

**LAPORAN TUGAS BESAR**  
**KU1102 PENGENALAN KOMPUTASI**  
**SEMESTER I 2022-2023**

**AUTONOMOUS CAR**



Disusun oleh:

Rizki Dwi Syarif	16622333
Nasa Amelia Rakhmat	16622348
Carissa Febriana	16622350
Cantikia Ayuningcahya	16622359
Mohamad Guntur Sutrisno	16622370
Muhammad Abiyyu Hafiansyah	16622371
Reyhan Kharisma Ramadhan	16622376

Dosen Pengajar:

Hadi Kardana, S.T., M.T., Ph.D.

Faizal Immaddudin Wira Rohmat, S.T., M.T., Ph.D.

**PROGRAM TAHAP PERSIAPAN BERSAMA**  
**FAKULTAS TEKNIK SIPIL DAN LINGKUNGAN**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2022**

## **KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan tugas besar yang berjudul “Autonomous Car”.

Adapun tujuan dari penulisan dari laporan ini adalah untuk memenuhi tugas akhir pada mata kuliah Pengenalan Komputasi. Selain itu, laporan ini juga bertujuan untuk menambah wawasan dan mengetahui tentang bagaimana sebuah mobil otomatis dapat bekerja dengan menggunakan bahasa pemrograman bagi para pembaca serta penulis.

Kelancaran dalam mempersiapkan dan menyelesaikan laporan tugas besar ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penyusun mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Hadi Kardhana, S.T., M.T., Ph.D., dan,
2. Bapak Faizal Immadudin Wira Rohmat, S.T., M.T., Ph.D., selaku dosen mata kuliah Pengenalan Komputasi,
3. Semua pihak yang telah memberikan masukan dan dorongan dalam menyelesaikan laporan tugas besar ini.

Penyusun menyadari bahwa masih banyak kesalahan dalam penyusunan laporan ini, baik dari segi teknik penulisan maupun tata bahasa. Oleh karena itu, berbagai bentuk kritik dan saran yang membangun sangat kami harapkan.

Penyusun berharap bahwa laporan ini akan sangat bermanfaat bagi siapa saja yang membaca dan menambah pengetahuan bagi kita semua.

Jatinangor, 14 November 2022

## DAFTAR ISI

KATA PENGANTAR .....	2
----------------------	---

DAFTAR ISI .....	3
------------------	---

### BAB I PENDAHULUAN

1.1 Latar Belakang.....	4
-------------------------	---

1.2 Rumusan Masalah.....	5
--------------------------	---

1.3 Batasan Masalah .....	6
---------------------------	---

1.4 Tujuan .....	6
------------------	---

### BAB II PEMBAHASAN

2.1 Deskripsi Program .....	7
-----------------------------	---

2.2 <i>Flowchart</i> .....	8
----------------------------	---

2.3 Algoritma .....	9
---------------------	---

2.4 Simulasi Program .....	13
----------------------------	----

### BAB III DATA

3.1 Analisis Data .....	15
-------------------------	----

3.2 Deskripsi Data .....	15
--------------------------	----

3.3 Karakteristik Data.....	15
-----------------------------	----

3.4 Visualisasi Data .....	16
----------------------------	----

### BAB IV PENUTUP

3.1 Kesimpulan.....	19
---------------------	----

3.2 Saran .....	19
-----------------	----

3.3 Pembagian Tugas.....	20
--------------------------	----

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan di dunia teknologi terjadi sangat pesat pada masa era digital ini. Bidang teknologi yang berkembang tidak hanya berpusat pada satu bidang saja. Namun, aspek komunikasi, informasi, telekomunikasi, maupun transportasi harus mengalami perkembangan signifikan yang bisa menunjang menuju kehidupan yang lebih baik. Pesatnya pertumbuhan inovasi teknologi berbasis *Artificial Intelligence* di sektor kendaraan sangat terlihat, ditandai dengan diproduksi mobil-mobil masal yang berteknologi tinggi. Terkait dengan tingginya akan kebutuhan transportasi, maka salah satu dampaknya adalah kecelakaan lalu lintas. Kecelakaan lalu lintas menurut UU RI NO. 22 tahun 2009 adalah suatu peristiwa di jalan raya yang tidak diduga dan tidak disengaja melibatkan kendaraan dengan atau tanpa pengguna jalan lain yang mengakibatkan korban manusia dan atau kerugian harta benda. Berdasarkan statistik data kecelakaan dan pelanggaran lalu lintas kepolisian Republik Indonesia untuk bagian Daerah Istimewa Yogyakarta pada tahun 2021 tercatat terdapat 5350 jumlah kecelakaan dengan korban meninggal dunia hingga 452 jiwa. Dari data tersebut dapat diketahui bahwa kecelakaan lalu lintas merupakan salah satu masalah yang sangat serius untuk diselesaikan selagi memakan korban jiwa. Seiring bertambahnya jumlah mobil yang beredar, dan mengingat banyaknya angka kecelakaan yang terjadi di Indonesia, maka tidaklah mudah untuk mengurangnya. Kecelakaan yang terjadi memiliki beberapa faktor, di antaranya ada kelalaian pengemudi atas ketertiban lalu lintas seperti mengantuk, kendaraan yang bermasalah, ataupun faktor-faktor yang lainnya. Dengan angka kecelakaan sebesar itu, maka seharusnya sudah ada upaya untuk mengurangnya, salah satu contohnya adalah “Autonomous Car”.

Mobil otonom (juga dikenal sebagai mobil tanpa sopir, mobil penggerak sendiri, mobil robotik) atau *autonomous car* merupakan mobil yang dapat berjalan atau bergerak secara otomatis sesuai dengan yang diperintahkan oleh pemilik mobil, dengan begitu pemilik mobil bisa lebih merasa nyaman dan tenang saat berada di dalam mobil. Autonomous car dapat mengurangi risiko kecelakaan jika persyaratannya terpenuhi, semisal marka jalan yang dibuat secara baik,

kondisi jalan yang sesuai dengan peta dan sebagainya. Oleh karena itu, mengapa *autonomous car* belum bisa dijalankan di Indonesia disebabkan banyaknya persyaratan yang belum terpenuhi ditambah dengan kondisi pengendara di Indonesia yang kurang mematuhi peraturan berkendara di jalan raya. Namun, sudah banyak *semi-autonomous car* yang dipasarkan di Indonesia, dengan mengusung sistem *self-driving car* yang sudah memiliki sistem keamanan lebih, memungkinkan mobil memberi peringatan kepada pengemudi jika ada kendaraan yang terlalu dekat dengan mobil, mampu menjaga jarak aman mobil dengan kendaraan 2 di depannya, lalu ada *line keeping assist* yang berguna untuk menjaga mobil selalu pada jalurnya, terutama saat pengendara mengantuk dan terindikasi akan keluar jalur, serta sistem keamanan lainnya yang disematkan pada mobil tersebut. Sehingga, pembuatan program Autonomous Car dengan sistem pencegah kecelakaan bertujuan untuk mengatasi tingginya tingkat kecelakaan mobil yang penyebab terbesarnya adalah pengemudi. Autonomous car merupakan sebuah sistem yang dijalankan oleh komputer yang tidak memiliki perasaan mengantuk ataupun kelelahan seperti manusia, maka diharapkan mampu lebih meningkatkan keamanan saat berkendara.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa permasalahan yang akan dikaji dalam laporan tugas besar ini:

1. Bagaimana cara mendesain sistem *autonomous car* yang dapat memiliki kemampuan dalam menganalisis garis markah jalan serta mendeteksi mobil di sekitarnya?
2. Bagaimana cara menganalisis kerja sistem serta cara kerja dari *autonomous car*?
3. Bagaimana mengimplementasikan cara kerja *autonomous car* dalam bahasa pemrograman Python versi 3.0 menggunakan beberapa modul yang tersedia di dalamnya?

### **1.3 Batasan Masalah**

Adapun batasan masalah pada pembuatan program ini yang di antaranya adalah:

1. Belum dilengkapi dengan objek yang terdapat pada jalan
2. Jalan yang datar

### **1.4 Tujuan**

Tujuan dasar atas pembuatan laporan tugas ini adalah untuk merancang sebuah program *autonomous car* dalam bahasa pemrograman Python 3.0 yang mampu berjalan sesuai dengan kemampuan dasarnya, yaitu berjalan mengikuti garis markah jalan dengan jalan yang berbelok-belok. Sehingga, kita dapat mengetahui cara kerja dari *autonomous car* tersebut.

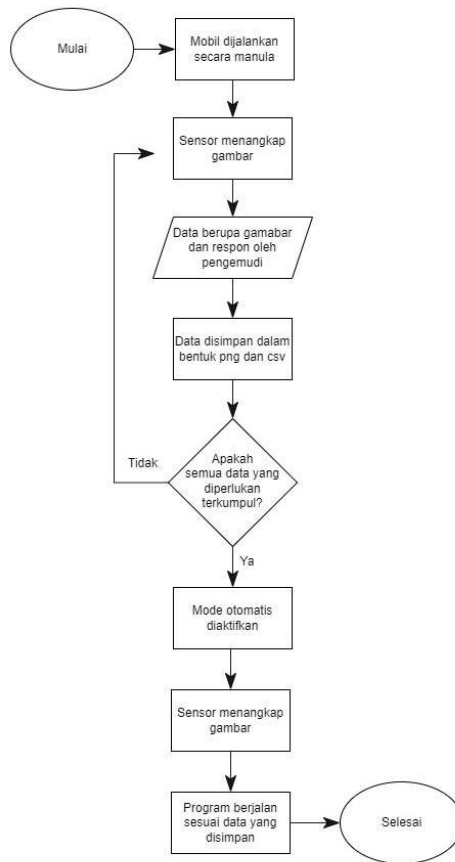
## **BAB II**

### **PEMBAHASAN**

#### **2.1 Deskripsi Program**

Autonomous car merupakan mobil yang dapat beroperasi dengan aman dan efektif tanpa perlu dikendalikan oleh manusia. Kendaraan ini terdiri atas kumpulan sistem-sistem yang saling bekerja sama untuk memungkinkan kendaraan tersebut melintasi lingkungannya. Autonomous car memakai berbagai Teknik untuk mendeteksi hal-hal di sekitarnya, seperti radar, sinar laser, GPS, odometri, dan penglihatan computer. Ada alasan teknologi autonomous ini dikembangkan, tujuannya yaitu untuk keamanan berkendara, dari teknologi ini juga dapat diprediksi mampu mengurangi 90 persen kecelakaan mobil di Amerika Serikat. Karena kebanyakan kecelakaan terjadi disebabkan oleh kesalahan pengemudi atau human error. Kegunaan lain dari teknologi ini juga dapat menambah efisiensi perjalanan, dapat membuat manusia lebih produktif, dan lebih ramah lingkungan. Beberapa kelebihan dari autonomous car yaitu yang pertama lebih nyaman karena tidak perlu menginjak kopling saat macet, bisa melakukan engine brake, lebih hemat kopling, diklaim lebih irit BBM, selain kelebihan terdapat juga kekurangan dari mobil autonomous car yaitu ada rasa tersendat saat berpindah gigi secara otomatis, tidak cukup responsive (lemot), jika rusak biaya perbaikannya cukup mahal.

## 2.2 Flowchart





## 2.3 Algoritma

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import os
5 from sklearn.utils import shuffle
6 import matplotlib.image as mpimg
7 from imgaug import augmenters as iaa
8 import cv2
9 import random
10
11 from tensorflow.keras.models import Sequential
12 from tensorflow.keras.layers import Convolution2D, Flatten, Dense
13 from tensorflow.keras.optimizers import Adam
14
15 def getName(filePath):
16     return filePath.split('\\')[-1]
17
18 def importDataInfo(path):
19     coloums = ['Center', 'Left', 'Right', 'Steering', 'Throttle', 'Brake', 'Speed']
20     data = pd.read_csv(os.path.join(path, 'driving_log.csv'), names=coloums)
21     print(data.head())
22     print(data['Center'][0])
23     print(getName(data['Center'][0]))
24     data['Center'] = data['Center'].apply(getName)
25     print(data.head())
26     print('Total Images Imported:', data.shape[0])
27     return data
28
29 def balanceData(data, display=True):
30     nBins = 31
31     samplesPerBin = 1000
32     hist, bins = np.histogram(data['Steering'], nBins)
33     # print(bins)
34     if display:
35         center = (bins[:-1] + bins[1:]) * 0.5
36         # print(center)
37         plt.bar(center, hist, width = 0.06)
38         plt.plot((-1, 1), (samplesPerBin, samplesPerBin))
39         plt.show()
40
41     removeIndexList = []
42     for j in range(nBins):
43         binDataList = []
44         for i in range(len(data['Steering'])):
45             if data['Steering'][i] >= bins[j] and data['Steering'][i] <= bins[j+1]:
46                 binDataList.append(i)
47             binDataList = shuffle(binDataList)
48             binDataList = binDataList[samplesPerBin:]
49             removeIndexList.extend(binDataList)
50     print('Removed Images: ', len(removeIndexList))
51     data.drop(data.index[removeIndexList], inplace = True)
52     print('Remaining Images: ', len(data))
53
54     if display:
55         hist, _ = np.histogram(data['Steering'], nBins)
56         plt.bar(center, hist, width = 0.06)
57         plt.plot((-1, 1), (samplesPerBin, samplesPerBin))
58         plt.show()
59
60     return data
61
62 def loadData(path, data):
63     imagesPath = []
64     steering = []
65     for i in range(len(data)):
66         indexedData = data.iloc[i]
67         imagesPath.append(os.path.join(path, 'IMG', indexedData[0]))
68         steering.append(float(indexedData[3]))
69     imagesPath = np.asarray(imagesPath)
70     steering = np.asarray(steering)
71     return imagesPath, steering
```

Pada file ini, diberikan nama **utlis.py** yang memiliki arti beberapa fungsi-fungsi dalam pembuatan model sensor yang akan digunakan pada program mobil otomasi. Fungsi-fungsi yang terdapat pada file ini berupa **getName** yang mendefinisikan untuk mengubah alamat file menjadi nama filenya saja. **importDataInfo** yang mendefinisikan pembacaan file gambar yang telah direkam mobil pada file .csv. **balanceData** yang berfungsi sebagai pendefinisian ‘Steering Wheel’ dalam file agar file yang tidak digunakan dapat dihapus. **loadData** merupakan proses fungsi sebelum data diolah menjadi angka-angka. Selanjutnya, **augmentImage** yang berfungsi untuk mengargumentasikan gambar-gambar yang telah terekam menjadi lebih detail tanpa adanya objek lain selain jalanan. Fungsi ini

```

1  def augmentImage(imgPath, steering):
2      img = mpimg.imread(imgPath)
3      ## PAN
4      if np.random.rand() < 0.5:
5          pan = iaa.Affine(translate_percent={'x':(-0.1,0.1), 'y':(-0.1,0.1)})
6          img = pan.augment_image(img)
7      ## ZOOM
8      if np.random.rand() < 0.5:
9          zoom = iaa.Affine(scale=(1,1.2))
10         img = zoom.augment_image(img)
11     ## BRIGHTNESS
12     if np.random.rand() < 0.5:
13         brightness = iaa.Multiply((0.4,1.2))
14         img = brightness.augment_image(img)
15     ## FLIP
16     if np.random.rand() < 0.5:
17         img = cv2.flip(img,1)
18         steering = -steering
19
20     return img, steering
21
22 # imgRe, st = augmentImage('test.jpg',0)
23 # plt.imshow(imgRe)
24 # plt.show()
25
26 def preProcessing(img):
27     img = img[60:135,:,:]
28     img = cv2.cvtColor(img, cv2.COLOR_RGB2YUV)
29     img = cv2.GaussianBlur(img, (3,3), 0)
30     img = cv2.resize(img, (200,66))
31     img = img/255
32     return img
33
34 imgRe = preProcessing(mpimg.imread('test.jpg'))
35 plt.imshow(imgRe)
36 plt.show()
37
38 def batchGen(imagesPath, steeringList, batchSize, trainFlag):
39     while True:
40         imgBatch = []
41         steeringBatch = []
42
43         for i in range(batchSize):
44             index = random.randint(0,len(imagesPath)-1)
45             if trainFlag:
46                 img, steering = augmentImage(imagesPath[index], steeringList[index])
47             else:
48                 img = mpimg.imread(imagesPath[index])
49                 steering = steeringList[index]
50                 img = preProcessing(img)
51                 imgBatch.append(img)
52                 steeringBatch.append(steering)
53             yield (np.asarray(imgBatch), np.asarray(steeringBatch))
54
55 def creatModel():
56     model = Sequential()
57
58     model.add(Convolution2D(24, (5, 5), (2, 2), input_shape= (66, 200, 3), activation='elu'))
59     model.add(Convolution2D(36, (5, 5), (2, 2), activation='elu'))
60     model.add(Convolution2D(48, (5, 5), (2, 2), activation='elu'))
61     model.add(Convolution2D(64, (3, 3), activation='elu'))
62     model.add(Convolution2D(64, (3, 3), activation='elu'))
63
64     model.add(Flatten())
65     model.add(Dense(100, activation='elu'))
66     model.add(Dense(50, activation='elu'))
67     model.add(Dense(10, activation='elu'))
68     model.add(Dense(1))
69
70     model.compile(Adam(lr=0.0001),loss='mse')
71
72     return model

```

menggunakan beberapa fungsi seperti *Pan* yang berguna untuk membuat data menjadi sebuah plot x dan y, *Zoom* yang berguna untuk memperbesar data agar tidak terlihat mobil, *Brightness* yang berguna untuk meningkatkan ataupun menurunkan cahaya pada data sehingga data lebih mudah untuk diproses, dan *Flip* yang berguna untuk membalikkan gambar. Setelah data di proses, fungsi tersebut membuat **return** img (gambar) untuk hasil akhir gambar, dan 'steering' untuk hasil akhir data steering. Selanjutnya, ada fungsi **preProcessing** yang berfungsi untuk membuat gambar menjadi lebih jelas dengan memproses menggunakan *GaussianBlur* yang terdapat pada modul CV2 (Computer Vision 2), mengubah warna, serta

mengatur ulang gambar dengan fungsi **resize**. Setelah selesai pada tahap *Pre-Processing*, data akan diproses melalui tahap *Processing Data* yaitu pada fungsi **batchGen**, dan **creatModel**.

```

1 print('Setting UP')
2 import os
3 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
4
5 from utlis import *
6 from sklearn.model_selection import train_test_split
7
8 path = 'data-mobil'
9 data = importDataInfo(path)
10
11 data = balanceData(data,display=False)
12
13 imagesPath, steerings = loadData(path,data)
14 # print(imagesPath[0], steering[0])
15
16 xTrain, xVal, yTrain, yVal = train_test_split(imagesPath, steerings, test_size=0.2, random_state=5)
17 print('Total Training Images: ', len(xTrain))
18 print('Total Validation Images: ', len(xVal))
19
20 model = creatModel()
21 model.summary()
22
23 history = model.fit(batchGen(xTrain, yTrain, 100, 1), steps_per_epoch=300, epochs=10,
24                     validation_data=batchGen(xVal,yVal,100,0), validation_steps=200)
25
26 model.save("model.h5")
27 print('Model Saved')
28
29 plt.plot(history.history['loss'])
30 plt.plot(history.history['val_loss'])
31 plt.legend(['Training', 'Validation'])
32 plt.ylim([0,1])
33 plt.title('Loss')
34 plt.xlabel('Epoch')
35 plt.show()

```

File disamping merupakan input dan pendefinsian variabel yang akan dimasukkan pada file **utlis.py** yang berisikan fungsi-fungsi yang telah dijelaskan pada di atas.

Terakhir, program mobil otomasi akan dijalankan menggunakan algoritma berikut yang berisi pembuatan *local server* untuk menjalankan program yang menggunakan port 4567.



```
1 print('Setting UP')
2 import os
3
4 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
5 import socketio
6 import eventlet
7 import numpy as np
8 from flask import Flask
9 from tensorflow.keras.models import load_model
10 import base64
11 from io import BytesIO
12 from PIL import Image
13 import cv2
14
15 sio = socketio.Server()
16 app = Flask(__name__) # '__main__'
17
18 maxSpeed = 10
19
20
21 def preProcess(img):
22     img = img[60:135, :, :]
23     img = cv2.cvtColor(img, cv2.COLOR_RGB2YUV)
24     img = cv2.GaussianBlur(img, (3, 3), 0)
25     img = cv2.resize(img, (200, 66))
26     img = img / 255
27     return img
28
29
30 @sio.on('telemetry')
31 def telemetry(sid, data):
32     speed = float(data['speed'])
33     image = Image.open(BytesIO(base64.b64decode(data['image'])))
34     image = np.asarray(image)
35     image = preProcess(image)
36     image = np.array([image])
37     steering = float(model.predict(image))
38     throttle = 1.0 - speed / maxSpeed
39     print(f'{steering}, {throttle}, {speed}')
40     sendControl(steering, throttle)
41
42
43 @sio.on('connect')
44 def connect(sid, environ):
45     print('Connected')
46     sendControl(0, 0)
47
48
49 def sendControl(steering, throttle):
50     sio.emit('steer', data={
51         'steering_angle': steering.__str__(),
52         'throttle': throttle.__str__()
53     })
54
55
56 if __name__ == '__main__':
57     model = load_model('model.h5')
58     app = socketio.Middleware(sio, app)
59     eventlet.wsgi.server(eventlet.listen(('', 4567)), app)
```

## 2.4 Simulasi Program

Pada program ini, langkah pertama yang harus dilakukan yaitu menjalankan simulator terlebih dahulu untuk merekam situasi jalan dan mendeteksi markah jalan menggunakan kamera kiri, tengah, dan kanan. Selain itu, program merekam respon pengemudi terhadap kondisi jalanan, seperti penyesuaian kecepatan, dan pemutaran setir dengan derajat yang ditentukan menggunakan tombol penggerak pada keyboard. Lalu, mobil merekam keadaan kondisi jalanan dan kondisi mobil dengan menggunakan tombol R untuk Record dalam program simulator selama 3 sampai 5 kali putaran untuk memperoleh hasil data yang lebih akurat.



Setelah program berjalan dan merekam keadaan jalan, foto akan disimpan pada *local drive* berbentuk file .png dan juga berbentuk file .csv (excel) untuk dibaca menggunakan modul pandas.





Setelah terekam, file akan dibaca menggunakan fungsi yang terdapat pada **utlis.py** yang telah dijelaskan pada subbab Algoritma. Lalu, data akan diproses dengan fungsi-fungsi yang telah dibuat dan dijelaskan di atas. Program akan dijalankan menggunakan *local server* untuk menghubungkan algoritma yang telah dibuat dengan aplikasi *autonomous car simulator*.

## BAB III

### DATA

#### 3.1 Analisis Data

Tabel yang dilampirkan dalam folder berbentuk file .csv dengan kolom gambar tengah, gambar kiri, gambar kanan, steering, throttle (pegas), rem (brake), dan kecepatan (speed). Data yang dianalisis berikut merupakan data yang diperoleh dari perekaman mobil saat berjalan selama 5 putaran.

#### 3.2 Deskripsi Data

```
In [1]: 1 import pandas as pd
        2
        3 df = pd.read_csv("driving_log.csv")
        4 df
```

Out[1]:

Data\IMG\left_2022_11_22_10_38_21_021.jpg	C:\Users\USER\OneDrive\Desktop\SimulationData\IMG\right_2022_11_22_10_38_21_021.jpg	0	0.1	0.2	0.09860765
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	0.178853	0	0.018796
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	0.366500	0	0.217015
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	-0.20	0.597974	0	0.653459
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	-0.35	0.779112	0	1.173933
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	1.000000	0	2.028327
...	...	...	...	...	...
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	0.953415	0	30.190300
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	0.774080	0	30.133950
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	0.528550	0	30.085110
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	0.351354	0	30.038180
ers\USER\OneDrive\Desktop\SimulationData\...	C:\Users\USER\OneDrive\Desktop\SimulationData\...	0.00	0.107892	0	29.966870

#### 3.3 Karakteristik Data

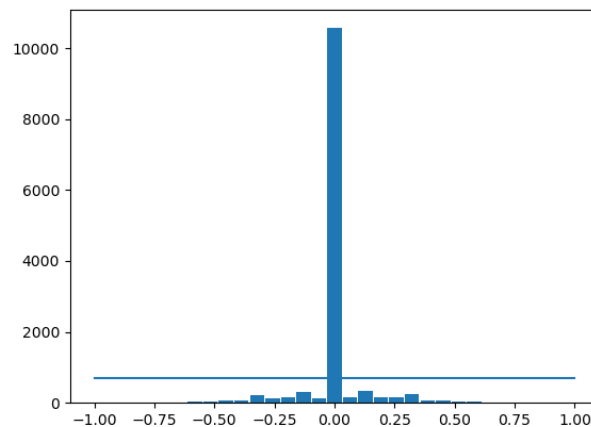
Kolom data yang terdapat pada file “driving\_log.csv” yaitu sebagai berikut:

1. Gambar tengah, yang merupakan gambar yang ditangkap kamera tengah
2. Gambar kiri, yang merupakan gambar yang ditangkap kamera kiri
3. Gambar kanan, yang merupakan gambar yang ditangkap kamera kanan
4. Data ‘steering’ yaitu sudut putaran kemudi mobil
5. Data ‘throttle’ yaitu data seberapa kuat pedal yang ditekan
6. Data ‘brake’, yaitu data pengereman mobil (seperti throttle)
7. Data ‘speed’, yaitu data kecepatan mobil pada waktu tertentu

### 3.4 Visualisasi Data

Bar Chart dari data 'Steering' pada file:

```
1 def balanceData(data, display=True):
2     nBin = 31
3     samplesPerBin = 500
4     hist, bins = np.histogram(data['Steering'], nBin)
5     if display:
6         center = (bins[:-1] + bins[1:]) * 0.5
7         plt.bar(center, hist, width=0.06)
8         plt.plot((np.min(data['Steering']), np.max(data['Steering'])), (samplesPerBin, samplesPerBin))
9         plt.show()
```



Pada data di atas, kita tidak dapat melihat data secara spesifik. Oleh karena itu, kita membuat batasan untuk melihat data lebih spesifik.

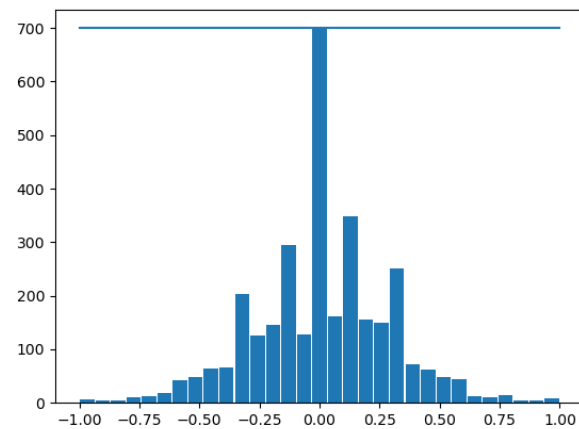
```
1 removeindexList = []
2 for j in range(nBin):
3     binDataList = []
4     for i in range(len(data['Steering'])):
5         if data['Steering'][i] >= bins[j] and data['Steering'][i] <= bins[j + 1]:
6             binDataList.append(i)
7     binDataList = shuffle(binDataList)
8     binDataList = binDataList[samplesPerBin:]
9     removeindexList.extend(binDataList)
10
11 print('Removed Images:', len(removeindexList))
12 data.drop(data.index[removeindexList], inplace=True)
13 print('Remaining Images:', len(data))
```



Lalu:

```
if display:
    hist, _ = np.histogram(data['Steering'], (nBin))
    plt.bar(center, hist, width=0.06)
    plt.plot((np.min(data['Steering']), np.max(data['Steering'])), (samplesPerBin, samplesPerBin))
    plt.show()
```

Maka, berikut merupakan hasil akhir data 'Steering' yang telah diperjelas:



Sample gambar kamera mobil yang telah diproses menggunakan fungsi yang telah dijelaskan:

### 1. Cropping



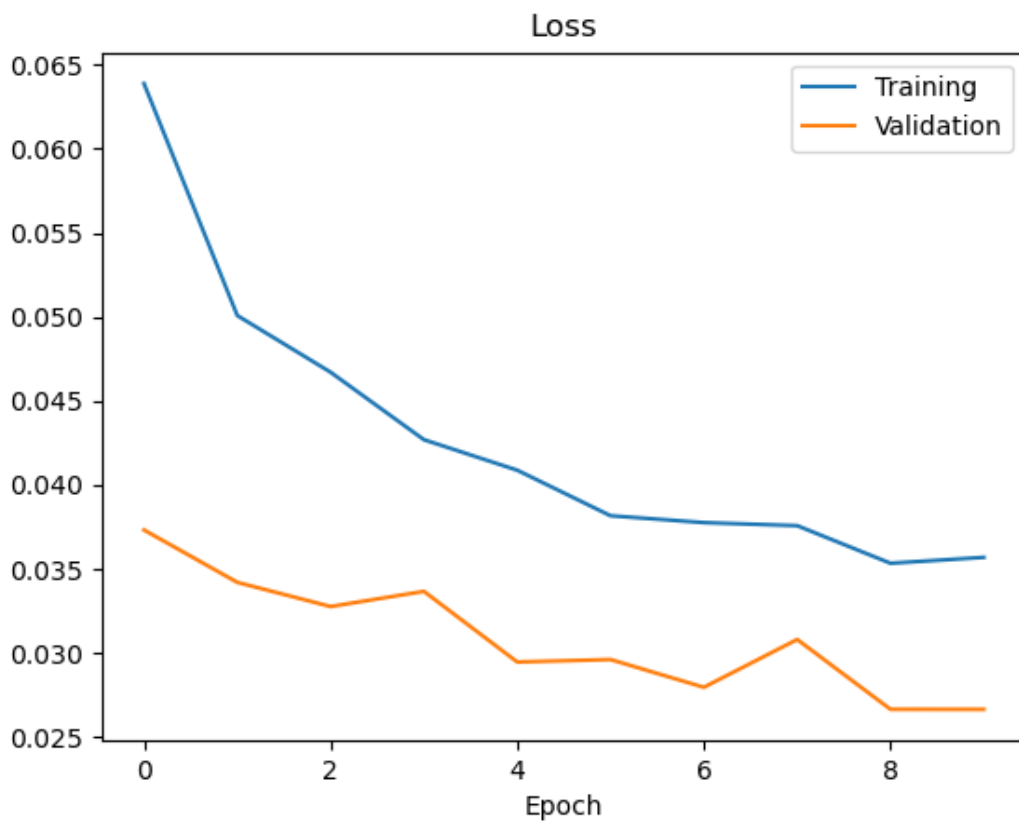
### 2. Zoom



### 3. Blur, cvtColor, dan Resize



Setelah data menjadi seperti di atas, data diproses menggunakan Tensorflow Keras dengan algoritma seperti di atas dan di buat model dengan file 'model.h5'. Berikut merupakan grafik data yang telah dilakukan validasi serta percobaan (training).



## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Kesimpulan yang dapat ditarik dari laporan yang berjudul “Autonomous Car” dengan mengaplikasikannya menggunakan bahasa pemrograman Python 3.0 yaitu, bahwa sistem dapat berjalan sesuai dengan tujuan yang telah ditentukan, walaupun memiliki beberapa keterbatasan seperti belum bisa menambahkan objek yang lainnya. Pembacaan markah jalan, pembacaan pedal mobil, serta sudut setir mobil dengan algoritma yang telah dibuat dapat bekerja dengan sangat baik sehingga mobil dapat bergerak otomatis pada jalannya.

Sehingga, dengan hadirnya *autonomous car* di zaman sekarang ini, tentunya menghadirkan gaya berkendara yang berbeda dibanding transmisi otomatis lainnya. Kita juga tetap bisa merasakan gaya berkendara layaknya mengendarai mobil dengan transmisi manual pada mobil ini.

#### **4.2 Saran**

Berikut merupakan saran yang dapat diberikan untuk memaksimalkan kinerja serta penggunaannya dalam kehidupan sehari-hari, sebagai berikut:

1. Menambahkan objek yang lainnya agar dapat mendeteksi objek selain mobil dan markah jalan.
2. Diharapkan sistem dapat dikembangkan lagi guna pengaplikasian pada mobil sesungguhnya.
3. Perlu ditambahkan sistem pengereman agar mobil dapat menghindari halangan yang ada di depannya.
4. Menggunakan algoritma komputasi yang lebih baik agar bekerja secara lebih optimal.

### 4.3 Pembagian Tugas

NO.	Nama	NIM	Tugas
1.	Rizki Dwi Syarif	16622333	Membuat Draft PowerPoint dan PowerPoint Final
2.	Nasa Amelia Rakhmat	16622348	Membuat Draft Laporan dan Proposal
3.	Carissa Febriana	16622350	Membuat Draft Laporan dan membantu akomodasi diskusi kelompok
4.	Cantikia Ayuningcahya	16622359	Membuat Draft Proposal dan mengakomodasi diskusi kelompok
5.	Mohamad Guntur Sutrisno	16622370	Revisi Draft PowerPoint dan PowerPoint Final serta proposal
6.	Muhammad Abiyyu Hafiansyah	16622371	Membuat Flow Chart dan merevisinya, serta merevisi PowerPoint Final
7.	Reyhan Karisma Ramadhan	16622376	Merancang program serta penanggung jawab atas program yang telah dibuat, mevisualisasikannya, serta revisi Laporan Final