

# **BAB III**

## **PELAKSANAAN**

### **KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Pada pelaksanaan kegiatan kerja magang ini memasuki posisi sebagai *data* visualisasi dan analisis di dalam divisi *AVC* (*automatic vehicle classification*), serta melakukan koordinasi dengan mentor yaitu pak ilham winar yang menjabat sebagai *Ast. Manager*. Proses koordinasi dengan mentor dilakukan setiap hari secara tatap muka di IotLab Jasamarga. Selama proses kerja magang dilakukan, pengerjaan dilakukan secara individu tanpa membentuk tim, dan dipantau langsung oleh pak ilham winar selaku mentor di divisi *AVC* (*AUTOMATIC VEHICLE CLASSIFICATION*) tersebut.

##### **3.1.1 Jobdesk**

Pada program kerja magang yang dilakukan pada *IotLab Jasamarga* sebagai *data* visualisasi, ada beberapa tugas dan tanggung jawab yang diberikan dalam menjalankan program magang di IotLab Jasamarga tersebut yaitu pertama harus melakukan *gathering data* yang bersifat *manual* dari *database* development implementasi yang berupa *data* gambar.

Kemudian melakukan *data* preprosesing yang bersifat *manual* dan otomatis( *using python script* ) yang akan berupa hasil excel *file* yang berisikan dari perbandingan jurnal transaksi GTO(gerbang tol otomatis) dan jurnal transaksi yang di peroleh dari *AI(Artificial Intelligence) AVC (AUTOMATIC VEHICLE CLASSIFICATION)* .  
Selanjutnya bila *data* prosesing selesai akan di lakukannya *data* visualisasi menggunakan tableau untuk membuat visualisasi berupa grafik dan *chart* sebagai pelaporan yang akan di berikan ke bapak Ilham winar selaku *PIC* dalam *team AVC (AUTOMATIC VEHICLE CLASSIFICATION)*.

## 3.2 Uraian Pekerjaan Magang

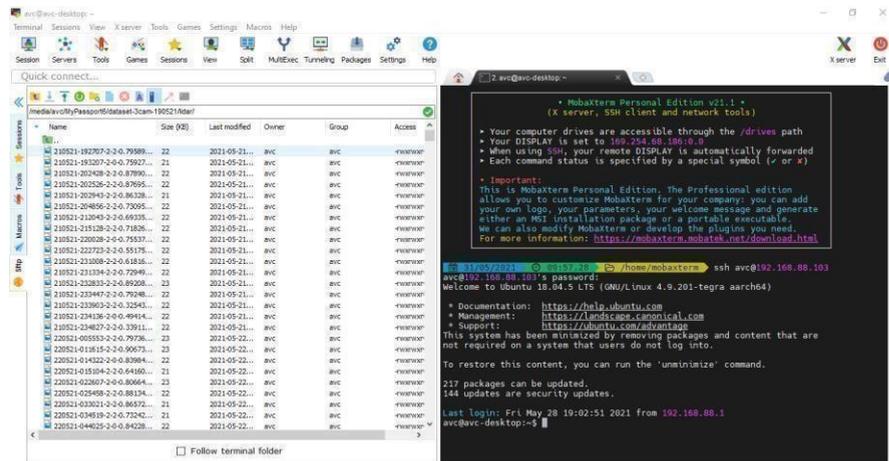
Pada saat melakukan kegiatan pekerjaan magang terdapat pekerjaan yang di lakukan. Berikut adalah merupakan uraian dari pekerjaan yang di lakukan selama program magang di *Iotlab* Jasamarga berlangsung. Pada kegiatan pekerjaan yang di lakukan di tempat magang merupakan semua arahan dari mentor yang mengawasi seluruh kegiatan magang yang di lakukan. Dalam melakukan pekerjaan magang ada beberapa serangkaian pekerjaan yang di lakukan, kegiatan tersebut di lakukan untuk membantu kelancaran dalam melakukan kegiatan magang tersebut. Beberapa tahapan dalam pengerjaan kegiatan magang tersebut akan di jelaskan di dalam sub dari sub bab sebagai berikut.

### 3.2.1 Pengambilan *Data*

Sebelum pengerjaan pembuatan visualisasi *data* pertamanya akan melakukan proses pengambilan *data* terlebih dahulu yang berguna sebagai penopang bahan dari tahapan kegiatan pekerjaan ini yang berupa sebuah foto kendaraan yang sudah memiliki *tag* golongan dari kendaraan tersebut menggunakan *mobaxterm*. Kegiatan pengambilan data ini di lakukan pada seluruh minggu pada kegiatan kerja magang ini, pada minggu pertama dalam kegiatan magang ini proses pengambilan data di lakukan untuk tanggal 14 dan 15 april .

pada minggu kedua dalam kegiatan magang ini proses pengambilan data di lakukan untuk tanggal 16,17,18, dan 19 april ,pada minggu ketiga dalam kegiatan magang ini proses pengambilan data di lakukan untuk tanggal 29 april dan 3 mei, pada minggu keempat dalam kegiatan magang ini proses pengambilan data di lakukan untuk tanggal 4 dan 5 mei, pada minggu kelima dalam kegiatan magang ini proses pengambilan data di lakukan untuk tanggal 11 dan 12 mei, pada minggu keenam dalam kegiatan magang ini proses pengambilan data di lakukan untuk tanggal 14 mei, pada minggu ketujuh dalam kegiatan magang ini proses pengambilan data di lakukan untuk tanggal 15 mei. Tujuan pengambilan data pada ke tujuh minggu tersebut untuk menopang proses kegiatan magang lainnya selama tujuh minggu kegiatan magang ini berlangsung.

Mobaxterm merupakan sebuah *software* untuk komputasi dengan jarak jauh dalam satu aplikasi. Mobaxterm menyediakan fungsi yang dapat di sesuaikan untuk *administrator programmer,webmaster, IT straff* dalam menyediakan semua alat jaringan jarak jauh seperti SSH,X11,RDP,VNC,FTP,MOSH) dan perintah unix (back,is,cat,sed,grep,awk,rsync). [2]



**Gambar 3.1 MobaXterm**

Pada gambar 3.1 menampilkan sebuah *UI* dari mobaxterm itu sendiri yang dimana nanti akan mencari directory di dalam mobaxterm yang telah terkoneksi oleh *json* yang menjadi mini komputer dalam device AVC(*automatic vehicle classification*) yang di gunakan juga sebagai monitoring apakah telah terjadi error atau semacamnya. Dan juga pengambilan foto atau gambar kendaraan dapat di lakukan dari jarak jauh tidak perlu lagi kelapangan.

### 3.2.2 Pengolahan *Data*

Dalam *fase* ini setelah *data* di ambil dari mobaxterm yang berupa gambar- gambar yang tertangkap sensor oleh lidar di gerbang tol lalu akan di bersihkan *data-data* yang kembar (*redundancy*) untuk di hapus secara manual atau juga bisa di cleansing dan di kelompokkan secara otomatis menggunakan *script python* yang telah di sediakan oleh mentor pada saat kegiatan pengerjaan kerja magang.

Kegiatan pembersihan data ini di lakukan pada setiap minggunya demi mendapatkan sumber data yang bersih dan bagus untuk di kelolah dalam tahap selanjutnya. pada minggu pertama dalam kegiatan magang ini proses pembersihan data di lakukan untuk tanggal 14 dan 15 april , pada minggu kedua dalam kegiatan magang ini proses pembersihan data di lakukan untuk tanggal 16,17,18, dan 19 april , pada minggu ketiga dalam kegiatan magang ini proses pembersihan data di lakukan untuk tanggal 29 april , 3 dan 4 mei.

pada minggu keempat dalam kegiatan magang ini proses pemberisihan data di lakukan untuk tanggal 5 mei, pada minggu kelima dalam kegiatan magang ini proses pembersihan data di lakukan untuk tanggal 11 dan 12 mei, pada minggu keenam dalam kegiatan magang ini proses pembersihan data di lakukan untuk tanggal 14 mei, pada minggu ketujuh dalam kegiatan magang ini proses pembersihan data di lakukan untuk tanggal 15 mei. Contoh dari *code pyhton* proses *cleansing* dan pengelompokan sebagai berikut:

```
E:\Internship\180521\AUDIT.py
temp.py x AUDIT.py
1 import cv2
2 from queue import Empty, Queue
3 import os
4 import glob
5 import time
6 import logging
7
8 #tanggal-jam-golongan-confidence-camberapa
9 # format tanggal-ddmmyy
10 # format jam-hhmmss
11 # folder 4 sama 5 ada cam 3
12 # folder 0 sama 3 ada cam 2 aja
13 # start ambil tanggal 4 april
14 logging.basicConfig(filename="Log-pindah-file.log", level=logging.INFO)
15
16 q = Queue(1)
17
18 os.chdir("E:/Internship/180521/golongan/5/") #change directory
19 IS_ABOVE_3 = True #ganti ke false kalo golongan yang dipilih <-| 3
20
21 CMD = os.getcwd()
22 VIDPATH = os.path.join(CMD)
23 LASTFILEPATH = os.path.join("Lastvidfile.txt")
24
25 if not os.path.exists(LASTFILEPATH):
26     open(LASTFILEPATH, 'w').close()
27
28 lastvidfile = open(LASTFILEPATH, 'r')
29 idx = 0
30
31 if __name__ == '__main__':
32     files = sorted(glob.glob(VIDPATH + "/*.jpg"))
33     # print(files)
34     total_files=len(files)
35     total_files=int(total_files)
36     for i in range(total_files):
37         try:
38             lastfile = lastvidfile.read()
39
40
41
42
43     if lastfile != '':
44         idx = files.index(lastfile)
45
46         if IS_ABOVE_3 == False:
47             if idx + 2 <= len(files) - 1:
48                 idx = idx + 2
49             else:
50                 idx = 0
51                 lastvidfile.close()
52                 idx=0
53
54         if IS_ABOVE_3 == True:
55             if idx + 3 <= len(files) - 2:
56                 idx = idx + 3
57             else:
58                 idx = 0
59                 lastvidfile.close()
60                 idx=0
61         except:
62             pass
63
64         if IS_ABOVE_3 == False:
65             idx=idx+2
66
67         if IS_ABOVE_3 == True:
68             idx=idx+3
69
70     name_file=os.path.basename(files[idx].replace('-cam1', ''))
71     name_file_cam1=os.path.basename(files[idx])
72     name_file_cam2=os.path.basename(files[idx].replace('cam1', 'cam2'))
73
74     if IS_ABOVE_3 == True:
75         name_file_cam3=os.path.basename(files[idx].replace('cam1', 'cam3'))
76
77     print("")
78     print(idx)
79     print(name_file)
80     print(name_file_cam1)
81     print(name_file_cam2)
```

Gambar 3.2 Code Phyton 1

Pada gambar 3.2 *python* yang terdapat di atas menjelaskan untuk melakukan pemusahaan untuk *file* golongan 0 yang akan nanti di saat di *run* akan meminta untuk memilih golongan kendaraan yang akan di mulai untuk di periksa contoh di atas itu untuk meminta mau memulai mengecek golongan berapa kalau golongan nya sesuai dan juga *directory filenya* sesuai akan *print* 3 foto sesuai golongan tersebut.

```
E:\Internship\180521\AUDIT.py
temp.py x AUDIT.py x
123     lastVidFile.close()
124     print("Logged. File in Category 1")
125     logging.info(name_file_cam1 + " in Folder of Category 1")
126     logging.info(name_file_cam2 + " in Folder of Category 1")
127
128     if IS_ABOVE_3 == True:
129         logging.info(name_file_cam3 + " in Folder of Category 1")
130
131     elif k==ord('0'):
132         dir_test_cam1="E:/Internship/180521/result/0/"+name_file_cam1
133         dir_test_cam2="E:/Internship/180521/result/0/"+name_file_cam2
134
135         if IS_ABOVE_3 == True:
136             dir_test_cam3="E:/Internship/180521/result/0/"+name_file_cam3
137
138             os.rename(files[idx], dir_test_cam1)
139             os.rename(files[idx].replace('cam1', 'cam2'), dir_test_cam2)
140
141             if IS_ABOVE_3 == True:
142                 os.rename(files[idx].replace('cam1', 'cam3'), dir_test_cam3)
143
144             lastVidFile = open(LASTFILEPATH, 'w')
145             lastVidFile.write(files[idx])
146             lastVidFile.close()
147             print("Logged. File in Category 0")
148             logging.info(name_file_cam1 + " in Folder of Category 0")
149             logging.info(name_file_cam2 + " in Folder of Category 0")
150
151             if IS_ABOVE_3 == True:
152                 logging.info(name_file_cam3 + " in Folder of Category 0")
153
154     elif k==ord('2'):
155         dir_test_cam1="E:/Internship/180521/result/2/"+name_file_cam1
156         dir_test_cam2="E:/Internship/180521/result/2/"+name_file_cam2
157
158         if IS_ABOVE_3 == True:
159             dir_test_cam3="E:/Internship/180521/result/2/"+name_file_cam3
160
161             os.rename(files[idx], dir_test_cam1)
162             os.rename(files[idx].replace('cam1', 'cam2'), dir_test_cam2)
163
E:\Internship\180521\AUDIT.py
temp.py x AUDIT.py x
82
83     if IS_ABOVE_3 == True:
84         print(name_file_cam3)
85
86     img1 = cv2.imread(files[idx])
87     img2 = cv2.imread(files[idx].replace('cam1', 'cam2'))
88
89     if IS_ABOVE_3 == True:
90         img3 = cv2.imread(files[idx].replace('cam1', 'cam3')) #kalo gol 4 da
91
92     cv2.imshow("kamera1", img1)
93     cv2.imshow("kamera2", img2)
94
95     if IS_ABOVE_3 == True:
96         cv2.imshow("kamera3", img3)
97
98     k = cv2.waitKey(0)
99
100
101     if k == 27:
102         lastVidFile = open(LASTFILEPATH, 'w')
103         lastVidFile.write(files[idx])
104         lastVidFile.close()
105         print("ok saved")
106         break
107
108     elif k==ord('1'): #1 next kalo bis
109         dir_test_cam1="E:/Internship/180521/result/1/"+name_file_cam1
110         dir_test_cam2="E:/Internship/180521/result/1/"+name_file_cam2
111
112         if IS_ABOVE_3 == True:
113             dir_test_cam3="E:/Internship/180521/result/1/"+name_file_cam3
114
115             os.rename(files[idx], dir_test_cam1)
116             os.rename(files[idx].replace('cam1', 'cam2'), dir_test_cam2)
117
118             if IS_ABOVE_3 == True:
119                 os.rename(files[idx].replace('cam1', 'cam3'), dir_test_cam3)
120
121         lastVidFile = open(LASTFILEPATH, 'w')
122         lastVidFile.write(files[idx])
```

Gambar 3.3 *Phyton code 2*

Pada gambar 3.2 *python* yang terdapat di atas menjelaskan untuk melakukan pemusahaan untuk *file* golongan 1,2,3 yang akan nanti di saat di run akan meminta untuk memilih golongan kendaraan yang akan di mulai untuk di periksa contoh di atas itu untuk meminta mau memulai mengecek golongan berapa kalau golongan nya sesuai dan juga *directory* *filenya* sesuai akan print 3 foto sesuai golongan tersebut .

```
E:\Internship\180521\AUDIT.py
temp.py x AUDIT.py x
163
164     if IS_ABOVE_3 == True:
165         os.rename(files[idx].replace('cam1', 'cam3'), dir_test_cam3)
166
167     lastVidFile = open(LASTFILEPATH, 'w')
168     lastVidFile.write(files[idx])
169     lastVidFile.close()
170     print("Logged. File in Category 2")
171     logging.info(name_file_cam1 + " in Folder of Category 2")
172     logging.info(name_file_cam2 + " in Folder of Category 2")
173
174     if IS_ABOVE_3 == True:
175         logging.info(name_file_cam3 + " in Folder of Category 2")
176
177     elif k==ord('3'):
178         dir_test_cam1="E:/Internship/180521/result/3/"+name_file_cam1
179         dir_test_cam2="E:/Internship/180521/result/3/"+name_file_cam2
180
181     if IS_ABOVE_3 == True:
182         dir_test_cam3="E:/Internship/180521/result/3/"+name_file_cam3
183
184     os.rename(files[idx], dir_test_cam1)
185     os.rename(files[idx].replace('cam1', 'cam2'), dir_test_cam2)
186
187     if IS_ABOVE_3 == True:
188         os.rename(files[idx].replace('cam1', 'cam3'), dir_test_cam3)
189
190     lastVidFile = open(LASTFILEPATH, 'w')
191     lastVidFile.write(files[idx])
192     lastVidFile.close()
193     print("Logged. File in Category 3")
194     logging.info(name_file_cam1 + " in Folder of Category 3")
195     logging.info(name_file_cam2 + " in Folder of Category 3")
196
197     if IS_ABOVE_3 == True:
198         logging.info(name_file_cam3 + " in Folder of Category 3")
199
200     elif k==ord('4'):
201         dir_test_cam1="E:/Internship/180521/result/4/"+name_file_cam1
202         dir_test_cam2="E:/Internship/180521/result/4/"+name_file_cam2
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242     if IS_ABOVE_3 == True:
243         logging.info(name_file_cam3 + " in Folder of Category 5")
244
245     elif k==ord('i'):
246         dir_test_cam1="E:/Internship/180521/result/invalid/"+name_file_cam1
247         dir_test_cam2="E:/Internship/180521/result/invalid/"+name_file_cam2
248
249     if IS_ABOVE_3 == True:
250         dir_test_cam3="E:/Internship/180521/result/invalid/"+name_file_c
251
252     os.rename(files[idx], dir_test_cam1)
253     os.rename(files[idx].replace('cam1', 'cam2'), dir_test_cam2)
254
255     if IS_ABOVE_3 == True:
256         os.rename(files[idx].replace('cam1', 'cam3'), dir_test_cam3)
257
258     lastVidFile = open(LASTFILEPATH, 'w')
259     lastVidFile.write(files[idx])
260     lastVidFile.close()
261     print("Logged. File is classified as Invalid")
262     logging.info(name_file_cam1 + " dumped into Invalid Folder")
263     logging.info(name_file_cam2 + " dumped into Invalid Folder")
264
265     if IS_ABOVE_3 == True:
266         logging.info(name_file_cam3 + " dumped into Invalid Folder")
267
268     else:
269         break
270
271     cv2.destroyAllWindows()
272
```

Gambar 3.4 *Python 3*

Pada gambar 3.4 phyton yang terdapat di atas menjelaskan untuk melakukan pemusahaan untuk *file* golongan 4, 5 yang akan nanti di saat di *run* akan meminta untuk memilih golongan kendaraan yang akan di mulai untuk di periksa contoh di atas itu untuk meminta mau memulai mengecek golongan berapa kalau golongan nya sesuai dan juga *directory filenya* sesuai akan *print* 3 foto sesuai golongan tersebut yang berapa di awal foto

```
E:\Internship\180521\AUDIT.py
temp.py x AUDIT.py
203
204     if IS_ABOVE_3 == True:
205         dir_test_cam3="E:/Internship/180521/result/4/"+name_file_cam3
206
207         os.rename(files[idx], dir_test_cam1)
208         os.rename(files[idx].replace('cam1', 'cam2'), dir_test_cam2)
209
210     if IS_ABOVE_3 == True:
211         os.rename(files[idx].replace('cam1', 'cam3'), dir_test_cam3)
212
213     lastVidFile = open(LASTFILEPATH, 'w')
214     lastVidFile.write(files[idx])
215     lastVidFile.close()
216     print("Logged: File in Category 4")
217     logging.info(name_file_cam1 + " in Folder of Category 4")
218     logging.info(name_file_cam2 + " in Folder of Category 4")
219
220     if IS_ABOVE_3 == True:
221         logging.info(name_file_cam3 + " in Folder of Category 4")
222
223     elif k==ord('5'):
224         dir_test_cam1="E:/Internship/180521/result/5/"+name_file_cam1
225         dir_test_cam2="E:/Internship/180521/result/5/"+name_file_cam2
226
227     if IS_ABOVE_3 == True:
228         dir_test_cam3="E:/Internship/180521/result/5/"+name_file_cam3
229
230     os.rename(files[idx], dir_test_cam1)
231     os.rename(files[idx].replace('cam1', 'cam2'), dir_test_cam2)
232
233     if IS_ABOVE_3 == True:
234         os.rename(files[idx].replace('cam1', 'cam3'), dir_test_cam3)
235
236     lastVidFile = open(LASTFILEPATH, 'w')
237     lastVidFile.write(files[idx])
238     lastVidFile.close()
239     print("Logged: File in Category 5")
240     logging.info(name_file_cam1 + " in Folder of Category 5")
241     logging.info(name_file_cam2 + " in Folder of Category 5")
242
```

Gambar 3.5 Code Python

Kesimpulan *code* pada gambar 3.5 merupakan codingan yang di gunakan untuk mengecek apakah *AI AVC (AUTOMATIC VEHICLE CLASSIFICATION)* sudah *valid* atau belum untuk penentuan golongannya. Cara kerja dari codingan nya memiliki sebuah syarat yaitu kita harus membuat *folder result* yang dimana hasil semua foto yang di *check* akan masuk ke dalam *folder result* yang berisikan *folder* golongan kendaraan 0-5 dan juga *folder invalid*, *folder invalid* ini berisikan foto- foto yang rusak tidak bisa di pakai karna *preprosesing* yang dilakukan untuk *data* visualisasi juga di gunakan untuk membantu mengumpulkan *dataset* dan *data training* untuk *improvement system AI (Artificial Intelligence) AVC (AUTOMATIC VEHICLE CLASSIFICATION)* itu sendiri.

Setelah di lakukan pembersihan tadi selanjutnya akan di kelolah yang dimana dari *data- data* foto yang sudah di kelompokkan dan sudah di bersihkan akan di bandingkan dari jurnal transaksi yang di *record* oleh *AI(Artificial Intelligence) AVC (AUTOMATIC VEHICLE CLASSIFICATION)* dengan jurnal transaksi dari GTO(gerbang tol otomatis) yang merupakan excel yang dimana hasil akhir dari excel tersebut merupakan sebuah akurasi *AI(Artificial Intelligence)* dalam menggolongkan kendaraan secara sistem dari per *shift* pada hari . Contoh dari gambar excel oleh *data* yang di kelolah sebagai berikut:

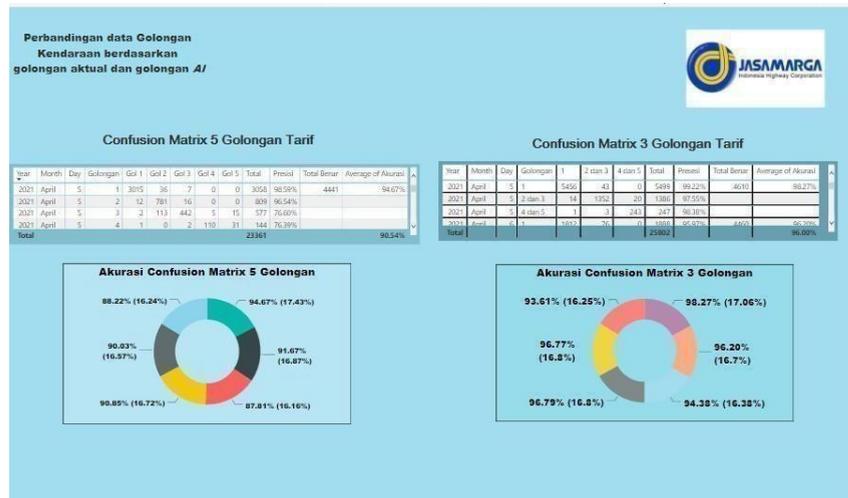
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
921	918	04/05/2021	12:42:34	1	1	1	1	1	1	1	1	1	1	1	1	1
922	920	04/05/2021	12:42:37	1	1	1	1	1	1	1	1	1	1	1	1	1
923	921	04/05/2021	12:42:39	1	1	1	1	1	1	1	1	1	1	1	1	1
924	922	04/05/2021	12:42:41	1	1	1	1	1	1	1	1	1	1	1	1	1
925	923	04/05/2021	12:42:43	1	1	1	1	1	1	1	1	1	1	1	1	1
926	924	04/05/2021	12:42:45	1	1	1	1	1	1	1	1	1	1	1	1	1
927	925	04/05/2021	12:42:47	1	1	1	1	1	1	1	1	1	1	1	1	1
928	926	04/05/2021	12:42:49	1	1	1	1	1	1	1	1	1	1	1	1	1
929	927	04/05/2021	12:42:51	1	1	1	1	1	1	1	1	1	1	1	1	1
930	928	04/05/2021	12:42:53	1	1	1	1	1	1	1	1	1	1	1	1	1
931	929	04/05/2021	12:42:55	1	1	1	1	1	1	1	1	1	1	1	1	1
932	930	04/05/2021	12:42:57	1	1	1	1	1	1	1	1	1	1	1	1	1
933	931	04/05/2021	12:42:59	1	1	1	1	1	1	1	1	1	1	1	1	1
934	932	04/05/2021	12:43:01	1	1	1	1	1	1	1	1	1	1	1	1	1
935	933	04/05/2021	12:43:03	1	1	1	1	1	1	1	1	1	1	1	1	1
936	934	04/05/2021	12:43:05	1	1	1	1	1	1	1	1	1	1	1	1	1
937	935	04/05/2021	12:43:07	1	1	1	1	1	1	1	1	1	1	1	1	1
938	936	04/05/2021	12:43:09	1	1	1	1	1	1	1	1	1	1	1	1	1
939	937	04/05/2021	12:43:11	1	1	1	1	1	1	1	1	1	1	1	1	1
940	938	04/05/2021	12:43:13	1	1	1	1	1	1	1	1	1	1	1	1	1
941	939	04/05/2021	12:43:15	1	1	1	1	1	1	1	1	1	1	1	1	1
942	940	04/05/2021	12:43:17	1	1	1	1	1	1	1	1	1	1	1	1	1
943	941	04/05/2021	12:43:19	1	1	1	1	1	1	1	1	1	1	1	1	1
944	942	04/05/2021	12:43:21	1	1	1	1	1	1	1	1	1	1	1	1	1
945	943	04/05/2021	12:43:23	1	1	1	1	1	1	1	1	1	1	1	1	1

Gambar 3.6 Contoh *Data Excel*

Gambar 3.6 merupakan *data excel* di atas merupakan gambaran dari perbandingan akurasi yang dilakukan oleh pultol( orang di gto yang menentukan golongan per transaksi secara *manual*) dan akurasi yang dilakukan oleh *system AI(Artificial Intelligence) AVC (AUTOMATIC VEHICLE CLASSIFICATION)* yang menjadi dasar dalam pembuatan *Data visualisasi*.

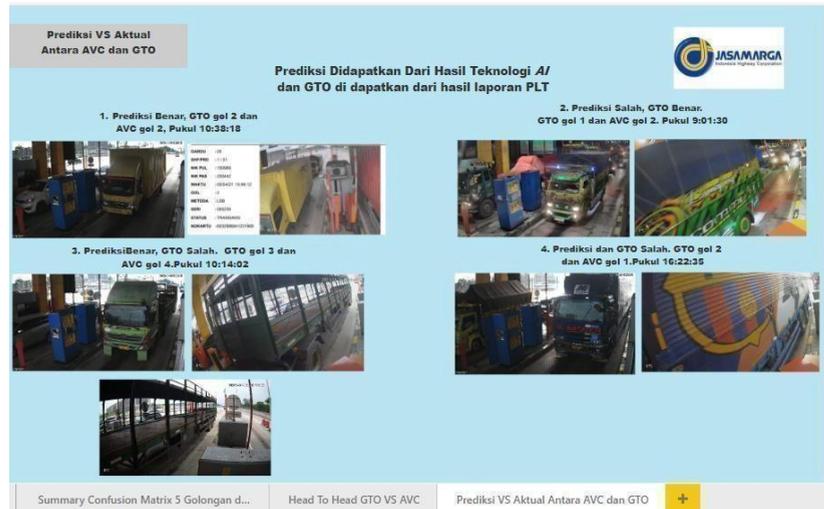
### **3.2.3 Pembuatan *Datavisualisasi***

Pada tahap pembuatan data visualisasi dalam kegiatan kerja magang ini di lakukan pada minggu-minggu pada kegiatan kerja magang yang langsung di berikan tugasnya oleh mentor pada kegiatan magang ini. Rincian kegiatan mingguan kegiatan magang ini adalah pada minggu pertama dalam kegiatan magang ini proses pembuatan data visualisasi data di lakukan untuk confusion matrix yang telah di sediakan oleh mentor pada kegiatan kerja magang ini, pada minggu ketiga dalam kegiatan magang ini proses pembuatan *data* visualisasi yang di lakukan untuk tanggal 29 april dan 3 mei 4 mei ,pada minggu kelima dalam kegiatan magang ini proses pembuatan *data* visualisasi yang di lakukan untuk tanggal 11 dan 12 mei, pada minggu keenam dalam kegiatan magang ini proses pembuatan *data* visualisasi yang di lakukan untuk tanggal 14 mei, pada minggu ketujuh dalam kegiatan magang ini proses pembuatan *data* visualisasi yang di lakukan untuk tanggal 15 mei. hasil pembuatan pembuatan hasil kerja magang pada proses ini sebagai berikut.



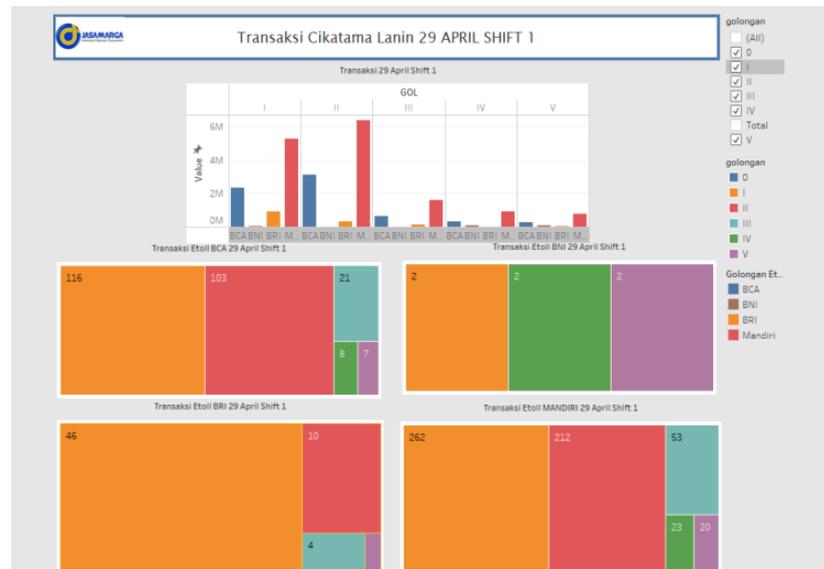
Gambar 3.7 Dashboard

Gambar 3.7 merupakan dashboard di atas menjelaskan tentang *result confusion matrix* dari 5 golongan dan yang sudah di bagi lagi menjadi 3 golongan yang menampilkan *data* akurasi dari *AI(Artificial Intelligence)* saat menentukan golongan kendaraan tersebut.



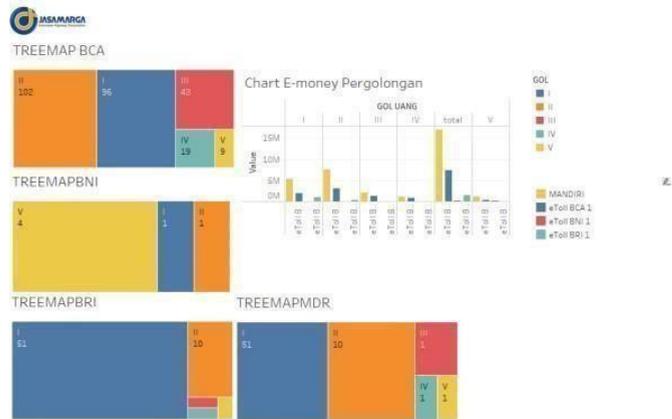
Gambar 3.8 *Dashboard*

Gambar 3.8 merupakan *dashboard*, *dashboard* ini menjelaskan contoh dari perbandingangolongan kendaraan yang di prediksi antara *AI (Artificial Intelligence) AVC (AUTOMATIC VEHICLE CLASSIFICATION)* dan prediksi yang di lakukan manual oleh pihak GTO(Gerbang tol otomatis).



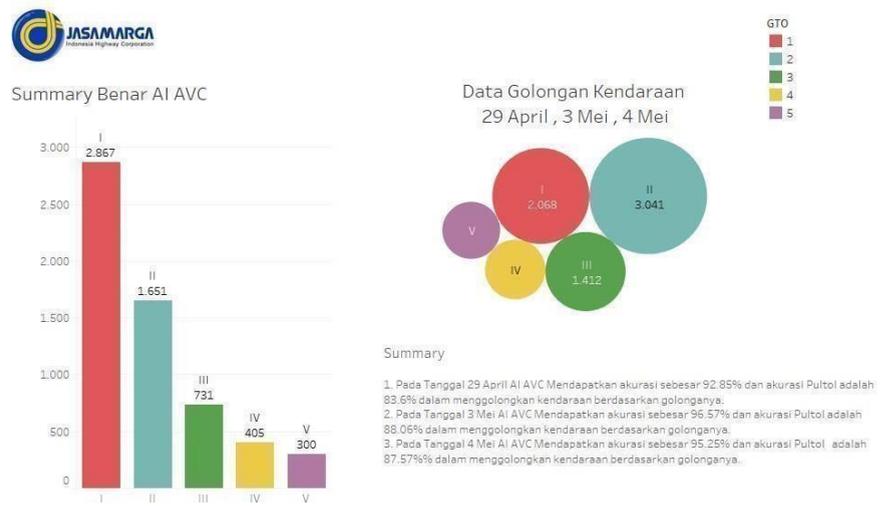
Gambar 3.9 Dashboard 29 shift 1

Gambar 3.9 merupakan *dashboard 29 shift 1*. Pada *dashboard* di atas menjelaskan sebuah transaksi pada tol yang di golongkan dari 4 jenis pembayaran *e-money* yang berbeda yang di akhiri dengan presentare *bar chart* profit dari ke empat jenis *e-money* pada tanggal 29april 2021 pada *shift 1* di gardu cikampek utama.



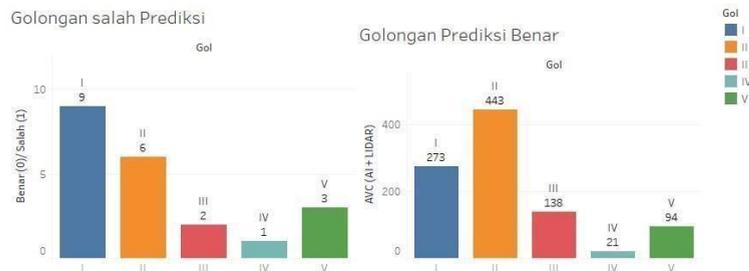
Gambar 3.10 *Dashboard 29 shift 2*

Gambar 3.10 merupakan *dashboard 29 shift 2*. Pada *dashboard* di atas menjelaskan sebuah transaksi pada tol yang di golongkan dari 4 jenis pembayaran *e-money* yang berbeda yang di akhiri dengan presentare *bar chart* profit dari ke empat jenis *e-money* pada tanggal 29april 2021 pada *shift 2* di gardu cikampek utama.



Gambar 3.11 Dashboard akurasi 1

Pada gambar 3.11 menjelaskan tentang 2 sheet yaitu *sheet* pertama menjelaskan tentang banyaknya total *data* yang benar di prediksi oleh *AI (Artificial Intelligence) AVC (AUTOMATIC VEHICLE CLASSIFICATION)* yang di bagi per golongan yang dapat di lihat di *bar chart* pada gambardi atas, lalu yang kedua menjelaskan *data*.



Gambar 3.12 *Dashboard* 11 Mei *Shift* 1

Pada gambar 3.12 menjelaskan pada hasil *AI (Artificial Intelligence) AVC (automatic vehicle classification)* memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI(Artificial Intelligence)* dalam memprediksi golongan kendaraan dengan benar.



**AKURASI AVC**  
**96,58%**

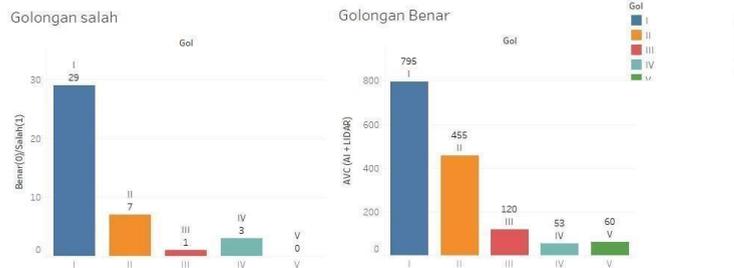


Gambar 3.13 Dashboard 11 Mei Shift 2

Pada gambar 3.13 menjelaskan pada hasil AI (*Artificial Intelligence*) AVC (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan *bar chart* golongan yang benar di prediksi dan juga *summary* total akurasi AI(*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.



AKURASI AVC  
96.14%



Gambar 3.14 *Dashboard 11 Mei Shift 3*

Pada gambar 3.14 menjelaskan pada hasil AI (*Artificial Intelligence*) AVC (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan *bar chart* golongan yang benar di prediksi dan juga *summary* total akurasi AI (*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.



Gambar 3.15 Dashboard 12 Mei Shift 1

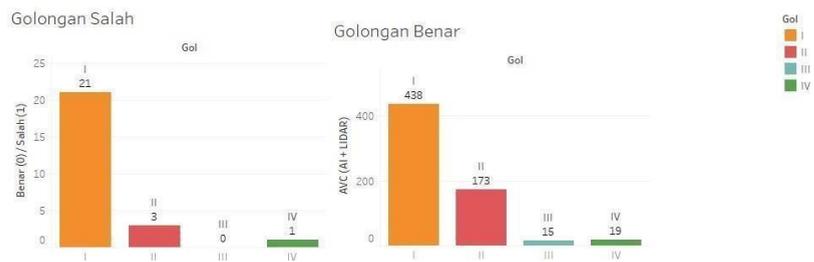
Pada gambar 3.15 menjelaskan pada hasil AI (*Artificial Intelligence*) AVC (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar AI golongan yang benar di prediksi dan juga *summary* total akurasi AI (*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.



Gambar 3.16 Dashboard 12 Mei Shift 2

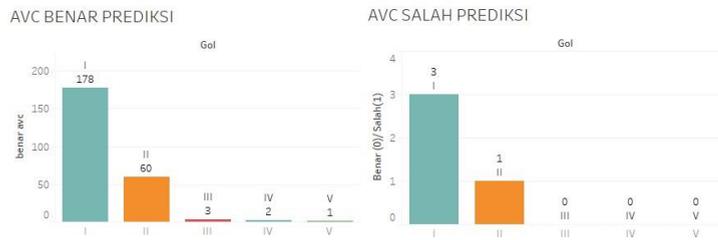
Pada gambar 3.16 menjelaskan pada hasil *AI (Artificial Intelligence) AVC (automatic vehicle classification)* memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI(Artificial Intelligence)* dalam memprediksi golongan kendaraan dengan benar.

**AKURASI AVC**  
**95,18%**

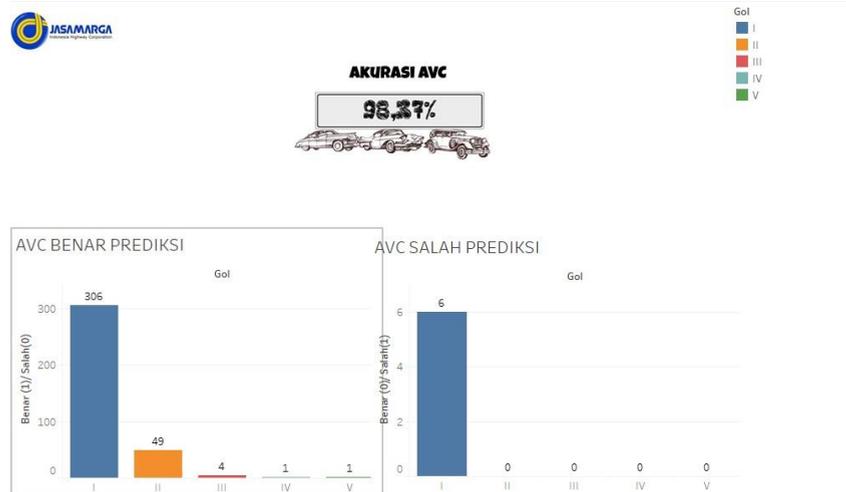
Gambar 3.17 Dashboard 12 Mei Shift 3

Pada gambar 3.17 menjelaskan pada hasil *AI (Artificial Intelligence) AVC (automatic vehicle classification)* memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI(Artificial Intelligence)* dalam memprediksi golongan kendaraan dengan benar.



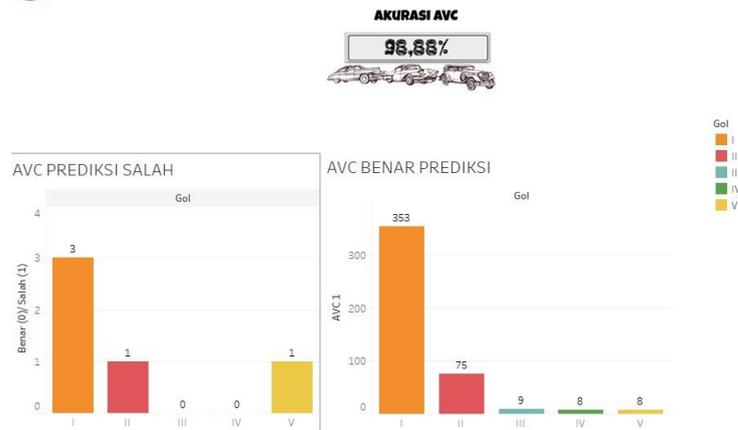
Gambar 3.18 *dashboard* 14 mei shift 1

Pada gambar 3.18 menjelaskan pada hasil *AI (Artificial Intelligence) AVC (automatic vehicle classification)* memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI(Artificial Intelligence)* dalam memprediksi golongan kendaraan dengan benar.



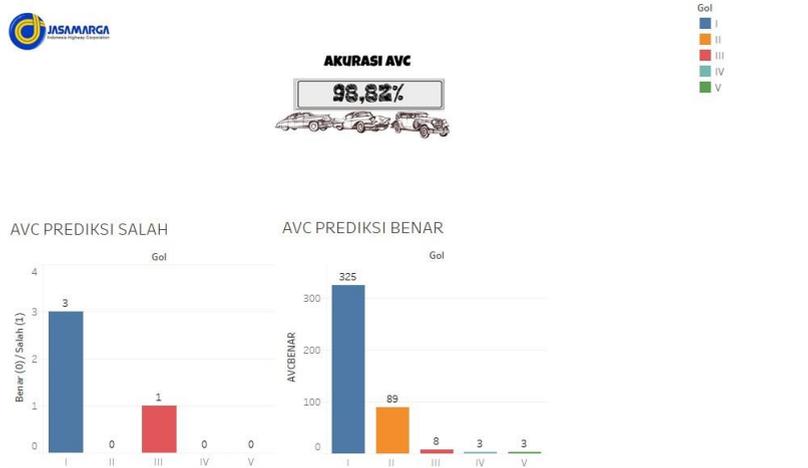
Gambar 3.19 *dashboard* 14 mei shift 2

Pada gambar 3.19 menjelaskan pada hasil *AI* (*Artificial Intelligence*) *AVC* (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI* (*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.



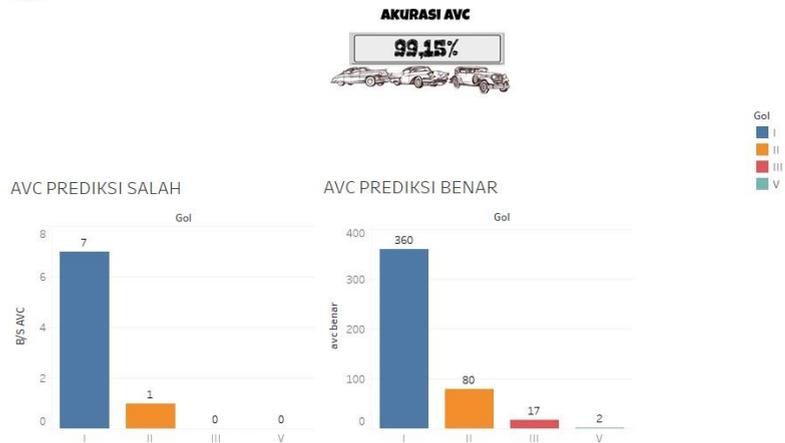
Gambar 3.20 *dashboard* 14 mei shift 3

Pada gambar 3.20 menjelaskan pada hasil *AI* (*Artificial Intelligence*) AVC (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI* (*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.



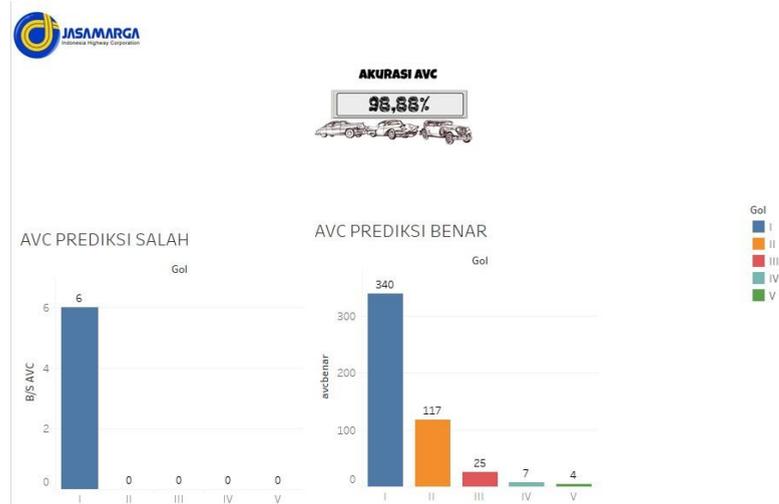
Gambar 3.21 *dashboard* 15 mei shift 1

Pada gambar 3.21 menjelaskan pada hasil *AI* (*Artificial Intelligence*) *AVC* (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI* (*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.



Gambar 3.22 dashboard 15 mei shift 2

Pada gambar 3.22 menjelaskan pada hasil *AI* (*Artificial Intelligence*) AVC (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI* (*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.



Gambar 3.23 dashboard 15 mei shift 3

Pada gambar 3.23 menjelaskan pada hasil *AI* (*Artificial Intelligence*) *AVC* (*automatic vehicle classification*) memprediksi golongan kendaraanya yang bisa di jelaskan berisi *chart* hasil salah prediksi dan perbandingan dengan bar *chart* golongan yang benar di prediksi dan juga *summary* total akurasi *AI* (*Artificial Intelligence*) dalam memprediksi golongan kendaraan dengan benar.

### 3.3 Kendala Yang Dihadapi

Pada proses kerja magang mahasiswa mengalami sebuah kendala yang ada pada saat melakukan proses kerja magang itu sendiri. Kendala tersebut merupakan sebagai berikut:

1. Pada saat proses anotasi *data* dan *data* processing yang di olah menjadi visualisasi *data* terjadi kendala yang dimana rumitnya *data* untuk di kelolah ke tableau karna banyak *data* yang mising dan *null* jadi proses cleansingnya membutuhkan waktu yang lama
2. Tidak ada *expertise* yang dapat membantu dan berdiskusi secara maksimal di saat mengalami problem saat pengolahan *data*.

### 3.3.1 Solusi Dari Kendala yang dihadapi

Solusi dari kendala yang di hadapi selama proses pengerjaan magang sebagai berikut:

1. Solusi dari kendala yang pertama adalah dengan mengecek ulang jurnal transaksi yang hilang dengan jurnal GTO sebagai komperasi *data* yang missing tersebut agar mengurangi null yang berdampakburuk saat di proses ke tableau.
2. Solusi kedua dengan mencari informasi di internet yang berupa *video* edukasi, jurnal penelitian dan buku tentang visualisasi *data* itu sendiri.