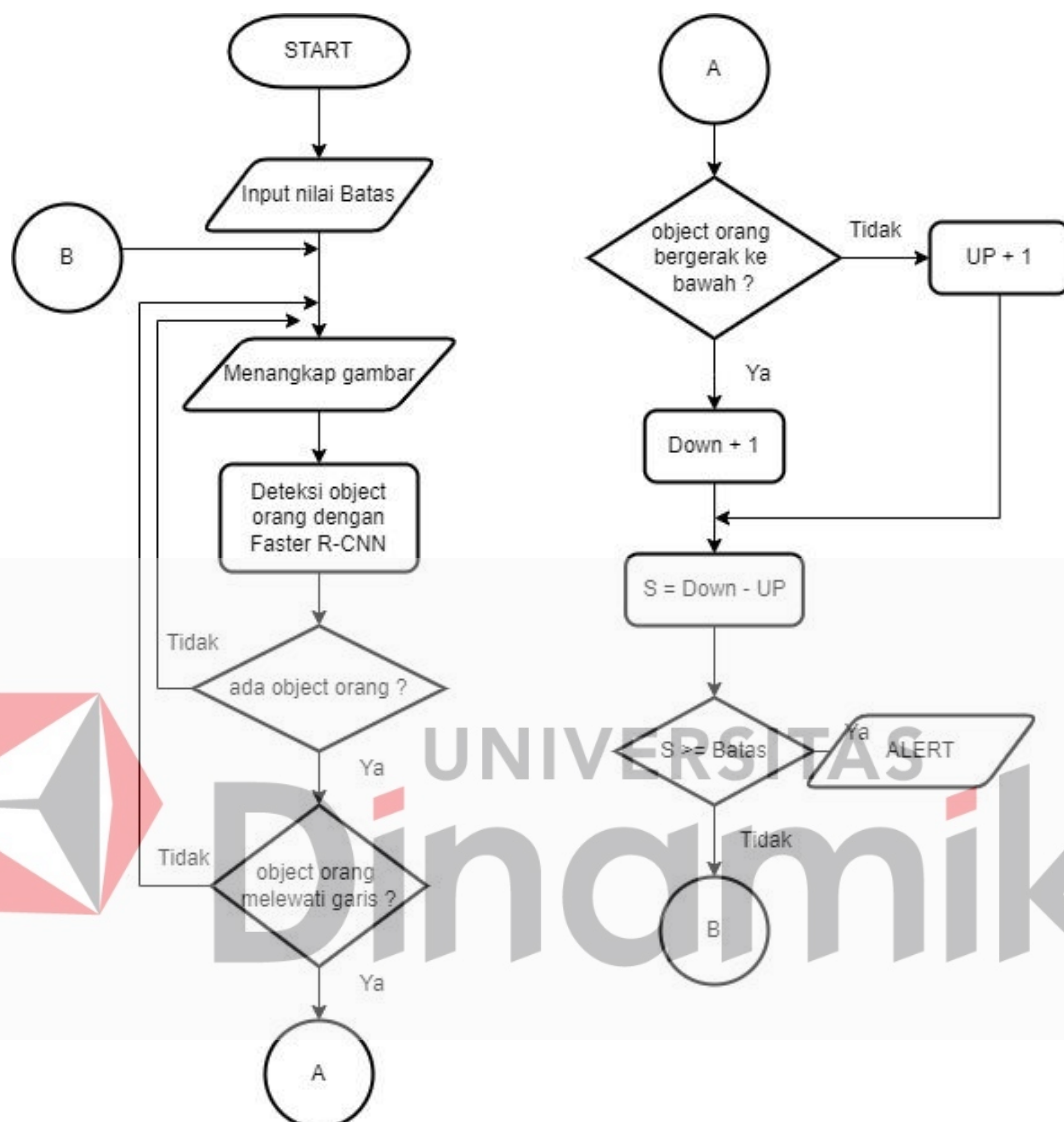


Gambar 3. 2 Tampilan *output* pada layar monitor

(Sumber : Olahan Penulis)

Dilihat dari Gambar 3.2 Tampilan *output* pada layar monitor di atas, dijelaskan bahwa *output* yang ditampilkan sistem pada layar monitor adalah tanggal dan waktu, Jumlah batas orang di dalam ruangan yang sudah ditentukan, jumlah orang yang masuk, keluar, dan jumlah orang yang berada di dalam ruangan. Selain itu terdapat *center line* berwarna merah pada output tersebut, yang berfungsi sebagai batas sekaligus garis perhitungan orang yang berjalan masuk dan orang yang berjalan keluar ruangan.

### 3.2 Flowchart Sistem



Gambar 3. 3 Flowchart sistem deteksi kapasitas orang

(Sumber : Olahan Penulis)

Berdasarkan Gambar 3.3 Flowchart sistem deteksi kapasitas orang di atas dapat dijelaskan bahwa pengguna sistem wajib memasukkan nilai maksimal jumlah orang dalam suatu ruangan yang disimpan di variable **batas**. Kemudian sistem mengambil data input berupa video dari kamera, yang kemudian akan diproses oleh metode Faster R-CNN untuk mendeteksi *object* orang. Apabila sistem tidak mendeteksi *object* orang, maka sistem akan melakukan proses *looping* mulai dari

sistem menangkap gambar pada video yang ditangkap, proses ini akan terus dilakukan hingga sistem mendeteksi adanya *object* orang. Setelah sistem mendeteksi adanya *object* orang, sistem akan mendeteksi apakah *object* orang telah melewati *center line*. Jika *object* orang tidak melewati *center line* maka sistem akan melakukan proses *looping* kembali pada proses menangkap gambar pada video. Akan tetapi jika *object* orang terdeteksi telah melewati *center line*, sistem melakukan proses *tracking* pada *object* orang yang sudah terdeteksi untuk menentukan apakah *object* orang berjalan ke bawah *line* (masuk ke dalam ruangan ) melewati *center line* atau *object* orang berjalan ke atas (keluar dari ruangan). Apabila *object* orang terdeteksi sistem berjalan ke bawah dan telah melewati *center line*, maka secara otomatis nilai dari variabel **Down** akan bertambah 1 dan seterusnya. Jika sistem mendeteksi *object* orang berjalan ke atas dan telah melewati *center line*, maka nilai dari variabel **UP** akan bertambah 1 secara otomatis, begitu juga seterusnya

Setelah itu sistem akan menjalankan proses perhitungan dengan menghitung jumlah orang yang berada di dalam ruangan dengan cara nilai dari variabel **Down** dikurangi nilai dari variabel **Up** kemudian hasilnya akan disimpan di variabel **S** dan menampilkannya pada monitor. Kemudian sistem akan melakukan proses perbandingan apakah nilai dari variabel **S** melebihi nilai variabel **Batas** atau tidak. Jika nilai variabel **S** melebihi nilai variabel **Batas** maka akan muncul **ALERT** pada monitor. Apabila nilai variabel **S** tidak melebihi nilai variabel **Batas** maka sistem akan melakukan *looping* mulai dari proses mendeteksi *object* orang.

## BAB IV

### HASIL DAN PEMBAHASAN

Pada Bab 4 ini membahas hasil pengujian dari sistem deteksi yang meliputi tiga parameter uji, yaitu: pengujian deteksi orang, pengujian deteksi *object* dan pengujian akurasi perhitungan data *report*.

#### 4.1 Pengujian Deteksi Orang

Pengujian ini dilakukan untuk mengetahui apakah metode Faster R-CNN mampu mendeteksi *object* orang.

##### 4.1.1 Tujuan Pengujian Deteksi Orang

Tujuan pengujian ini adalah untuk mengetahui apakah sistem dapat mendeteksi orang yang berada dalam suatu *frame* sebuah video.

##### 4.1.2 Prosedur Pengujian Deteksi Orang

1. Menjalankan program yang sudah dibuat menggunakan metode Faster R-CNN untuk mendeteksi orang menggunakan program Visual Studio Code.
2. Ketika orang berada dalam video, maka orang tersebut akan dideteksi oleh sistem melalui tampilan *bounding box* beserta ID nya.
3. Kemudian menghitung tingkat akurasi data menggunakan rumus 1 dibawah ini dengan jumlah pengujian data sebanyak 30 kali. Dalam pengujian ini, setiap orang melakukan 3 kondisi pergerakan yang berbeda kecepatannya, yaitu: berjalan lambat, berjalan normal dan berjalan cepat.

$$\text{Akurasi Data} = \frac{\text{terdeteksi}}{\text{banyak data pengujian}} * 100\% \quad (5)$$

### 4.1.3 Hasil Pengujian Deteksi Orang

Tabel 4. 1 Hasil Uji Deteksi Orang

Uji ke-	Orang ke-	Kondisi Pergerakan			Terdeteksi	Tidak Terdeteksi
		Lambat	Normal	Cepat		
1	1	√	-	-	√	-
2	1	-	√	-	√	-
3	1	-	-	√	√	-
4	2	√	-	-	√	-
5	2	-	√	-	√	-
6	2	-	-	√	√	-
7	3	√	-	-	-	√
8	3	-	√	-	√	-
9	3	-	-	√	√	-
10	4	√	-	-	√	-
11	4	-	√	-	√	-
12	4	-	-	√	√	-
13	5	√	-	-	√	-
14	5	-	√	-	-	√
15	5	-	-	√	√	-
16	6	√	-	-	√	-
17	6	-	√	-	√	-
18	6	-	-	√	√	-
19	7	√	-	-	-	√
20	7	-	√	-	-	√
21	7	-	-	√	√	-
22	8	√	-	-	√	-
23	8	-	√	-	√	-
24	8	-	-	√	√	-
25	9	√	-	-	√	-
26	9	-	√	-	-	√

Uji ke-	Orang ke-	Kondisi Pergerakan			Terdeteksi	Tidak Terdeteksi
		Lambat	Normal	Cepat		
27	9	-	-	√	√	-
28	10	√	-	-	√	-
29	10	-	√	-	-	√
30	10	-	-	√	-	√
Tingkat Akurasi					77%	23%

Dari hasil pengujian pada Tabel 4.1 Hasil Uji Deteksi Orang dapat diketahui bahwa sistem yang dibuat dengan menggunakan metode Faster R-CNN, mampu mendeteksi orang yang berada dalam frame tersebut, dengan tingkat akurasi data 80% yang diperoleh dari perhitungan :

$$\text{Akurasi Data} = \frac{23}{30} * 100\% \quad (6)$$

$$= 77\%$$

Dapat disimpulkan juga, bahwa sistem tidak dapat mendeteksi orang, apabila orang tersebut berjalan terlalu cepat atau setengah berlari sebesar 23%.

## 4.2 Pengujian Deteksi Object

Pengujian ini dilakukan untuk mengetahui apakah sistem yang dibuat dapat mendeteksi *object*, definisi *object* disini adalah *object* orang dan *object* selain orang.

### 4.2.1 Tujuan Pengujian Deteksi Object

Tujuan pengujian ini dilakukan untuk mengetahui apakah sistem yang dibuat mampu mendeteksi *object* orang saja atau *object* lain, pengujian ini perlu dilakukan untuk memastikan bahwasannya sistem hanya mendeteksi *object* orang saja, sedangkan *object* lain tidak akan terdeteksi.

#### 4.2.2 Prosedur Pengujian Deteksi *Object*

1. *Input* data video yang sudah disiapkan. Video di-download dari *link* [www.youtube.com](http://www.youtube.com) dan [www.istockphoto.com](http://www.istockphoto.com) dengan jumlah 30 video yang di-download.
2. Menjalankan program yang sudah dibuat menggunakan metode Faster R-CNN untuk mendeteksi *object*.
3. Mengamati *layout output* pada sistem untuk mengetahui apakah *object* lain (barang) selain *object* orang dapat terdeteksi atau tidak oleh sistem.
4. Melakukan perhitungan secara manual untuk mendapatkan jumlah *object* orang dan *object* barang yang terdeteksi, *object* orang yang terdeteksi sebagai *object* barang, serta *object* barang yang terdeteksi sebagai *object* orang.
5. Perhitungan dilakukan dengan mengamati jumlah ID *Object* yang terdeteksi pada *layout output*.

#### 4.2.3 Hasil Pengujian Deteksi *Object*

Tabel 4. 2 Hasil Uji Deteksi *Object*

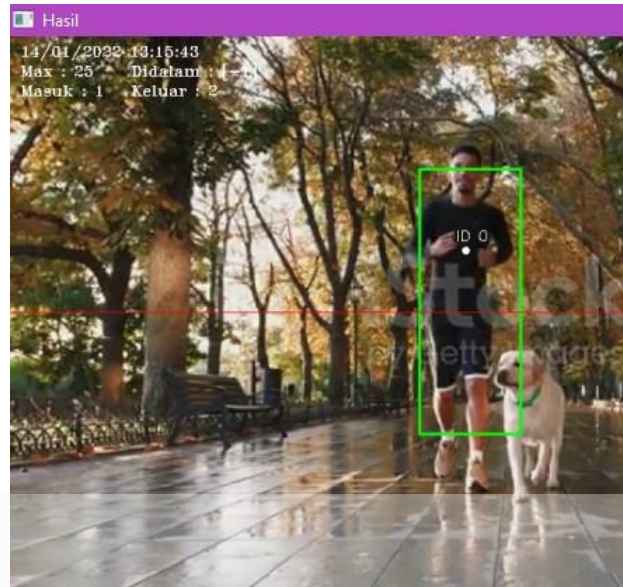
Uji Ke-	Data Uji	Hasil Deteksi <i>Object</i>					
		Jumlah Orang Pada Video	Jumlah Orang Terdeteksi	Jumlah Orang Tidak Terdeteksi	Jumlah Barang Terdeteksi	Jumlah Orang Terdeteksi Barang	Jumlah Barang Terdeteksi Orang
1	Video 1	15	14	1	0	0	0
2	Video 2	28	20	8	0	0	0
3	Video 3	45	39	6	0	0	0
4	Video 4	25	18	7	0	0	0
5	Video 5	5	5	0	0	0	0
6	Video 6	20	19	1	0	0	0
7	Video 7	10	9	1	0	0	0
8	Video 8	45	34	11	0	0	0
9	Video 9	82	70	12	0	0	0



Uji Ke-	Data Uji	Jumlah Orang Pada Video	Hasil Deteksi <i>Object</i>				
			Jumlah Orang Terdeteksi	Jumlah Orang Tidak Terdeteksi	Jumlah Barang Terdeteksi	Jumlah Orang Terdeteksi Barang	Jumlah Barang Terdeteksi Orang
10	Video 10	1	1	0	0	0	0
11	Video 11	36	27	9	0	0	0
12	Video 12	5	4	1	0	0	0
13	Video 13	13	9	4	0	0	0
14	Video 14	6	6	0	0	0	0
15	Video 15	1	1	0	0	0	0
16	Video 16	4	4	0	0	0	0
17	Video 17	44	31	13	0	0	0
18	Video 18	31	23	8	0	0	0
19	Video 19	3	3	0	0	0	0
20	Video 20	4	4	0	0	0	0
21	Video 21	18	14	4	0	0	0
22	Video 22	3	3	0	0	0	0
23	Video 23	11	7	4	0	0	0
24	Video 24	17	13	4	0	0	0
25	Video 25	12	9	3	0	0	0
26	Video 26	26	19	7	0	0	0
27	Video 27	14	10	4	0	0	0
28	Video 28	3	2	1	0	0	0
29	Video 29	1	1	0	0	0	0
30	Video 30	25	20	5	0	0	0
<b>Jumlah</b>		<b>553</b>	<b>439</b>	<b>114</b>	<b>0</b>	<b>0</b>	<b>0</b>

Berdasarkan hasil pengujian pada Tabel 4.2 Hasil Uji Deteksi *Object* diatas, pada baris 1 uji ke-1 dari data uji video 1 menunjukkan bahwa jumlah *object* orang yang terdeteksi ada 14, *object* barang yang terdeteksi ada 0, *object* orang yang terdeteksi sebagai *object* barang ada 0 dan *object* barang yang terdeteksi sebagai

*object* orang ada 0. Sehingga dapat disimpulkan bahwa sistem yang dibuat penulis mampu untuk mendeteksi *object* orang saja.



Gambar 4. 1 Hasil deteksi *object* sistem pertama

(Sumber : Olahan Penulis)



Gambar 4. 2 Hasil deteksi *object* sistem kedua

(Sumber : Olahan Penulis)

Dari Gambar 4.1 Hasil deteksi *object* sistem pertama dan Gambar 4.2 Hasil deteksi *object* sistem kedua, dapat dilihat bahwa sistem hanya mendeteksi *object* orang saja meskipun disekitar *object* orang terdapat *object* yang lain.

### 4.3 Pengujian Akurasi Perhitungan Data Report

Pengujian ini dilakukan untuk mengetahui tingkat kecocokan nilai dari hasil *reporting* data dengan perhitungan sistem.

#### 4.3.1 Tujuan Pengujian Akurasi Perhitungan Data Report

Tujuan pengujian ini dilakukan dengan membandingkan perhitungan yang dilakukan oleh sistem dengan data *report* yang disimpan oleh sistem berbentuk file *spreadsheet* dalam kurung waktu kurang lebih 20 menit dengan pengujian dilakukan sebanyak 30x.

#### 4.3.2 Prosedur Pengujian Akurasi Data Report

1. Menjalankan program yang sudah dibuat menggunakan metode Faster R-CNN dengan file berekstensi *.py*
2. Merekam layar output sistem saat program dijalankan menggunakan aplikasi OSBS Studio
3. Membandingkan hasil perhitungan dari video rekaman dengan hasil report yang sudah tersimpan dalam bentuk file *spreadsheet* (*.csv*)

#### 4.3.3 Hasil Pengujian Akurasi Data Report

Tabel 4. 3 Hasil Uji Akurasi Data Report

Uji Ke-	Hasil perhitungan sistem			Data Report		
	Masuk	Keluar	Di dalam	Masuk	Keluar	Di dalam
1	2	1	1	2	1	1
2	3	1	2	3	1	2
3	8	3	5	8	3	5

Uji Ke-	Hasil perhitungan sistem			Data Report		
	Masuk	Keluar	Di dalam	Masuk	Keluar	Di dalam
4	7	3	4	7	3	4
5	9	3	6	9	3	6
6	14	4	10	14	4	10
7	13	3	10	13	3	10
8	10	3	7	10	3	7
9	11	3	8	11	3	8
10	12	3	9	12	3	9
11	13	4	9	13	4	9
12	7	4	3	7	4	3
13	8	3	5	8	3	5
14	9	4	5	9	4	5
15	5	2	3	5	2	3
16	5	3	2	5	3	2
17	7	3	4	7	3	4
18	2	0	2	2	0	2
19	4	0	4	4	0	4
20	4	1	3	4	1	3
21	6	2	4	6	2	4
22	6	3	3	6	3	3
23	6	1	5	6	1	5
24	10	4	7	10	4	7
25	10	0	10	10	0	10
26	8	5	3	8	5	3
27	7	2	5	7	2	5
28	7	4	3	7	4	3
29	11	5	6	11	5	6
30	13	7	6	13	7	6

Dari hasil pengujian pada Tabel 4.2 Hasil Uji Akurasi Data Report disimpulkan, perhitungan yang tercatat pada aplikasi dan perhitungan pada data

report file *spreadsheet* (.csv) memiliki nilai yang sama tanpa angka selisih, dari sini dapat diketahui bahwa akurasi data report adalah 100%.

#### 4.4 Perhitungan *Confusion Matrix*

Untuk perhitungan *Confusion Matrix* pada Tugas Akhir ini penulis menggunakan jenis perhitungan *confusion matrix* dengan klasifikasi *Binary* dimana hanya ada dua klasifikasi yaitu 0 dan 1.

##### 4.4.1 Tujuan Perhitungan *Confusion Matrix*

Perhitungan *Confusion Matrix* ini dilakukan dengan tujuan untuk mendapatkan nilai *Rate accuracy*, *Rate loss*, *Precision*, *Recall*, *F1- Score*.

##### 4.4.2 Prosedur Perhitungan *Confusion Matrix*

1. Perhitungan *Confusion Matrix* dihitung dengan berdasarkan pada Tabel 4.2 Hasil Pengujian Deteksi *Object*.
2. Perhitungan *Confusion Matrix* dilakukan dengan memanfaatkan *Microsoft Excel*.
3. Membuat tabel *Confusion Matrix*
4. Menghitung nilai *Performance Metric* dari *Confusion Matrix* yaitu, *Rate accuracy*, *Rate loss*, *Precision*, *Recall*, *F1- Score* berdasarkan data pada Tabel 4.2 Hasil Pengujian Deteksi *Object*.

##### 4.4.3 Hasil Perhitungan *Confusion Matrix*

Tabel 4. 4 *Confusion Matrix*

<b><u>CONFUSION MATRIX</u></b>		
<i>Actual Values</i>		
<b>Predict Values</b>	<b>TP</b>	<b>FP</b>
	439	0
	<b>FN</b>	<b>TN</b>
	0	0

Keterangan :

- **TP** = Jumlah *object* orang terdeteksi.
- **TN** = Jumlah *object* barang terdeteksi.
- **FP** = Jumlah *object* barang terdeteksi sebagai *object* orang.
- **FN** = Jumlah *object* orang terdeteksi sebagai *object* barang.

Tabel 4.4 di atas adalah tabel *confusion matrix* yang sudah dibuat oleh penulis dengan berdasarkan data pada Tabel 4.2 Hasil Pengujian Deteksi *Object*. Dapat dilihat dari Tabel 4.4 diatas bahwa TP bernilai 439 sedangkan TN, FP dan FN bernilai 0.

Berikut ini hasil perhitungan *performance metric* dari Tabel 4.4 di atas :

- *Accuracy*

Untuk mendapatkan nilai *accuracy*, penulis menghitung menggunakan rumus

(1) sehingga didapatkan hasil :

$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \\
 &= \frac{439 + 0}{439 + 0 + 0 + 0} \times 100\% \\
 &= 100\%
 \end{aligned}$$

Proses perhitungan nilai *accuracy*

(7)

Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis mampu mendeteksi *object* orang saja dengan nilai keakuratan sebesar 100%.

- *Precision*

Untuk mendapatkan nilai *precision*, penulis menghitung menggunakan rumus

(2) sehingga didapatkan hasil :

$$Precision = \frac{TP}{TP + FP}$$

$$\begin{aligned}
 &= \frac{439}{439 + 0} \\
 &= 1
 \end{aligned}$$

Proses perhitungan nilai *precision* (8)

Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *precision* sebesar 1.

- *Recall*

Untuk mendapatkan nilai *recall*, penulis menghitung menggunakan rumus (3) sehingga didapatkan hasil :

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \\
 &= \frac{439}{439 + 0} \\
 &= 1
 \end{aligned}$$

Proses perhitungan nilai *recall* (9)

Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *recall* sebesar 1.

- *F1 – Score*

Untuk mendapatkan nilai *F1 - Score*, penulis menghitung menggunakan rumus (4) sehingga didapatkan hasil :

$$\begin{aligned}
 F1 - Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\
 &= 2 \times \frac{1 \times 1}{1 + 1} \\
 &= 1
 \end{aligned}$$

Proses perhitungan nilai *F1 - Score* (10)



Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *F1 - Score* sebesar 1.

- *Rate Accuracy*

Untuk mendapatkan nilai *rate accuracy*, penulis menghitung menggunakan rumus (11) sehingga didapatkan hasil :

$$\begin{aligned} \text{Rate Accuracy} &= \frac{\text{Jumlah } \textit{object} \text{ orang terdeteksi}}{\text{Jumlah } \textit{object} \text{ orang keseluruhan pada video}} \times 100\% \\ &= \frac{439}{553} \times 100\% \\ &= 79\% \end{aligned}$$

Proses perhitungan nilai *rate accuracy* (11)

Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *rate accuracy* sebesar 79% dalam mendeteksi *object* orang.

- *Rate Loss*

Untuk mendapatkan nilai *rate loss*, penulis menghitung menggunakan rumus (12) sehingga didapatkan hasil :

$$\begin{aligned} \text{Rate Loss} &= 100\% - \text{Rate Accuracy} \\ &= 100\% - 79\% \\ &= 21\% \end{aligned}$$

Proses perhitungan nilai *rate accuracy* (12)

Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *rate loss* sebesar 21% dalam mendeteksi *object* orang.

## BAB V

### PENUTUPAN

#### 5.1 Kesimpulan

Dari hasil pengujian yang dilakukan oleh penulis untuk penelitian Tugas Akhir ini, didapatkan beberapa kesimpulan sebagai berikut :

1. Hasil pengujian deteksi *object* pada Tugas Akhir ini membuktikan bahwa sistem yang dibuat sudah mampu untuk mendeteksi *object* orang saja.
2. Hasil pengujian sistem yang dibuat pada Tugas Akhir ini mampu mendeteksi tiga macam pergerakan orang, yaitu: berjalan lambat, berjalan normal dan berjalan cepat dengan tingkat akurasi 77%, sedangkan sistem tidak dapat mendeteksi beberapa pergerakan orang pada kondisi berjalan lambat dan berjalan cepat sebesar 23%.
3. Tingkat akurasi perhitungan sistem untuk jumlah orang yang masuk, keluar dan jumlah orang yang berada di dalam ruangan dengan hasil data *reporting* yang disimpan dalam bentuk file *spreadsheet* (.csv) memiliki kesesuaian data 100%.
4. Data *reporting* pada sistem ini dicatat setiap detik dalam waktu kurang lebih 20 menit.
5. Sistem (model) yang dibuat oleh penulis memiliki nilai *Accuracy* sebesar 100%, memiliki nilai *Precision* sebesar 1, nilai *Recall* sebesar 1, nilai *F1 – Score* sebesar 1, nilai *Rate Accuracy* sebesar 79% dan *Rate Loss* sebesar 21%.

#### 5.2 Saran

Bagi para peneliti yang ingin melanjutkan penelitian ini, penulis memiliki beberapa saran yang dapat diimplementasikan dan dikembangkan untuk penelitian selanjutnya.

1. Melakukan pengujian deteksi orang dengan berbagai kondisi seperti kondisi

ketika orang itu berhimpitan, berlari, berjalan cepat, atau orang yang menggunakan kursi roda dan lain sebagainya, bisa dicoba menggunakan metode *real time detector* yang lain, misalnya : YOLO dan SSD.

2. Meningkatkan efisiensi hasil *reporting* data dengan menyimpan secara *real time* sesuai dengan kondisi orang yang keluar dan masuk.
3. Memperbaiki tampilan GUI (*Graphical User Interface*) sistem agar lebih mudah dibaca oleh pengguna.



UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

- Admaja, Y. P. (2021). *Sistem penghitung jumlah pengunjung di restoran menggunakan kamera berbasis*.
- Alamsyah, D., & Pratama, D. (2019). Deteksi Ujung Jari menggunakan Faster-RCNN dengan Arsitektur Inception v2 pada Citra Derau. *JuSiTik : Jurnal Sistem Dan Teknologi Informasi Komunikasi*, 2(1), 1. <https://doi.org/10.32524/jusitik.v2i1.435>
- Anggara, T., Irawan, B., Setianingsih, C., & Telkom, U. (n.d.). *HUMAN DETECTION COUNTING WITH FASTER R-CNN ARCHITECTURE FOR*.
- Elekkomp. (2018). *Pengertian Webcam Beserta Fungsi dan Cara Kerjanya - Elektronika dan Komputer*. Elekkomp. <https://elekkomp.blogspot.com/2018/10/pengertian-webcam-beserta-fungsi-dan.html>
- Kandir, N. (2016). *Mengenal OpenCV dan Python serta Kaitan Keduanya – Oase Ilmu Multimedia dan Keislaman*. Norkandirblog. <https://norkandirblog.wordpress.com/2016/12/23/mengenal-opencv-dan-python/>
- Kemenkes. (2020). *Peta Sebaran COVID-19*. Gugus Tugas Percepatan Penanganan COVID-19. <https://covid19.go.id/peta-sebaran-covid19%0Ahttps://covid19.go.id/peta-sebaran> [sitasi 15 Juni 2020]
- Kemenkes RI. (2020). KMK Nomor Hk.01.07/Menkes/328/2020 Tentang Panduan Pencegahan Dan Pengendalian Corona Virus Disease 2019 (Covid-19) Di Tempat Kerja. In *Menteri Kesehatan Republik Indonesia* (Vol. 2019, pp. 1–39).
- Li, W. (2021). Analysis of Object Detection Performance Based on Faster R-CNN. *Journal of Physics: Conference Series*, 1827(1). <https://doi.org/10.1088/1742-6596/1827/1/012085>
- Plazakamera.com. (n.d.). *Logitech C270 HD Webcam Harga Murah dan Spesifikasi*. Plazakamera.Com. Retrieved September 19, 2021, from

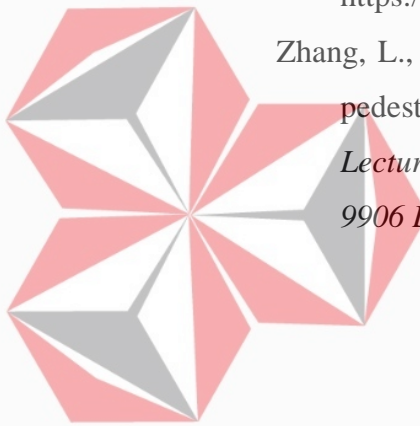
<https://www.plazakamera.com/shop/logitech-c270-hd-webcam/>

Rismiyati, R., & Luthfiarta, A. (2021). VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification. *Telematika*, 18(1), 37. <https://doi.org/10.31315/telematika.v18i1.4025>

Sena, S. (2018). *Pengenalan Deep Learning Part 8 : Gender Classification using Pre-Trained Network (Transfer Learning) | by Samuel Sena | Medium*. Medium.Com. <https://medium.com/@samuelsena/pengenalan-deep-learning-part-8-gender-classification-using-pre-trained-network-transfer-37ac910500d1>

Sunario Megawan, & Wulan Sri Lestari. (2020). Deteksi Spoofing Wajah Menggunakan Faster R-CNN dengan Arsitektur Resnet50 pada Video. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 9(3), 261–267. <https://doi.org/10.22146/.v9i3.231>

Zhang, L., Lin, L., Liang, X., & He, K. (2016). Is faster R-CNN doing well for pedestrian detection? *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9906 LNCS, 443–457. [https://doi.org/10.1007/978-3-319-46475-6\\_28](https://doi.org/10.1007/978-3-319-46475-6_28)



UNIVERSITAS  
Dinamika