



**PENGELOLAAN SISTEM *PHYSICAL DISTANCING* DI DALAM
RUANGAN RAPAT**



UNIVERSITAS
Dinamika

Oleh:

HIZKIA RAMA ANTARES SIMANJUNTAK

18410200029

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2022

**PENGELOLAAN SISTEM *PHYSICAL DISTANCING* DI DALAM
RUANGAN RAPAT**

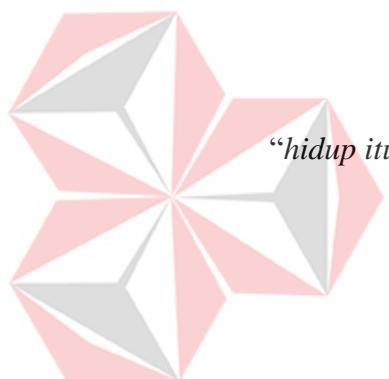
Diajukan sebagai salah satu syarat untuk menyelesaikan mata
kuliah Kerja Praktik



UNIVERSITAS
Dinamika

Disusun Oleh:
Nama : Hizkia Rama Antares Simanjuntak
NIM : 18410200029
Program : S1 (Strata Satu)
Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2022



“hidup itu seperti reaksi kimia jika tidak dikocok maka tidak akan ada perubahan.”

UNIVERSITAS
Dinamika



UNIVERSITAS
Dinamika

“Kupersembahkan kepada ibu, bapak, dan teman-teman ku yang selalu memberi support selama ini sehingga saya mampu menyelesaikan tulisan ini.”

LEMBAR PENGESAHAN

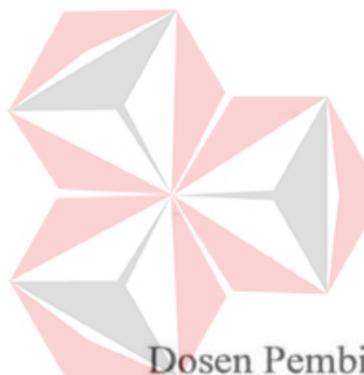
PENGELOLAAN SISTEM *PHYSICAL DISTANCING* DI DALAM RUANGAN RAPAT

Laporan Kerja Praktik oleh

Hizkia Rama Antares Simanjuntak

NIM : 18410200029

Telah diperiksa, diuji, dan disetujui



Dosen Pembimbing,



Digitally signed by
Universitas Dinamika
Date: 2022.01.14
19:01:33 +07'00'

Yosefine Triwidayastuti, M.T.

NIDN: 0729038504

UNIVERSITAS
Dinamika
Disetujui,
Surabaya, 11 Januari 2022

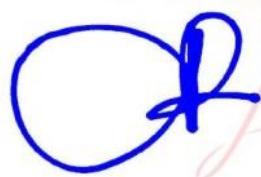
Penyelia,



Wildani Firdaus

NIP: 0392102719

Ketua Program Studi
S1 Teknik Komputer



Universitas
Dinamika
2022.01.17
10:10:03 +07'00'

Pauladie Susanto, S.Kom., M.T.

NIDN: 0729047501

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **Hizkia Rama Antares Simanjuntak**
NIM : **18410200029**
Program Studi : **S1 Teknik Komputer**
Fakultas : **Fakultas Teknologi dan Informatika**
Jenis Karya : **Laporan Kerja Praktek**
Judul Karya : **PENGELOLAAN SISTEM PHYSICAL DISTANCING
DI DALAM RUANGAN RAPAT**

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 kesarjanaan yang telah diberikan kepada Saya.

Surabaya, 12 Januari 2022



Hizkia Rama Antares Simanjuntak
NIM : 18410200029

ABSTRAK

Berbasis di negara kepulauan terbesar dengan sejarah panjang pengaruh maritimnya di dunia, PT Pelindo III mengalami penurunan arus kapal dan barang pada bulan Juni 2020, selain itu PT Pelindo III juga memproyeksikan kinerja Perseroan hingga akhir tahun 2020 akan megalami penurunan 6% hingga 9%. Selain berdampak pada perusahaan, karyawan PT Pelindo III juga terkena dampak dari pandemi Covid-19 yang mengakibatkan karyawan tidak dapat bekerja penuh dan tidak efisien. Oleh sebab itu dibuatlah sistem untuk membantu sistem kerja WFH (*work from home*) untuk mencegah penyebaran COVID-19, sistem tersebut dapat mendekteksi jarak antar orang pada ruang rapat melalui kamera webcam/CCTV sehingga karyawan yang ada di PT Pelindo III tetap memenuhi protocol kesehatan yang diterapkan pemerintah. Selain itu sistem ini juga dapat mengirimkan hasil data yang didapat pada database yang sudah disiapkan.

Pada percobaan yang dilakukan sistem untuk pendektesian jarak manusia yang dilakukan percobaan sebanyak 10 kali mendapatkan persentase keberhasilan 100%. Begitu juga dengan pengiriman data yang dilakukan oleh sistem melalui API ke database pada percobaan pengiriman data dilakukan sebanyak 10 kali dan hasil yang didapatkan sesuai dengan harapan dengan kesimpulan yang didapatkan persentase keberhasilan sebesar 100%.

Kata kunci: *WFH (work from home), physical distancing*

KATA PENGANTAR

Syukur alhamdullilah, penulis ucapan atas rahmad dan ridha Allah Yang Maha Esa, karena atas nikmat dan kuasanya penulis dapat menyelesaikan Laporan Kerja Praktik yang berjudul "Pengelolaan Sistem *Physical Distancing* Di Dalam Ruangan Rapat" tepat pada waktunya, sebagai sebuah syarat untuk menyelesaikan mata kuliah Kerja Praktik

Pada kesempatan ini, penulis mengucapkan terima kasih pada berbagai pihak yang membimbing dan memberi dukungan selama proses mengerjakan Laporan Kerja Praktik ini ini :

1. Orang Tua, yang telah memberikan dukungan besar hingga penulis dapat menyelesaikan Laporan Kerja Praktik ini dengan baik.
2. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
3. Ibu Yosefine Widayastuti, M.T. selaku Dosen Wali dan Dosen Pembimbing yang selalu memberi waktu dan bimbingan selama Kerja Praktik.
4. Bapak Wildani Firdaus selaku Penyelia dan Mentor penulis selalu mendampingi dan membimbing penulis serta memberikan penulis banyak ilmu dalam mengerjakan proyek yang ditugaskan.
5. Seluruh teman-teman S1 Teknik Komputer angkatan 2018 yang memberikan semangat pantang menyerah dan selalu menemani selama proses mengerjakan Laporan Kerja Praktik.

Penulis menyadari, masih banyak kesalahan yang terdapat dalam penulisan Laporan Kerja Praktik, 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

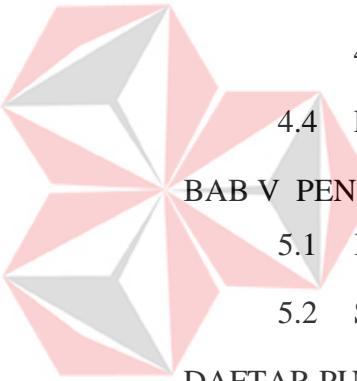
Surabaya, 11 Januari 2022

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 Rumusan Masalah..... | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan | 3 |
| 1.5 Manfaat | 3 |
| BAB II GAMBARAN UMUM PERUSAHAAN..... | 4 |
| 2.1 Sejarah | 4 |
| 2.2 Visi, Misi dan Nilai Perusahaan | 5 |
| 2.2.1 Visi | 5 |
| 2.2.2 Misi..... | 6 |
| 2.2.3 Nilai – Nilai Perusahaan..... | 6 |
| 2.3 Entitas Bisnis | 6 |
| 2.4 Logo..... | 7 |
| 2.5 Lokasi Perusahaan | 7 |
| BAB III LANDASAN TEORI..... | 8 |
| 3.1 Bahasa Python..... | 8 |
| 3.2 Deep Learning..... | 8 |
| 3.3 Raspberry Pi 3..... | 9 |
| 3.4 API (Application Programming Interface) | 10 |

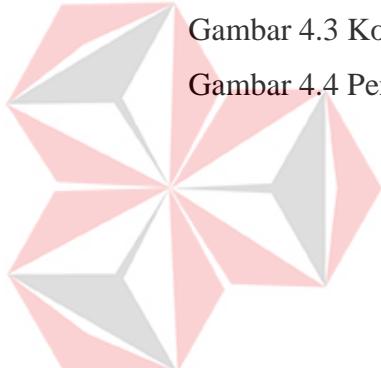




| | | |
|--------------------------------------|--|----|
| 3.5 | CNN (<i>Convolutional Neural Network</i>)..... | 11 |
| 3.6 | Caffe Deep Learning Framework | 12 |
| 3.6 | Open CV | 13 |
| 3.7 | Numpy | 14 |
| 3.8 | SSD Single Shot MultiBox Detector | 15 |
| BAB IV DESKRIPSI HASIL PROJECT | | 16 |
| 4.1 | FlowChart | 16 |
| 4.2 | Menginstal Library pada Rasberry Pi | 17 |
| 4.3 | Membuat program..... | 17 |
| 4.3.1 | Mendekteksi Manusia..... | 18 |
| 4.3.2 | Koneksi CAFFE | 19 |
| 4.3.3 | Kondisi Jarak Pada Box | 21 |
| 4.4 | Pengiriman data ke database..... | 23 |
| BAB V PENUTUP | | 25 |
| 5.1 | Kesimpulan | 25 |
| 5.2 | Saran | 25 |
| DAFTAR PUSTAKA | | 26 |
| BIODATA PENULIS | | 47 |

DAFTAR GAMBAR

| | Halaman |
|---|---------|
| Gambar 2. 1 Logo PT. Pelabuhan Indonesia Rregional III..... | 7 |
| Gambar 2. 2 Lokasi PT. Pelindo Regional III dari Universitas Dinamika | 7 |
| Gambar 3.1 Logo Bahasa Python..... | 8 |
| Gambar 3.2 Raspberry Pi 3 Model B+..... | 9 |
| Gambar 3.3 API | 11 |
| Gambar 3.4 CNN | 11 |
| Gambar 3.5 OpenCV..... | 13 |
| Gambar 3.6 SSD Single Shot MultiBox Detector..... | 15 |
| Gambar 4.1 FlowChart..... | 16 |
| Gambar 4.2 Menginstal Library pada Rasberry Pi..... | 17 |
| Gambar 4.3 Kondisi Jarak Pada Box | 21 |
| Gambar 4.4 Pengiriman data ke database | 23 |



UNIVERSITAS
Dinamika

DAFTAR TABEL

| | |
|---|----|
| Tabel 4.1 Kondisi Jarak Pada Box..... | 21 |
| Tabel 4.2 Pengiriman data ke database | 23 |



DAFTAR LAMPIRAN

| | Halaman |
|---|-----------|
| Lampiran 1 Surat Balasan dari Perusahaan | 27 |
| Lampiran 2 Kartu Bimbingan Kerja Praktik | 29 |
| Lampiran 3 Form KP-5 | 30 |
| Lampiran 4 Form KP-6 | 33 |
| Lampiran 5 Form KP-7 | 37 |
| Lampiran 6 Form KP-3A | 40 |
| Lampiran 7 Form KP-4 | 41 |
| Lampiran 8. 1 Source code program..... | 42 |



BAB I

PENDAHULUAN

1.1 Latar Belakang

Berbasis di negara kepulauan terbesar dengan sejarah panjang pengaruh maritimnya di dunia, Pelabuhan Indonesia (Pelindo) adalah pelabuhan kelas dunia yang menawarkan anda layanan terintegrasi antar pelabuhan di Indonesia. Pelindo merupakan perusahaan hasil integrasi dari empat (4) BUMN pelabuhan yaitu PT Pelindo I (Persero), PT Pelindo II (Persero), PT Pelindo III (Persero) dan PT Pelindo IV (Persero) yang resmi berdiri pada tanggal 1 Oktober 2021. Berdirinya Pelindo sebagai perusahaan hasil integrasi ini adalah inisiatif strategis pemerintah selaku pemegang saham untuk mewujudkan konektivitas nasional dan jaringan ekosistem logistik yang lebih kuat (Indonesia, 2021c).

Ketika pandemi virus Covid-19 menyebar di Indonesia, PT Pelindo III merupakan salah satu sector yang terkena dampaknya. Akibatnya, PT Pelindo III mengalami penurunan arus kapal dan barang pada bulan Juni 2020, selain itu PT Pelindo III juga memproyeksikan kinerja Perseroan hingga akhir tahun 2020 akan megalami penurunan 6% hingga 9%. Selain berdampak pada perusahaan, karyawan PT Pelindo III juga terkena dampak dari pandemi Covid-19 yang mengakibatkan karyawan tidak dapat bekerja penuh dan tidak efisien. Akan tetapi PT Pelindo III selalu berupaya untuk tetap beroperasi dan menyesuaikan diri pola hidup yang baru selama pandemi agar dapat menggerakkan Kembali ekonomi nasional. Beberapa upaya yang dilakukan oleh PT Pelindo III adalah menerapkan kebijakan bekerja penuh pada karyawannya setelah Lebaran Tahun 2020, tentu dengan menerapkan skema mulai dari penerapan protokol kesehatan, WFH (*work from home*), hingga memberikan pelayanan digital untuk kegiatan operasionalnya.

Salah satu protokol kesehatan yang harus diterapkan oleh karyawan PT Pelindo III adalah *Physical Distancing*, artinya seluruh karyawan PT Pelindo III harus menjaga jarak sejauh 1 meter seuai yang dianjurkan oleh WHO (*World Health Organization*). Meskipun sudah PT Pelindo III sudah menerapkan WFH, bukan tidak mungkin terjadi perkumpulan masa meskipun jumlahnya sedikit,

mengingat salah satu cara virus Covid-19 menyebar adalah melakukan kegiatan social atau berkumpul tanpa memperdulikan jarak yang harus dijaga. Oleh sebab itu *Physical Distancing* berperan penting untuk menekan dan meminimalisir penyebaran virus Covid-19.

Berdasarkan permasalahan tersebut dibuat sistem deteksi physical distancing di ruangan rapat PT Pelindo III, karena ruang rapat merupakan ruangan yang memungkinkan terjadinya perkumpulan massa. Sistem deteksi *physical distancing* ini dibuat menggunakan bahasa python dengan menggunakan raspberry pi sebagai minikomputer yang memanfaatkan kamera/CCTV ruangan sebagai inputan atau pemantau. Sistem ini diuat dengan memberikan batasan jumlah karyawan yang boleh berada di dalam ruangan tersebut, serta memantau jarak antar karyawan yang berada di dalam ruangan tersebut. Dengan adanya sistem ini, penyebaran virus Covid-19 dapat diminimalisir dan dicegah secepat mungkin yang dapat mengurangi dampak negatif, apabila terdapat salah satu karyawan yang terinfeksi virus Covid-19, serta membantu pendataan sistem kerja WFH (*Work From Home*) pada perkantoran.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah pada Kerja Praktek ini sebagai berikut :

1. Bagaimana melakukan pendekteksian manusia untuk sistem social distancing menggunakan *deep learning* ?
2. Bagaimana melakukan pendekteksian jarak antar manusia untuk sistem social distancing menggunakan *deep learning* ?
3. Bagaimana melakukan pengiriman data pada pendekteksian kondisi jarak antar manusia ke database ?

1.3 Batasan Masalah

Dalam pembuatan Kerja Praktek ini, pembahasan masalah dibatasi pada beberapa hal berikut :

1. Sistem tidak dapat mendekteksi jarak antar manusia dalam satuan centimenter.

2. Sistem yang digunakan adalah *pre-trained program*.
3. Sistem yang digunakan cukup berat untuk device raspberry pi

1.4 Tujuan

Tujuan dari penggerjaan *project* ini adalah :

1. Mampu melakukan pendekteksian jarak antar manusia
2. Mempu memberikan kondisi pada jarak antar manusia
3. Mamppu mengirimkan data kondisi jarak antar manusia ke data base

1.5 Manfaat

Manfaat yang diperoleh dari penggerjaan *project* ini adalah :

1. Mengurangi penyebaran COVID-19
2. Membantu pendataan pada program WFH pada perkantoran



BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Sejarah

Indonesia memiliki sejarah panjang sebagai negara maritim. Di masa lalu, kerajaan-kerajaan maritim nusantara seperti Sriwijaya, Majapahit, kerajaan di Maluku pernah memegang kunci jalur perdagangan dunia lewat rempah-rempah. Pedagang-pedagang dari Gujarat dan China mengambil rempah-rempah dari Kepulauan Maluku lalu mengirimkannya melalui kapal-kapal dagang menuju Cina, Semenanjung Arab, Eropa, hingga ke Madagaskar. Pelabuhan-pelabuhan kecil di Indonesia menjadi tempat persinggahan dan pusat perdagangan yang mempertemukan para pedagang dari berbagai bangsa, sehingga menjadi bandar niaga yang besar. Hal ini melatar lahirnya Pelabuhan Indonesia di era kemerdekaan.

Sebelumnya, untuk mengelola kepelabuhanan di Indonesia, dibentuk 4 pelindo yang terbagi berdasar wilayah yang berbeda. Pelindo I misalnya mengelola pelabuhan di Provinsi Nangroe Aceh Darussalam, Sumatera Utara, Riau dan Kepulauan Riau. Pelindo I dibentuk berdasar PP No.56 Tahun 1991, sedang nama Pelindo I ditetapkan berdasar Akta Notaris No.1 tanggal 1 Desember 1992.

Pelindo II mengelola pelabuhan di wilayah 10 provinsi, yaitu Sumatera Barat, Jambi, Sumatera Selatan, Bengkulu, Lampung, Bangka Belitung, Banten, DKI Jakarta, Jawa Barat, dan Kalimantan Barat. Pelindo II dibentuk berdasar PP No.57 Tahun 1991, Pelindo II Persero) didirikan berdasar Akta Notaris Imas Fatimah SH, No.3, tanggal 1 Desember 1992.

Pelindo III mengelola pelabuhan di wilayah 7 provinsi, yaitu Jawa Timur, Jawa Tengah, Kalimantan Selatan, Kalimantan Tengah, Bali, NTB dan NTT. Pembentukan Pelindo III tertuang dalam Akta Notaris Imas Fatimah, SH No.5 tanggal 1 Desember 1992, berdasar PP No.58 Tahun 1991.

Sedang Pelindo IV mengelola pelabuhan di wilayah 11 provinsi, yaitu Provinsi Kalimantan Timur, Kalimantan Utara, Sulawesi Selatan, Sulawesi

Tengah, Sulawesi Tenggara, Gorontalo, Sulawesi Utara, Maluku, Maluku Utara, Papua, dan Papua Barat. Pelindo IV dibentuk berdasar PP No.59 Tanggal 19 Oktober 1991. Sedang akta pembentukannya adalah Akta Notaris Imas Fatimah, SH no,7 tanggal 1 Desember 1992.

Masing-masing Pelindo memiliki cabang dan anak usaha untuk mengelola bisnisnya. Pelindo I, II, III, IV adalah Perusahaan BUMN *Non Listed* yang sahamnya 100% dimiliki oleh Kementerian BUMN selaku Pemegang Saham Negara Republik Indonesia. Oleh karena itu, tidak terdapat informasi Pemegang Saham Utama maupun Saham Pengendali Individu di Pelindo. Negara Republik Indonesia yang diwakili oleh Kementerian Badan Usaha Milik Negara Republik Indonesia merupakan satu-satunya pemilik dan Pemegang saham tunggal.

Merger atau integrasi keempat Pelindo menjadi satu Pelindo yang kemudian diberi bernama PT Pelabuhan Indonesia ini berdasar Peraturan Pemerintah Nomor 101 Tahun 2021 Tentang Penggabungan PT Pelindo I, III, dan IV (Persero) ke Dalam PT Pelabuhan Indonesia II (Persero). Pelindo II bertindak sebagai *holding* induk (perusahaan induk) dan ke-3 Pelindo (I,III, IV) bertindak sebagai *sub-holding*. Pembentukan *sub-holding* yang mengelola klaster-klaster usaha ditujukan untuk meningkatkan kapasitas pelayanan Pelindo dan efisiensi usaha. Berdasarkan Surat Menteri Badan Usaha Milik Negara Republik Indonesia nomor : S-756/MBU/10/2021 tanggal 1 Oktober 2021 perihal Persetujuan Perubahan nama, Perubahan Anggaran dasar dan Logo Perusahaan. Sehingga Pelindo II berganti nama menjadi PT Pelabuhan Indonesia (Persero) atau Pelindo (Indonesia, 2021b)

2.2 Visi, Misi dan Nilai Perusahaan

2.2.1 Visi

Visi dari PT. Pelabuhan Indonesia Regional III adalah “Menjadi Pemimpin Ekosistem Maritim Terintegrasi dan Berkelas Dunia”. Visi ini merupakan sebuah pernyataan dari cita-cita perusahaan menjadi pintu gerbang utama jaringan logistic global di Indonesia. Cita-cita perusahaan ini, muncul berdasarkan landasan dengan potensi geografis, peluang bisnis serta kebijakan nasional yang membuka peluang bagi perusahaan untuk merealisasikan visi dimaksud.

2.2.2 Misi

Dalam mewujudkan Visi yang sudah ditetapkan diatas, PT. Pelabuhan Indonesia Regional III mengembangkan misi sebagai berikut :

1. Mewujudkan jaringan ekosistem maritim nasional melalui peningkatan koneksi dan integrasi pelayanan guna mendukung pertumbuhan ekonomi negara
2. Menyediakan jasa kepelabuhan dan maritim yang handal dan terintegrasi dengan Kawasan industry untuk mendukung jaringan logistik Indonesia dan global dengan memaksimalkan manfaat ekonomi Selat Malaka.

2.2.3 Nilai – Nilai Perusahaan

PT. Pelabuhan Indonesia Regional III, menjunjung tinggi nilai-nilai perusahaan yang sudah ditetapkan untuk mewujudkan Visi dan Misi yang diatas.

Berikut ini adalah nilai-nilai perusahaan PT. Pelabuhan Indonesia Regional 3 :

1. Amanah, artinya memegang teguh kepercayaan yang diberikan.
2. Kompeten, artinya terus belajar dan mengembangkan kapabilitas.
3. Harmonis, artinya saling peduli dan menghargai perbedaan.
4. Loyal, artinya berdedikasi dan mengutamakan kepentingan Bangsa dan Negara.
5. Adaptif, artinya terus berinovasi dan antusias dalam menggerakkan ataupun menghadapi perubahan.
6. Kolaboratif, artinya membangun kerjasama yang sinergis

2.3 Entitas Bisnis

Ada empat klaster bisnis utama Pelindo, yaitu :

1. Klaster Petikemas
2. Klaster Non Petikemas
3. Klaster Logistik dan Pengembangan Daerah Pesisir (*Hinterland*),
4. Klaster Kelautan, Peralatan, dan Pelayanan Pelabuhan

Pengelompokan klaster bisnis ini dilakukan agar membuat pengembangan bisnis di Pelindo lebih terfokus, meningkatkan kemampuan dan keahlian SDM yang bekerja di setiap klaster, sehingga mampu bekerja dengan lebih efisien dan

membuat kepuasan pelanggan meningkat. Jika ini terjadi maka akan membuka pangsa pasar yang lebih luas dan menaikkan keuntungan perusahaan (Indonesia, 2021a) .

2.4 Logo

Dibawah ini adalah gamabr logo dari perusahaan PT. Pelabuhan Indonesia Regional III :

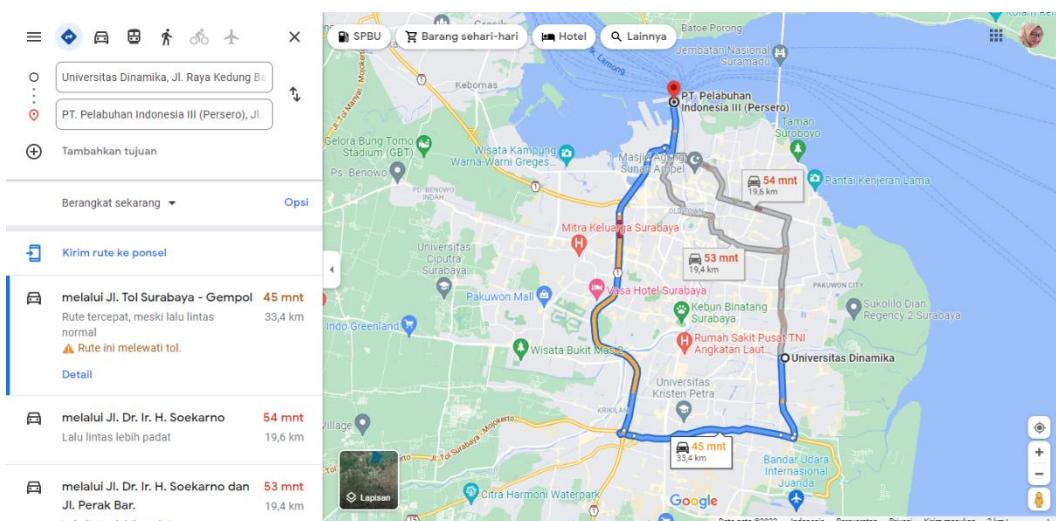


Gambar 2.1 Logo PT. Pelabuhan Indonesia Rregional III

(Sumber : <https://iconape.com/wp-content/files/dx/182789/svg/182789.svg>)

2.5 Lokasi Perusahaan

Lokasi PT. Pelabuhan Indonesia Regional III, terletak di Jl. Perak Timur No.610, Perak Utara, Kecamatan. Pabean Cantian, Kota Surabaya, Jawa Timur 60165. Peta lokasi PT. Pelabuhan Indonesia Regional III dari Universitas Dinamika dapat dilihat pada gambar dibawah ini.



Gambar 2.2 Lokasi PT. Pelindo Regional III dari Universitas Dinamika

(Sumber : <https://maps.google.com/>)

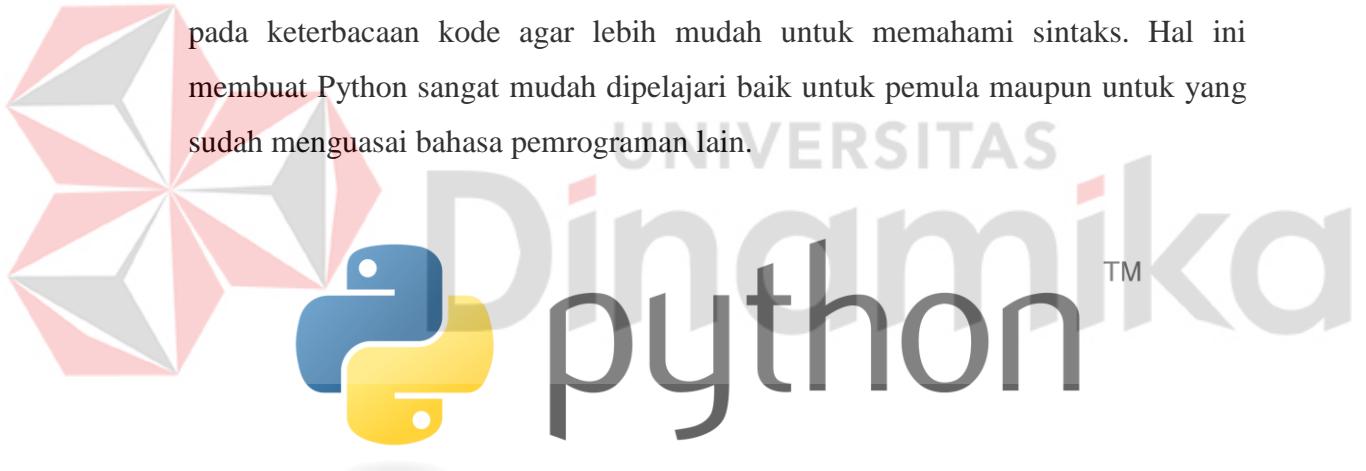
BAB III

LANDASAN TEORI

3.1 Bahasa Python

Python merupakan bahasa pemrograman tingkat tinggi yang diracik oleh Guido van Rossum. Python banyak digunakan untuk membuat berbagai macam program, seperti: program CLI, Program GUI (desktop), Aplikasi Mobile, Web, IoT, Game, Program untuk Hacking, dan lain-lainnya. Python juga dikenal dengan bahasa pemrograman yang mudah dipelajari, karena struktur sintaknya rapi dan mudah dipahami. (Muhardian, 2022)

Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain.



Gambar 3.1 Logo Bahasa Python

3.2 Deep Learning

Deep learning merupakan subbidang machine learning yang algoritmanya terinspirasi dari struktur otak manusia. Struktur tersebut dinamakan *Artificial Neural Networks* atau disingkat ANN. Pada dasarnya, ia merupakan jaringan saraf yang memiliki tiga atau lebih lapisan ANN. Ia mampu belajar dan beradaptasi terhadap sejumlah besar data serta menyelesaikan berbagai permasalahan yang sulit diselesaikan dengan algoritma *machine learning* lainnya.

Saat ini, teknik deep learning sangat populer di kalangan praktisi data dan menarik perhatian banyak pihak. Hal ini karena teknologi deep learning telah diterapkan dalam berbagai produk berteknologi tinggi seperti self-driving car. Selain itu, ia juga ada di balik produk dan layanan yang kita gunakan sehari-hari. Contohnya antara lain, asisten digital, Google Translate, dan voice-activated device (Setiawan, 2021)

3.3 Raspberry Pi 3

Raspberry Pi, sering disingkat dengan nama Raspi, adalah komputer papan tunggal (single-board circuit; SBC) yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris.

Raspberry Pi 3 Model B+ adalah mini komputer versi terbaru yang di keluarkan Raspberry Pi sebagai pengembangan dari versi Raspberry Pi sebelumnya dan menjadi penerus Raspberry Pi 3 Model B yang sudah release. Secara design fisik Raspberry Pi 3 Model B dengan Raspberry Pi 3 model B+ sama sehingga untuk case versi Raspberry Pi 3 Model B bisa dipakai. (Lab Elektronika, 2018)



Gambar 3.2 Raspberry Pi 3 Model B+

Berikut Spesifikasi Raspberry Pi 3 Model B+ :

1. Processor: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
2. Memory: 1GB LPDDR2 SDRAM
3. Power Supply: 5V/2.5A DC power input (microUSB)
4. Wireless: 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
5. Ethernet: Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
6. GPIOs: Extended 40-pin GPIO header
7. Video Output: Full-size HDMI
8. Audio Output: 4-pole stereo output and composite video port
9. USB Port: 4 USB 2.0 ports
10. CSI camera port for connecting a Raspberry Pi camera
11. DSI display port for connecting a Raspberry Pi touchscreen display
12. Micro SD port for loading your operating system and storing data
13. Power-over-Ethernet (PoE) support (requires separate PoE HAT)
14. Operating Temperature: 0~50°C
15. Dimension: 120mm x 75mm x 34mm
16. Weight: 75g

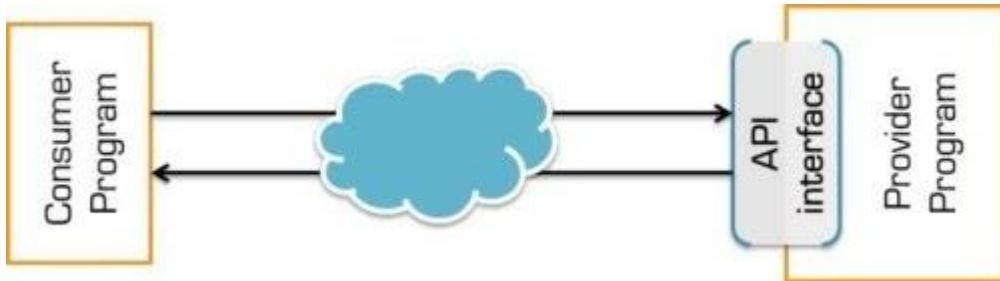
3.4 API (Application Programming Interface)

API adalah singkatan dari Application Programming Interface. API sendiri merupakan interface yang dapat menghubungkan satu aplikasi dengan aplikasi lainnya. Dengan kata lain, peran API adalah sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau pun lintas platform.

Lebih jauh, API sendiri bisa digunakan untuk komunikasi dengan berbagai bahasa pemrograman yang berbeda. Hal ini tentu cukup memudahkan bagi developer. Bahkan, developer tidak perlu menyediakan semua data sendiri karena cukup mengambil data yang dibutuhkan dari platform lain melalui API.

Tidak hanya itu, API juga memungkinkan Anda mengembangkan sebuah website dengan berbagai fitur yang lebih lengkap. Jika menggunakan WordPress,

Anda bisa melakukan integrasi dengan berbagai platform menggunakan API. Salah satu contohnya adalah WordPress REST API. (Lawrence, 2020)

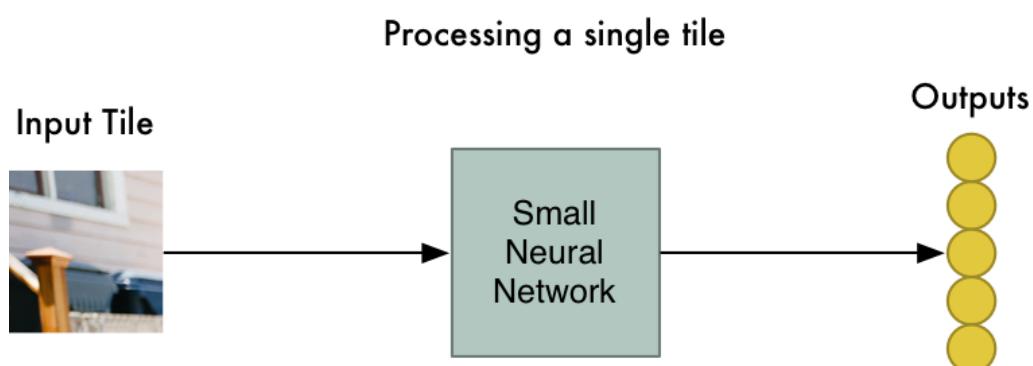


Gambar 3.3 API

3.5 CNN (*Convolutional Neural Network*)

Convolutional Neural Network (CNN) adalah salah satu jenis neural network yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image. CNN adalah sebuah teknik yang terinspirasi dari cara mamalia — manusia, menghasilkan persepsi visual seperti contoh diatas.

Secara garis besar Convolutional Neural Network (CNN) tidak jauh beda dengan neural network biasanya. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Convolutional layer juga terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels). (Lina, 2019)



Gambar 3.4 CNN

3.6 Caffe Deep Learning Framework

Caffe adalah framework deep learning open-source yang dikembangkan untuk Machine Learning. Itu ditulis dalam C++ dan antarmuka Caffe dikodekan dengan Python. Ini telah dikembangkan oleh Berkeley AI Research, dengan kontribusi dari pengembang komunitas. Perangkat lunak ini telah dirancang dengan mengingat ekspresi, kecepatan, modularitas, keterbukaan, dan dukungan komunitas penuh untuk memungkinkan pembuatan model Deep Learning yang mulus. (Recode, 2020)

Fitur Caffe Framework :

1. Kode yang Dapat Diperluas: Kerangka kerja Caffe menampilkan kode, algoritme, dan model terbaik. Seiring waktu, Caffe telah menyaksikan banyak perubahan yang disumbangkan oleh pengembang dan peneliti yang menjadikannya platform yang kuat untuk proyek pembelajaran mendalam.
2. Arsitektur Ekspresif: Arsitektur Caffe yang ekspresif dan modular memungkinkan peningkatan pengembangan aplikasi dan program berdasarkan pembelajaran mesin. Seseorang dapat menentukan model dan mengoptimalkannya tanpa upaya hard-coding. Pengguna dapat beralih antara GPU (Graphics Processing Unit) dan CPU dengan menggunakan satu flag dan melatih model ML pada mesin GPU. Model tersebut kemudian dapat diterapkan ke platform perangkat seluler atau klaster komoditas.
3. Kecepatan: Fitur lain yang menjadikan Caffe sebagai pilihan populer untuk operasi Deep Learning. Dengan satu GPU Nvidia K40, Caffe dapat memproses lebih dari 60 juta gambar per hari. Kecepatan itu diterjemahkan menjadi 1 milidetik/gambar untuk inferensi dan 4 milidetik/gambar untuk operasi pembelajaran. Versi perpustakaan terbaru dan perangkat keras terbaru masih meningkatkan kinerja kecepatan Caffe. Hal ini membuat kerangka kerja Caffe menjadi kandidat yang sempurna untuk menerapkan model ML pembelajaran mendalam di tingkat industri serta untuk eksperimen penelitian.

3.6 Open CV

OpenCV (Open Source Computer Vision Library), adalah sebuah library open source yang dikembangkan oleh intel yang fokus untuk menyederhanakan programing terkait citra digital. Di dalam OpenCV sudah mempunyai banyak fitur, antara lain : pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, dan berbagai jenis metode AI (Artificial Intellegence). Dan menyediakan berbagai algoritma sederhana terkait Computer Vision untuk low level API.

OpenCV merupakan open source computer vision library untuk bahasa pemrograman C/C++, dan telah dikembangkan ke phyton, java, matlab. Library dapat diunduh melalui link <http://sourceforge.net/projects/opencvllibrary> atau melalui <http://opencv.org/releases.html>. Saat ini versi yang terbaru adalah 3.3.0 dengan tanggal rilis 3 Agustus 2017. (Hanugra Aulia Sidharta, 2017)



Gambar 3 5 OpenCV

3.7 Numpy

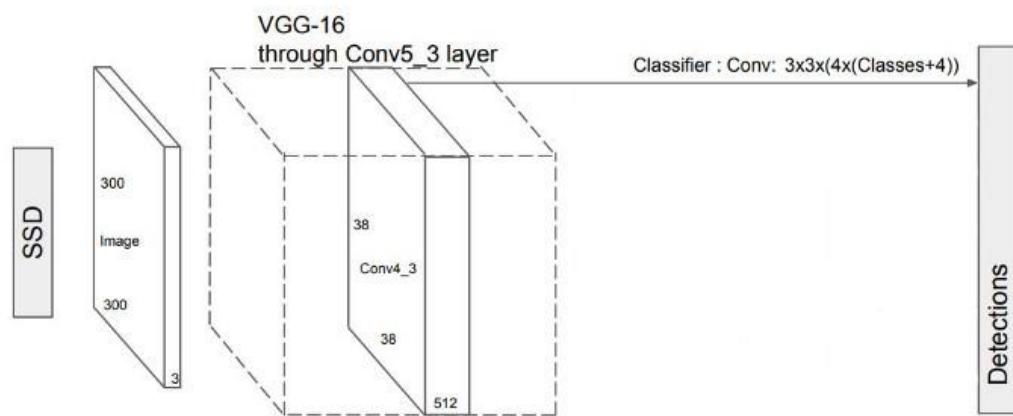
Berasal dari kata "Numerical Python", Numpy merupakan salah satu library pada Python yang berfungsi melakukan proses komputasi numerik. Sedangkan Array merupakan kumpulan variabel yang memiliki tipe data yang sama. Numpy menyimpan datanya dalam bentuk array. Bentuk dari numpy array adalah multidimensional yang mana dapat berupa 1-dimensi maupun 2-dimensi. Array 1-dimensi adalah sekumpulan data yang berisikan nama variabel dan tipe data yang sama yang dapat diakses menggunakan 1 buah index saja. Sedangkan array 2-dimensi adalah sekumpulan data yang berisikan nama dan tipe data yang sama dimana elemennya dapat diakses menggunakan 2 buah index yaitu index kolom dan index baris.

Dalam mengimplementasikan library numpy kita harus terlebih dahulu memanggil library. Setelah import library numpy, kita dapat melakukan array. Array1 merupakan contoh bentuk array 1-dimensi dengan perintah mengambil index pertama array yaitu index 0. Hasil yang keluar adalah angka 1 yang berada di urutan pertama. Array2 merupakan bentuk dari array 2-dimensi. Terdapat fungsi .shape pada array multidimensional yang menghasilkan tuple berisikan jumlah baris dan jumlah kolom, contohnya seperti berikut. Menjelaskan bahwa pada array2 berisikan 2 kolom dan 6 baris data. (Kurniasari, 2020)

3.8 SSD Single Shot MultiBox Detector

SSD dirancang untuk deteksi objek secara real-time. R-CNN yang lebih cepat menggunakan jaringan proposal wilayah untuk membuat kotak batas dan menggunakan kotak tersebut untuk mengklasifikasikan objek. Meskipun dianggap sebagai permulaan seni dalam akurasi, seluruh proses berjalan pada 7 frame per detik. Jauh di bawah apa yang dibutuhkan pemrosesan real-time. SSD mempercepat proses dengan menghilangkan kebutuhan akan jaringan proposal wilayah. Untuk memulihkan penurunan akurasi, SSD menerapkan beberapa peningkatan termasuk fitur multi-skala dan kotak default. Peningkatan ini memungkinkan SSD untuk mencocokkan akurasi Faster R-CNN menggunakan gambar beresolusi lebih rendah, yang selanjutnya mendorong kecepatan lebih tinggi. Menurut perbandingan berikut, ia mencapai kecepatan pemrosesan waktu nyata dan bahkan mengalahkan akurasi R-CNN yang Lebih Cepat.

SSD tidak menggunakan jaringan proposal wilayah yang didelegasikan. Sebaliknya, itu memutuskan untuk metode yang sangat sederhana. Ini menghitung lokasi dan skor kelas menggunakan filter konvolusi kecil. Setelah mengekstrak peta fitur, SSD menerapkan filter konvolusi 3×3 untuk setiap sel untuk membuat prediksi. (Filter ini menghitung hasil seperti filter CNN biasa.) Setiap filter menghasilkan 25 saluran: 21 skor untuk setiap kelas ditambah satu kotak batas. (Hui, 2018)



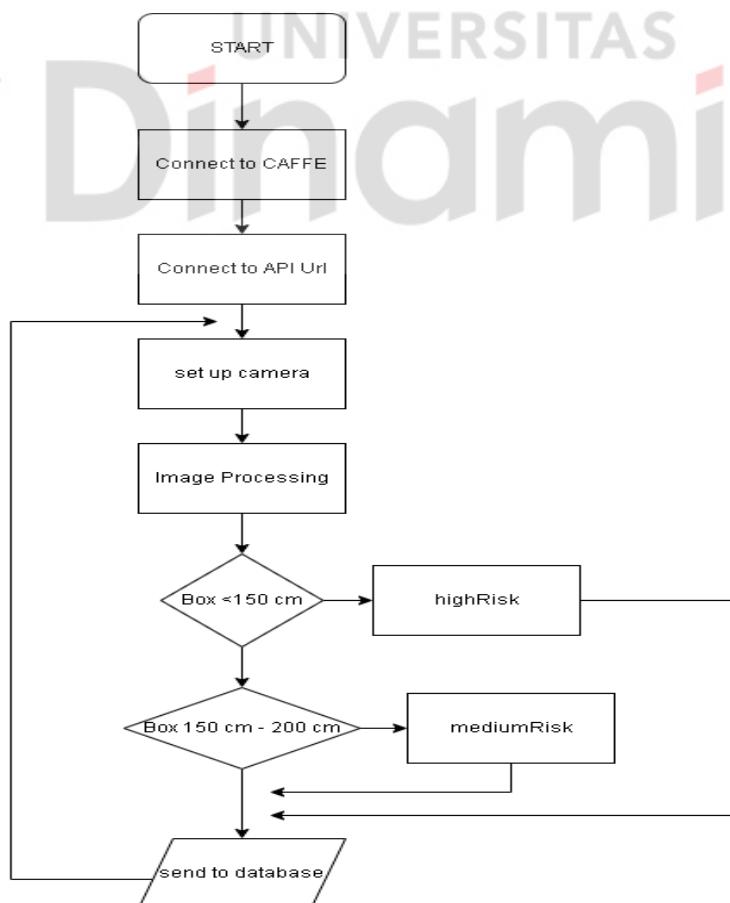
Gambar 3 6 SSD Single Shot MultiBox Detector

BAB IV

DESKRIPSI HASIL PROJECT

4.1 FlowChart

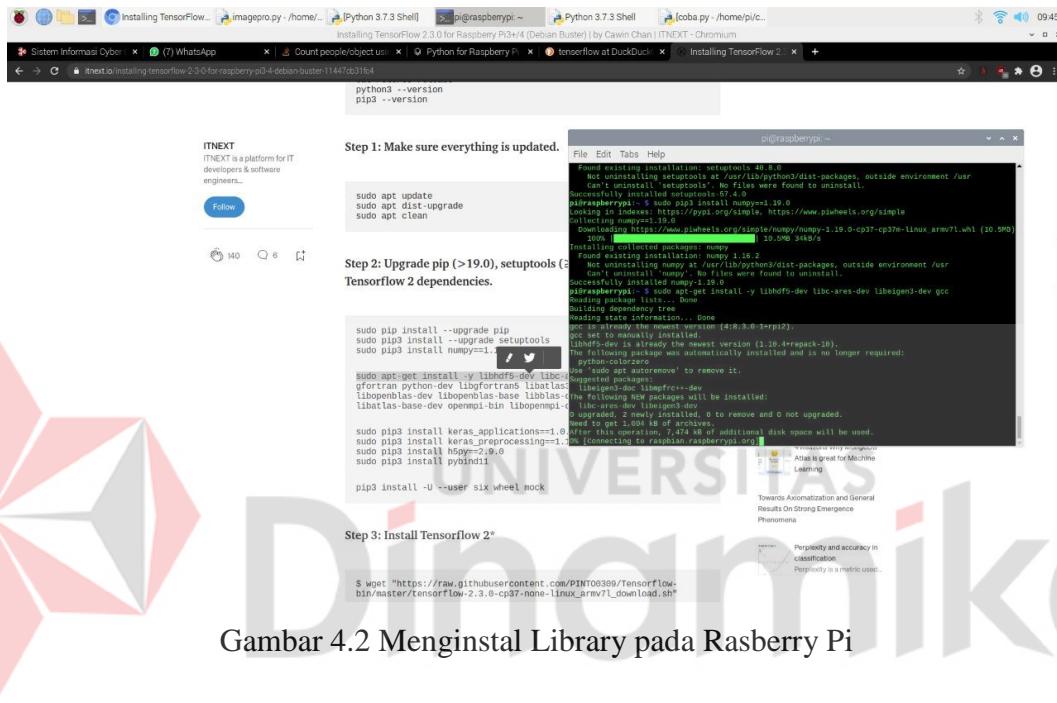
Pada gambar 4.4 flowchart sistem perlu melakukan koneksi ke data *pre-trained* CAFFE setelah itu mengoneksikan ke API melalui url yang ada pada database setelah proses pengoneksian tersebut sistem dapat melakukan set up kamera untuk melakukan image prosesing gambar yang ditangkap, pada proses tersebut sistem memiliki kondisi untuk menentukan jarak antar box pada kamera. Jika box berjarak kurang dari 150 cm antar box maka data yang keluar adalah highRisk, jika box berjarak antara 150 cm dan 200 cm antar box maka data yang keluar adalah mediumRisk, setelah itu data-data yang didapat dikirimkan ke database yang telah terkoneksi melalui API url.



Gambar 4 1 FlowChart

4.2 Menginstal Library pada Rasberry Pi

Menginstall Library pada Rasberry Pi agar program yang digunakan untuk mendekteksi jarak antar orang. Proses instalasi library tersebut membutuhkan beberapa library seperti caffe, API, dan tensorflow sebagai *pre-trained* deep learning pada program.



Gambar 4.2 Menginstal Library pada Rasberry Pi

List library yang diinstal:

1. import cv2
2. import numpy as np
3. import pymysql.cursors
4. import mysql.connector
5. import socket
6. import requests

4.3 Membuat program

Pada pembuatan program disini memerlukan beberapa hal seperti program pendekteksian jarak antar orang, program mendekteksi manusia, program

pengkoneksian API, program pengkoneksian CAFFE, dan program pengiriman data disini penjelasan setiap program tersebut akan dijelaskan.

4.3.1 Mendekteksi Manusia

Pada program pendekteksiaan manusia disini memerlukan library openCV untuk melakukan *image processing*. *Contrast Limited AHE* (CLAHE) adalah algoritma yang digunakan untuk mendekteksi manusia berdasarkan contras terang dan gelap pada warna yang bertujuan untuk mengurangi masalah noise pada gambar sehingga dapat mendekteksi lebih baik. Selain itu pada pendekteksian manusia juga diberikan pembuatan kotak atau *box* sebagai indicator jika sistem mendekteksi manusia yaitu program centroid, disini centroid dapat menentukan ukuran pada kotak, dan ketinggian kota tersebut.

Program Mendekteksi Manusia :

```
def CLAHE(bgr_image: np.array) -> np.array:
    hsv = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2HSV)
    hsv_planes = cv2.split(hsv)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    hsv_planes[2] = clahe.apply(hsv_planes[2])
    hsv = cv2.merge(hsv_planes)
    return cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

def centroid(startX,endX,startY,endY):
    centroid_x = round((startX+endX)/2,4)
    centroid_y = round((startY+endY)/2,4)
    bboxHeight = round(endY-startY,4)
    return centroid_x,centroid_y,bboxHeight

def calcDistance(bboxHeight):
    distance = (calculateConstant_x * calculateConstant_y) /
bboxHeight
    return distance

def drawResult(frame,position):
    for i in position.keys():
        if i in highRisk:
            rectangleColor = RED
        elif i in mediumRisk:
            rectangleColor = YELLOW
        else:
            rectangleColor = GREEN
        (startX, startY, endX, endY) = detectionCoordinates[i]

        cv2.rectangle(frame, (startX, startY), (endX, endY),
rectangleColor, 2)
```

4.3.2 Koneksi CAFFE

Pada program pengkoneksian CAFFE dibutuhkan tipe file yang akan dilakukan training untuk melakukan pendekteksian jarak, sehingga CAFFE dapat memberikan hasil antar *box* yang telah dibuat pada pendekteksian manusia. Pada hal ini metode yang digunakan untuk mendekteksi jarak antar manusia adalah metode SSD *singgel shot multibox detector* yang di disain untuk mendekteksi objek gambar secara *real-time* dengan akurasi pendekteksian yang baik. Pada hal ini file yang akan dilakukan pendekteksian berupa *video capture* pada kamera web cam atau CCTV.

Program Koneksi CAFFE :

```
if __name__ == "main__":
    caffeNetwork =
        cv2.dnn.readNetFromCaffe("./SSD_MobileNet_prototxt.txt",
        "./SSD_MobileNet.caffemodel")
        cap = cv2.VideoCapture(0)
        cap.set(cv2.CAP_PROP_FPS, 30)
        fourcc = cv2.VideoWriter_fourcc(*"XVID")
        output_movie = cv2.VideoWriter("./result.avi", fourcc, 120,
        (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
        int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))))

    while cap.isOpened():
        debug_frame, frame = cap.read()
        highRisk = set()
        mediumRisk = set()
        position = dict()
        detectionCoordinates = dict()

        if not debug_frame:
            print("Video cannot opened or finished!")
            break

        if preprocessing:
            frame = CLAHE(frame)

            (imageHeight, imageWidth) = frame.shape[:2]
            pDetection = cv2.dnn.blobFromImage(cv2.resize(frame,
            (imageWidth, imageHeight)), 0.007843, (imageWidth, imageHeight),
            127.5)

            caffeNetwork.setInput(pDetection)
            detections = caffeNetwork.forward()

            for i in range(detections.shape[2]):
                accuracy = detections[0, 0, i, 2]
                if accuracy > accuracyThreshold:
                    # Detection class and detection box coordinates.
                    idOfClasses = int(detections[0, 0, i, 1])
                    box = detections[0, 0, i, 3:7] *
                        np.array([imageWidth, imageHeight, imageWidth, imageHeight])
```

```

        (startX, startY, endX, endY) = box.astype('int')

        if idOfClasses == personLabelID:
            # Default drawing bounding box.
            bboxDefaultColor = (255,255,255)
            cv2.rectangle(frame, (startX, startY), (endX,
endY), bboxDefaultColor, 2)
            detectionCoordinates[i] = (startX, startY,
endX, endY)

            # Centroid of bounding boxes
            centroid_x, centroid_y, bboxHeight =
centroid(startX,endX,startY,endY)
            distance = calcDistance(bboxHeight)
            # Centroid in centimeter distance
            centroid_x_centimeters = (centroid_x *
distance) / calculateConstant_y
            centroid_y_centimeters = (centroid_y *
distance) / calculateConstant_y
            position[i] = (centroid_x_centimeters,
centroid_y_centimeters, distance)

```



1. cap.set(cv2.CAP_PROP_FPS, 30)

Untuk menyetting pengambilan data pada kamera perdetik dalam 30 FPS (*frame per second*).

2. pDetection = cv2.dnn.blobFromImage(cv2.resize(frame, (imageWidth, imageHeight)), 0.007843, (imageWidth, imageHeight), 127.5)

Untuk melakukan pemrosesan ukuran gambar yang didapat dari ukuran frame yang akan dikalian 0.007843 serta lebar dan tinggi gambar yang akan dikurangi 127.5 data tersebut berasal dari caffe sendiri dan tidak dapat diubah karena *pre-trained* program.

3.

```

accuracy = detections[0, 0, i, 2]
    if accuracy > accuracyThreshold:
        # Detection class and detection box coordinates.
        idOfClasses = int(detections[0, 0, i, 1])
        box = detections[0, 0, i, 3:7] *
np.array([imageWidth, imageHeight, imageWidth, imageHeight])

```

program ini digunakan untuk mengambil nilai akurasi yang didapat dari tinggi dan lebar pada program pemrosesan frame berdasarkan koordinasi pada *box* dengan perbandingan 3:7 yang dikalikan dengan ukuran gambar .

```

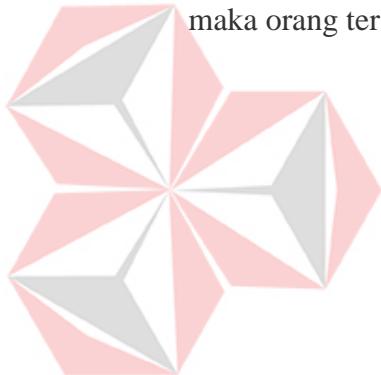
4. # Default drawing bounding box.
        bboxDefaultColor = (255,255,255)
        cv2.rectangle(frame, (startX, startY),
        (endX, endY), bboxDefaultColor, 2)
        detectionCoordinates[i] = (startX,
        startY, endX, endY)

```

pada program diatas berfungsi sebagai penggambaran *bounding box* atau kotak yang akan digunakan sebagai indicator pendekteksian yang diberikan warna pada line kotak tersebut dengan warna `bboxDefaultColor = {255,255,255,}` berdasarkan RGB R=255 B=255 G=255 sehingga warna default yang digunakan adalah warna putih.

4.3.3 Kondisi Jarak Pada Box

Pada program Kondisi Jarak Pada *box* diberikan 2 buah kondisi yaitu: bila box berwarna hijau maka orang tersebut berkondisi aman, jika box berwarna kuning maka orang tersebut berkondisi cukup dekat, jika box berwarna merah maka orang tersebut berkondisi berbahaya atau jarak terlalu dekat.



Gambar 4.3 Kondisi Jarak Pada Box

Tabel 4.1 Kondisi Jarak Pada Box

| NO | Jarak antar orang | Kondisi |
|----|-------------------|-----------|
| 1 | 50cm | High Risk |
| 2 | 200cm | Safe |

| | | |
|----|-------|-------------|
| 3 | 100cm | High Risk |
| 4 | 300cm | Safe |
| 5 | 150cm | Medium Risk |
| 6 | 160cm | Medium Risk |
| 7 | 170cm | Medium Risk |
| 8 | 40cm | High Risk |
| 9 | 250cm | Safe |
| 10 | 90cm | High Risk |

Pada tabel diatas dilakukan sebanyak 10 kali percobaan untuk mendapatkan hasil kondisi pada sistem kondisi jara pada *box*, setiap percobaan dilakukan pengukuran jarak pada setiap orang untuk mendapatkan nilai kondisi yang diinginkan. Sehingga hasil yang didapatkan dari 10 kaliperobaan tersebut mendapatkan hasil keberhasilan 100%.



Program Kondisi Jarak Pada Box:

```

for i in position.keys():
    for j in position.keys():
        if i < j:
            distanceOfBboxes = sqrt(pow(position[i][0]-
position[j][0],2)+ pow(position[i][1]-position[j][1],2)
+ pow(position[i][2]-position[j][2],2))
            if distanceOfBboxes < 150: # 150cm or lower
                highRisk.add(i),highRisk.add(j)
            elif distanceOfBboxes < 200 > 150: # between
150 and 200
                mediumRisk.add(i),mediumRisk.add(j)
  
```

Jika pada posisi dari sebuah box dibawah 150 cm atau kurang maka kondisi yang diberikan adalah memasukan data nilai hingrisk kepada 2 buah posisi kotak, bila sebuah antar kotak berjarak antara 150 cm – 200 cm maka posisi yang diberikan kepada dua posisi kotak tersebut bernilai mediumRisk. Program tersebut berjalan berdasarkan library openCV sebagai *image precessing* pendekteksi gambar citra yang ada pada kamera yang dibantu oleh numpy untuk penghitungan jarak antar 2 kotak menggunakan array, disini library caffe juga diperlukan untuk mendekteksi manusia, library tersebut telah di training sehingga program tidak perlu melakukan tranning data untuk mendapatkan persentase

akurasi dekripsi manusia, data training yang diambil tersebut diambil melalui caffenetwork menggunakan internet.

4.4 Pengiriman data ke database

Pada percobaan pengiriman data ke database hal pertama yang dibutuhkan ada library API sebagai pengiriman data. Pengiriman data tersebut harus menyesuaikan dengan tipe data pada database dan lokasi peletakan data sehingga data dapat diterima oleh database.



Python 3.7.3 Shell

dit Shell Debug Options Window Help

```
array key "status" in <b>C:\\xampp\\htdocs\\realtime\\send.php</b> on line 11<br />\n{"value":1,"message":"Data berhasil masuk"}  
ss: Send to API...  
>\n<b>Warning</b>: Undefined array key "status" in <b>C:\\xampp\\htdocs\\realtime\\send.php</b> on line 11<br />\n{"value":1,"message":"Data berhasil masuk"}  
ss: Send to API...  
>\n<b>Warning</b>: Undefined array key "jml_high" in <b>C:\\xampp\\htdocs\\realtime\\send.php</b> on line 10<br />\n<b>Warning</b>: Undefined array key "status" in <b>C:\\xampp\\htdocs\\realtime\\send.php</b> on line 11<br />\n{"value":1,"message":"Data berhasil masuk"}  
ss: Send to API...  
>\n<b>Warning</b>: Undefined array key "jml_high" in <b>C:\\xampp\\htdocs\\realtime\\send.php</b> on line 10<br />\n<b>Warning</b>: Undefined array key "status" in <b>C:\\xampp\\htdocs\\realtime\\send.php</b> on line 11<br />\n{"value":1,"message":"Data berhasil masuk"}  
ss: Send to API...
```

Gambar 4.4 Pengiriman data ke database

Tabel 4.2 Pengiriman data ke database

| NO | Data | Kondisi | Database |
|----|-------------|-------------------|-------------|
| 1 | High Risk | Berhasil Mengirim | High Risk |
| 2 | Medium Risk | Berhasil Mengirim | Medium Risk |
| 3 | Safe | Berhasil Mengirim | Safe |
| 4 | Safe | Berhasil Mengirim | Safe |
| 5 | Medium Risk | Berhasil Mengirim | Medium Risk |
| 6 | Medium Risk | Berhasil Mengirim | Medium Risk |
| 7 | Medium Risk | Berhasil Mengirim | Medium Risk |
| 8 | High Risk | Berhasil Mengirim | High Risk |
| 9 | Safe | Berhasil Mengirim | Safe |
| 10 | High Risk | Berhasil Mengirim | High Risk |

Pada percobaan pengiriman data ke database dilakukan 10 kali percobaan yang diambil data pada hasil percobaan kondisi jarak pada box dengan melihat apakah data tersebut berhasil mengirim , dan apakah data tersebut sama dengan data yang dikirimkan pada sistem. Pada tabel diatas dapat disimpulkan dari 10 kali percobaan yang dilakukan mendapatkan hasil yang sesuai dengan harapan, sehingga percobaan tersebut dapat disimpulkan berhasil dengan persentase 100%.

Program pengiriman data ke database:

```
jumlahorg = detectionCoordinates
jumlahmedium = mediumRisk
jumlahhigh = highRisk

hasil_jml_org = jumlahorg
hasil_jml_medium = jumlahmedium
hasil_jml_high = jumlahhigh
myobj = {'jml_org': hasil_jml_org, 'jml_medium':
hasil_jml_medium, 'jml_high': hasil_jml_high}
print ('Process: Send to API...')
x = requests.post(url, data = myobj)
print(x.content)
```

Pada program diatas adalah program untuk pengiriman data melalui API melalui IP local yang ada pada database yang diambil nilai data dari jumlah orang, mediumRisk, dan highRisk yang natinya nilai tersebut akan dimasukan ke database yang telah disediakan pada tabel database. Jika sebuah data telah berhasil mengirimkan maka sistem akan memberikan pesan telah berhasil mengirim melalui API.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil Kerja Praktik yang dilakukan dan selama proses penggerjaan *project Pengelolaan Sistem Physical Distancing Di Dalam Ruangan Rapat* dapat disimpulkan bahwa :

1. Penginstalan library yang dilakukan dapat berjalan dengan baik tanpa adanya kesalahan pada instalasi.
2. Percobaan program yang dilakukan secara live dapat mendeteksi seseorang serta jarak dengan baik sesuai dengan kondisi yang ditentukan dengan parameter jarak centimeter, dan dari percobaan tersebut dapat disimpulkan tingkat keberhasilan percobaan yang dilakukan sebesar 100%.
3. Pengiriman data ke database yang dilakukan dapat berjalan dengan baik tanpa adanya kesalahan pengiriman data, sehingga pengiriman data yang dilakukan dapat berjalan dengan keberhasilan 100% tanpa adanya kesalahan pengiriman data.

5.2 Saran

Berdasarkan hasil Kerja Praktik yang dilakukan dan hasil kesimpulan yang didapatkan oleh penulis, terdapat beberapa saran yang dapat dijadikan pengembangan dari penelitian berikutnya, yaitu :

1. Gunakan program training sendiri sehingga hasil yang ditampilkan mendapatkan persentase akurasi yang baik
2. Tambahkan program untuk membaca jumlah pixel pada gambar untuk dikonversikan kedalam satuan centimeter.
3. Gunakan raspberry pi yang memiliki *graphic card* seperti Nvidia K40

DAFTAR PUSTAKA

- Lab Elektronika. (2018, Juni 5). *MENGENAL SINGLE BOARD KOMPUTER RASPBERRY Pi 3 MODEL B+*. Retrieved Januari 11, 2022, from Lab Elektronika: <http://www.labelektronika.com/2018/06/mengenal-raspberry-pi-3-model-b-plus.html>
- Lawrence, A. (2020, Oktober 24). *API: Pengertian, Fungsi, dan Cara Kerjanya*. Retrieved Januari 11, 2022, from NIAGAHOSTER: <https://www.niagahoster.co.id/blog/api-adalah/>
- Muhardian, A. (2022, Januari 3). *Belajar Pemrograman Python: Pengenalan Dasar Python dan Persiapan Awal*. Retrieved Januari 11, 2022, from Petani Kode: <https://www.petanikode.com/python-linux/>
- Recode. (2020, Januari 9). *Ultimate beginner's guide to Caffe for Deep Learning*. Retrieved Januari 11, 2022, from Recode: <https://recodeminds.com/blog/a-beginners-guide-to-caffe-for-deep-learning/>
- Setiawan, R. (2021, Oktober 9). *Mengenal Deep Learning Lebih Jelas*. Retrieved Januari 11, 2022, from dicoding: <https://www.dicoding.com/blog/mengenal-deep-learning/>
- Hanugra Aulia Sidharta, S. M. (2017, Oktober 28). *INTRODUCTION TO OPEN CV*. Retrieved Januari 13, 2022, from binus: <https://binus.ac.id/malang/2017/10/introduction-to-open-cv/>
- Kurniasari, D. (2020, Desember 7). *Belajar Numpy Array Python, Fungsi Yang Populer Dalam Proses Manipulasi Data*. Retrieved Januari 13, 2022, from DALab: <https://dqlab.id/belajar-numpy-array-python-bersama-dqlab>
- Lina, Q. (2019, Januari 2). *Apa itu Convolutional Neural Network?* Retrieved Januari 11, 2022, from Medium: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>