

**LAPORAN AKHIR**  
**PENGOLAHAN CITRA VIDEO**



Aiza Fuji Sari

5024211019

**Dosen Pengampu:**

Dr. Eko Mulyanto Yuniarno, S.T., M.T.

**DEPARTEMEN TEKNIK KOMPUTER**  
**FAKULTAS TEKNOLOGI ELEKTRO INFORMATIKA CERDAS**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA**  
**2023/2024**

# 1 Deskripsi Tugas

Pada tugas yang diberikan adalah melakukan deteksi kartu dan membuat game dari hasil deteksi kartu tersebut. Dalam tugas ini saya menggunakan kartu remi. Untuk deteksi kartu dengan menggunakan metode CNN.

# 2 Penjelasan Game

Game yang saya buat merupakan game kartu dengan besar - besaran nilai kartu. Jadi pada game ini kartu player dan computer akan dibandingkan nilainya, yang memiliki nilai kartu paling besar atau tinggi maka scorenya akan ditambah satu. Pada game ini dapat mengeluarkan sampai maksimal 6 kartu dan tiap kartu yang dikeluarkan akan di bandingkan nilainya. Score yang dimiliki setelah membandingkan semua kartu akan menjadi pemenang.

# 3 Source Code

## 3.1 Modul CNN

```
1 #####
2 # Progrm Ini dijadikan modul dengan nama
3 # ModulKlasifikasiCitraCNN.py
4 #####
5
6 import os
7 from keras.models import load_model
8 import cv2
9 import numpy as np
10 from keras.layers import Input, Dense
11 from keras.layers import Conv2D, MaxPooling2D, Flatten
12 from keras.models import Model
13 import matplotlib.pyplot as plt
14 from datetime import datetime
15 from numpy import expand_dims
16 from keras.utils import load_img
17 from keras.utils import img_to_array
18 from keras.preprocessing.image import ImageDataGenerator
19 from matplotlib import pyplot
20
21 # load the image
22 def LoadCitraTraining(sDir,LabelKelas):
23     JumlahKelas=len(LabelKelas)
24     TargetKelas = np.eye(JumlahKelas)
25     # Menyiapkan variabel list untuk data menampung citra
26     # dan data target
27     X=[]#Menampung Data Citra
28     T=[]#Menampung Target
29     for i in range(len(LabelKelas)):
30         #Membaca file citra di setiap direktori data set
```

```

31     DirKelas = os.path.join(sDir, LabelKelas[i])
32     files = os.listdir(DirKelas)
33     for f in files:
34         ff=f.lower()
35         print(f)
36         #memilih citra dengan extensi jpg,jpeg,dan png
37         if (ff.endswith('.jpg')|ff.endswith('.jpeg')|ff.
38 endswith('.png')):
39             NmFile = os.path.join(DirKelas,f)
40             img= np.double(cv2.imread(NmFile,1))
41             img=cv2.resize(img,(128,128))
42             img= np.asarray(img)/255
43             img=img.astype('float32')
44             #Menambahkan citra dan target ke daftar
45             X.append(img)
46             T.append(TargetKelas[i])
47         #-----akhir loop :Pfor f in files-----
48 #-----akhir loop :for i in range(len(LabelKelas))----
49
50 #Mengubah List Menjadi numppy array
51 X=np.array(X)
52 T=np.array(T)
53 X=X.astype('float32')
54 T=T.astype('float32')
55 return X,T
56
57 def ModelDeepLearningCNN(JumlahKelas):
58     input_img = Input(shape=(128, 128, 3))
59     x = Conv2D(32, (3, 3), activation='relu', padding='same')
60     (input_img)
61     x = MaxPooling2D((2, 2), padding='same')(x)
62     x = Conv2D(32, (3, 3), activation='relu', padding='same')
63     (x)
64     x = MaxPooling2D((2, 2), padding='same')(x)
65     x = Conv2D(32, (3, 3), activation='relu', padding='same')
66     (x)
67     x = Flatten()(x)
68     x = Dense(100,activation='relu')(x)
69     x=Dense(JumlahKelas,activation='softmax')(x)
70     ModelCNN = Model(input_img, x)
71     ModelCNN.compile(loss='categorical_crossentropy',
72         optimizer='adam', metrics=['accuracy'])
73     #ModelCNN.compile(loss='mse', optimizer='adam', metrics
74     =['accuracy'])
75     return ModelCNN
76
77 def TrainingCNN(JumlahEpoh,DirektoriDataSet,LabelKelas,
78     NamaFileBobot):
79     #Membaca Data training dan label Kelas

```

```

X,D=LoadCitraTraining(DirektoriDataSet,LabelKelas)
73 JumlahKelas = len(LabelKelas)
#Membuat Model CNN
75 ModelCNN =ModelDeepLearningCNN(JumlahKelas)
#Trainng
77 history=ModelCNN.fit(X, D,epochs=JumlahEpoh,shuffle=
    True)
#Menyimpan hasil learning
79 ModelCNN.save>NamaFileBobot)
#Mengembalikan output
81 return ModelCNN,history

83
def Klasifikasi(DirDataSet,DirKlasifikasi,LabelKelas,
    ModelCNN=[]):
85 #Menyiapkan Data input Yang akan di kasifikasikan
    X=[]
87 ls = []
    DirKelas = DirDataSet+"/"+DirKlasifikasi
89 print(DirKelas)
    files = os.listdir(DirKelas)
91 n=0
    for f in files:
93         ff=f.lower()
            print(f)
95         if (ff.endswith('.jpg')|ff.endswith('.jpeg')|ff.
            endswith('.png')):
                ls.append(ff)
97         NmFile = os.path.join(DirKelas,f)
            img= cv2.imread(NmFile,1)
99         img=cv2.resize(img,(128,128))
            img= np.asarray(img)/255
101         img=img.astype('float32')
            X.append(img)
103         #----Akhir if-----
#---Akhir For
105 X=np.array(X)
X=X.astype('float32')
107 #Melakukan prediksi Klasifikasi
hs=ModelCNN.predict(X)

109
LKlasifikasi=[]
111 LKelasCitra =[]
n = X.shape[0]
113 for i in range(n):
    v=hs[i,:]
115     if v.max()>0.5:
        idx = np.max(np.where( v == v.max()))
117         LKelasCitra.append(LabelKelas[idx])

```

```

119         else:
120             idx=-1
121             LKelasCitra.append("-")
122             #-----akhir if
123             LKlasifikasi.append(idx)
124             #----akhir for
125             LKlasifikasi = np.array(LKlasifikasi)
126             return ls, hs, LKelasCitra
127
128 def LoadModel(sf):
129     ModelCNN=load_model(sf)
130     return ModelCNN
131
132
133
134 def ImageAugmentation(SPath,Kelas):
135     parent_dir = SPath
136     directory = Kelas
137     directoryExt =directory + "_ext"
138     sdirExt = os.path.join(parent_dir, directoryExt)
139     if not os.path.exists(sdirExt):
140         os.mkdir(sdirExt)
141     directory_image =directory
142     sDir = os.path.join(parent_dir, directory_image)
143     files = os.listdir(sDir)
144     ii=0
145     for f in files:
146         ff=f.lower()
147         if (ff.endswith('.jpg')|ff.endswith('.jpeg')|ff.
148             endswith('.png')):
149             print(ff)
150             sfs= os.path.join(sDir,ff)
151             img = load_img(sfs)
152             img2 = np.array(img)
153             sfn= os.path.join(sdirExt, ff)
154             cv2.imwrite(sfn,img2)
155             data = img_to_array(img)
156             samples = expand_dims(data, 0)
157             datagen = ImageDataGenerator(rotation_range=90,
158                 brightness_range=[0.2,2.0],zoom_range=[0.5,2.0],
159                 width_shift_range=0.2,height_shift_range=0.2)
160             it = datagen.flow(samples, batch_size=1)
161             for i in range(9):
162                 batch = it.next()
163                 image = batch[0].astype('uint8')
164                 now = datetime.now()
165                 ii=ii+1

```

```

163         sf = now.strftime("%Y%m%d%H%M%S")+"_"+str(ii)+".
    jpg"
        sfn= os.path.join(sdirExt, sf)
165     cv2.imwrite(sfn,image)

```

code/ModulKlasifikasiCitraCNN.py

### 3.2 Training Dataset

```

1 import ModulKlasifikasiCitraCNN as mCNN
3 # Nama folder datasetnya
DirektoriDataSet = "Kartu"
5
JumlahEpoh = 5
7
LabelKelas = (
9     "Dua Club",
    "Tiga Club",
11    "Empat Club",
    "Lima Club",
13    "Enam Club",
    "Tujuh Club",
15    "Delapan Club",
    "Sembilan Club",
17    "Sepuluh Club",
    "Jack Club",
19    "Queen Club",
    "King Club",
21    "Ace Club",
    "Dua Diamond",
23    "Tiga Diamond",
    "Empat Diamond",
25    "Lima Diamond",
    "Enam Diamond",
27    "Tujuh Diamond",
    "Delapan Diamond",
29    "Sembilan Diamond",
    "Sepuluh Diamond",
31    "Jack Diamond",
    "Queen Diamond",
33    "King Diamond",
    "Ace Diamond",
35    "Dua Heart",
    "Tiga Heart",
37    "Empat Heart",
    "Lima Heart",
39    "Enam Heart",
    "Tujuh Heart",

```

```

41     "Delapan Heart",
    "Sembilan Heart",
43     "Sepuluh Heart",
    "Jack Heart",
45     "Queen Heart",
    "King Heart",
47     "Ace Heart",
    "Dua Spade",
49     "Tiga Spade",
    "Empat Spade",
51     "Lima Spade",
    "Enam Spade",
53     "Tujuh Spade",
    "Delapan Spade",
55     "Sembilan Spade",
    "Sepuluh Spade",
57     "Jack Spade",
    "Queen Spade",
59     "King Spade",
    "Ace Spade",
61 )

63 # Mulai training
mCNN.TrainingCNN(JumlahEpoh, DirektoriDataSet, LabelKelas
, "BobotKartu.h5")

```

code/TrainingDataKartu.py

### 3.3 Deteksi Kartu dan Game

```

import cv2
2 import numpy as np
import copy
4 import ModulKlasifikasiCitraCNN as mCNN
import random
6
##### Inisiasi Kamera#####
8 vid = cv2.VideoCapture(2)

10 ##### Mengatur ukuran frame #####
fwidth = 640
12 fheight = 480

14 ##### Fungsi Deteksi Tepi #####
def preproses(img):
16     imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray,(5,5),1)
18     imgCanny = cv2.Canny(imgBlur,50,50)
    return imgCanny

```

```

20 ##### Fungsi Mendapatkan Kontur #####
22 def getcontours(img):
23     biggest = np.array([])
24     maxArea = 0
25     contours, hierarchy = cv2.findContours(img, cv2.
RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
26     for cnt in contours:
27         area = cv2.contourArea(cnt)
28         if area>2000:
29             cv2.drawContours(imgContour, cnt, -1, (255,
0, 0), 3)
30             peri = cv2.arcLength(cnt,True)
31             approx = cv2.approxPolyDP(cnt,0.02*peri,True)
32             if area > maxArea and len(approx) == 4:
33                 biggest = approx
34                 maxArea = area
35     return biggest
36
37 #####Load Model#####
38 model = mCNN.LoadModel("BobotKartu.h5")
39
40 ##### Iniasiasi Label Kelas #####
LabelKelas = (
41     "Dua Club",
42     "Tiga Club",
43     "Empat Club",
44     "Lima Club",
45     "Enam Club",
46     "Tujuh Club",
47     "Delapan Club",
48     "Sembilan Club",
49     "Sepuluh Club",
50     "Jack Club",
51     "Queen Club",
52     "King Club",
53     "Ace Club",
54     "Dua Diamond",
55     "Tiga Diamond",
56     "Empat Diamond",
57     "Lima Diamond",
58     "Enam Diamond",
59     "Tujuh Diamond",
60     "Delapan Diamond",
61     "Sembilan Diamond",
62     "Sepuluh Diamond",
63     "Jack Diamond",
64     "Queen Diamond",
65     "King Diamond",

```



```

68     "Ace Diamond",
    "Dua Heart",
70     "Tiga Heart",
    "Empat Heart",
    "Lima Heart",
72     "Enam Heart",
    "Tujuh Heart",
74     "Delapan Heart",
    "Sembilan Heart",
76     "Sepuluh Heart",
    "Jack Heart",
78     "Queen Heart",
    "King Heart",
80     "Ace Heart",
    "Dua Spade",
82     "Tiga Spade",
    "Empat Spade",
84     "Lima Spade",
    "Enam Spade",
86     "Tujuh Spade",
    "Delapan Spade",
88     "Sembilan Spade",
    "Sepuluh Spade",
90     "Jack Spade",
    "Queen Spade",
92     "King Spade",
    "Ace Spade",
94 )

96 ##### Fungsi Nilai Kartu #####
def getValue():
98     value = {
        "Dua Club": 2,
100     "Tiga Club": 3,
        "Empat Club": 4,
102     "Lima Club": 5,
        "Enam Club": 6,
104     "Tujuh Club": 7,
        "Delapan Club": 8,
106     "Sembilan Club": 9,
        "Sepuluh Club": 10,
108     "Jack Club": 11,
        "Queen Club": 12,
110     "King Club": 13,
        "Ace Club": 14,
112     "Dua Diamond": 2,
        "Tiga Diamond": 3,
114     "Empat Diamond": 4,
        "Lima Diamond": 5,

```

```

116         "Enam Diamond": 6,
        "Tujuh Diamond": 7,
118         "Delapan Diamond": 8,
        "Sembilan Diamond": 9,
120         "Sepuluh Diamond": 10,
        "Jack Diamond": 11,
122         "Queen Diamond": 12,
        "King Diamond": 13,
124         "Ace Diamond": 14,
        "Dua Heart": 2,
126         "Tiga Heart": 3,
        "Empat Heart": 4,
128         "Lima Heart": 5,
        "Enam Heart": 6,
130         "Tujuh Heart": 7,
        "Delapan Heart": 8,
132         "Sembilan Heart": 9,
        "Sepuluh Heart": 10,
134         "Jack Heart": 11,
        "Queen Heart": 12,
136         "King Heart": 13,
        "Ace Heart": 14,
138         "Dua Spade": 2,
        "Tiga Spade": 3,
140         "Empat Spade": 4,
        "Lima Spade": 5,
142         "Enam Spade": 6,
        "Tujuh Spade": 7,
144         "Delapan Spade": 8,
        "Sembilan Spade": 9,
146         "Sepuluh Spade": 10,
        "Jack Spade": 11,
148         "Queen Spade": 12,
        "King Spade": 13,
150         "Ace Spade": 14
    }
152     return value

154 ##### Fungsi Nampilin Kartu pada Frame Game #####
    def DrawResizedCard(img, card_image, pos, size):
156         resized_card = cv2.resize(card_image, size)
        img[pos[1]:pos[1]+size[1], pos[0]:pos[0]+size[0]] =
            resized_card
158         return img

160 ##### Dictionary Gambar Kartu Game #####
    card_images = {}
162 for label in LabelKelas:

```

```

        card_images[label] = cv2.imread(f"card/{label}.jpg")
164
##### Fungsi Nampilin Tulisan #####
166 def DrawText(img,sText,pos,color):
        font            = cv2.FONT_HERSHEY_SIMPLEX
168        posf           = pos
        fontScale        = .7
170        fontColor      = color
        thickness         = 2
172        lineType       = 2
        cv2.putText(img,sText,
174                posf,
                font,
176                fontScale,
                fontColor,
178                thickness,
                lineType)
180        return copy.deepcopy(img)

182
##### Membaca Video #####
184 success, frame = vid.read()

186 if not success:
        print("Gagal membaca video")
188        exit()

190 ##### Inisialisasi Frame Game #####
        game_frame = np.zeros_like(frame)
192
##### Inisialisasi list kartu #####
194 opened_card = []
        player_card = []
196 computer_card = []

198 ##### inisiasi score game #####
        score_player = 0
200 score_computer = 0

202 while True:
        success, frame = vid.read()
204
        if not success:
206                break

208
        frame = cv2.resize(frame, (fwidth, fheight))
210        imgContour = frame.copy()
        preprocessed_img = preproses(frame)

```

```

212     biggest = getcontours(preprocessed_img)
214
215     if len(biggest) != 0:
216         ##### Memotong bagian contour #####
217         x, y, w, h = cv2.boundingRect(biggest)
218         cropped_frame = frame[y:y+h, x:x+w]
219         cv2.imshow('Crop', cropped_frame)
220
221         ##### Prepare image for prediction
222         #####
223         X = []
224         image = cv2.resize(cropped_frame, (128, 128))
225         image = np.asarray(image) / 255.0
226         image = image.astype('float32')
227         X.append(image)
228         X = np.array(X)
229
230         ##### Predict #####
231         hs = model.predict(X, verbose=0)
232         n = np.argmax(hs)
233
234         label_text = LabelKelas[n]
235         result_img = DrawText(frame.copy(), label_text,
236                               (200, 100), (0, 0, 255))
237     else:
238         result_img = frame.copy()
239
240     cv2.imshow('Result Contour', imgContour)
241     cv2.imshow('Result Prediction', result_img)
242
243     key = cv2.waitKey(1) & 0xFF
244     if key == ord('z'):
245         break
246     ##### menampilkan dan score #####
247     elif key == ord(' '):
248         if len(biggest) != 0:
249             opened_card.append((label_text, (0, 0, 255)))
250             player_card.append((label_text, (0, 0, 255)))
251             computer_take = random.choice(LabelKelas)
252             opened_card.append((computer_take, (255, 0,
253             0)))
254             computer_card.append((computer_take, (255, 0,
255             0)))
256
257             game_frame = np.zeros_like(frame)

```

```

256         game_frame = DrawText(game_frame, "Computer
Card", (15, int(fheight * 0.05)), (255, 0, 0))
        game_frame = DrawText(game_frame, "Player
Card", (15, int(fheight * 0.5)), (0, 255, 0))
258
        ##### Menampilkan Kartu #####
260        card_size = (96, 144) # ukuran kartu pada
game frame
        for i, (text, _) in enumerate(opened_card):
262            if i % 2 == 0: # player
                y_offset = int(fheight * 0.55)
264                x_offset = 20 + i * 50
            else: # computer
266                y_offset = int(fheight * 0.1)
                x_offset = 20 + (i-1) * 50
268
                card_img = card_images[text]
270                game_frame = DrawResizedCard(game_frame,
card_img, (x_offset, y_offset), card_size)
272
                ##### Perhitungan Score #####
                if len(player_card) >= 1 and len(
computer_card) >= 1:
274                    player_value = getValue()[player_card
[-1][0]]
                    computer_value = getValue()[computer_card
[-1][0]]
276
                    if player_value > computer_value:
278                        score_player += 1
                    elif computer_value > player_value:
280                        score_computer += 1
                    else:
282                        pass # Nilai kartu sama tidak ada
penambahan score
284
                game_frame = DrawText(game_frame, f"Score
Player: {score_player}", (350, int(fheight * 0.5)),
(0, 255, 0))
                game_frame = DrawText(game_frame, f"Score
Computer: {score_computer}", (350, int(fheight * 0.05)
), (255, 0, 0))
286
288                cv2.imshow('GAME', game_frame)
290
                ##### menampilkan pemenang #####
                elif key == ord('a'):
292                    winner = ""

```

```

294         if score_player > score_computer:
295             winner = "Player wins!"
296         elif score_computer > score_player:
297             winner = "Computer wins!"
298         else:
299             winner = "It's a tie!"
300
301     print(winner)
302     # Mengosongkan frame
303     game_frame = np.zeros_like(frame)
304     # Menampilkan pemenang pada konsol
305     game_frame = DrawText(game_frame, winner, (int(
306         fwidth * 0.45), int(fheight * 0.5)), (0, 0, 255))
307     cv2.imshow('GAME', game_frame)
308
309 vid.release()
310 cv2.destroyAllWindows()

```

code/DeteksiKartu.py

## 4 Penjelasan Source Code

### 4.1 Modul CNN

Source code ini merupakan modul yang akan digunakan untuk mendeteksi kartu.

### 4.2 Training Data set

Source code ini berguna untuk melakukan training pada dataset yang dimiliki yang akan menghasilkan file Bobotkartu.h5 yang akan dijadikan model pada source code deteksi kartu dan game

### 4.3 Deteksi Kartu dan Game

Berikut penjelasan dari code - code pada source code Deteksi Kartu dan Game

- Fungsi preproses(img)

Pada fungsi ini terdapat beberapa proses yaitu mengubah menjadi citra Grayscale, kemudian melakukan penggaburan dengan gaussianblur, dan terakhir menggunakan cv2.canny untuk menemukan tepi.

- Fungsi getcontours()

Fungsi getcontours digunakan untuk mendapatkan kontur terbesar dari suatu citra, dan kemudian memproses kontur tersebut untuk mendeteksi objek berbentuk empat persegi panjang.

- Fungsi `getValue()`  
Fungsi ini berfungsi untuk memberi nilai atau value kepada tiap kartu yang ada.
- Fungsi `DrawResizedCard()`  
Fungsi ini berguna untuk menampilkan gambar kartu yang sudah diresize.
- Kode berikut berguna untuk membuat dictionary yang memetakan label setiap kartu dengan gambar yang sesuai.

```

card_images = {}
2 for label in LabelKelas:
    card_images[label] = cv2.imread(f"card/{label}.jpg")
4

```

- Fungsi `DrawText()`  
Fungsi ini berguna untuk menampilkan teks.
- Pada bagian berikut menginisiasi frame untuk game, frame ini adalah frame kosong hitam. Inisiasi list kartu, `opened_card` menyimpan kartu yang sudah dibuka, `player_card` merupakan kartu untuk player dan `computer_card` adalah kartu untuk computer. Inisiasi score awal player dan computer adalah nol

```

1 ##### Inisialisasi Frame Game #####
game_frame = np.zeros_like(frame)
3
4 ##### Inisialisasi list kartu #####
5 opened_card = []
6 player_card = []
7 computer_card = []
8
9 ##### inisiasi score game #####
10 score_player = 0
11 score_computer = 0

```

- Pada source code berikut melakukan pemotongan bagian yang terkena kontur atau memisahkan kartu dengan background. hasil pemotongan yang akan di masukan ke model untuk melakukan prediksi atau deteksi kartu yang ditampilkan.

```

1 if len(biggest) != 0:
    ##### Memotong bagian contour #####
3     x, y, w, h = cv2.boundingRect(biggest)
    cropped_frame = frame[y:y+h, x:x+w]
5     cv2.imshow('Crop', cropped_frame)
6
7     ##### Prepare image for prediction #####
    X = []
9     image = cv2.resize(cropped_frame, (128, 128))
    image = np.asarray(image) / 255.0
11    image = image.astype('float32')
    X.append(image)
13    X = np.array(X)

```

```

15     ##### Predict #####
16     hs = model.predict(X, verbose=0)
17     n = np.argmax(hs)
18
19     label_text = LabelKelas[n]
20     result_img = DrawText(frame.copy(), label_text, (200,
21     100),(0,0,255))
22     else:
23         result_img = frame.copy()
24
25     cv2.imshow('Result Contour', imgContour)
26     cv2.imshow('Result Prediction', result_img)

```

- Pada bagian berikut saat dipencet tombol spasi maka akan menampilkan gambar kartu yang diteksi sebagai kartu player, menampilkan kartu computer yang di random dan score pada game frame.  
Pada bagian ini terjadi perhitungan score dimana akan membandingkan tiap kartu yang ada pada player dan computer. Apabila nilai kartu player lebih besar dari kartu computer maka score player akan bertambah satu, begitupun sebaliknya.

```

1 elif key == ord(' '):
2     if len(biggest) != 0:
3
4         opened_card.append((label_text, (0, 0, 255)))
5         player_card.append((label_text, (0, 0, 255)))
6         computer_take = random.choice(LabelKelas)
7         opened_card.append((computer_take, (255, 0, 0)))
8         computer_card.append((computer_take, (255, 0, 0)))
9
10        game_frame = np.zeros_like(frame)
11
12        game_frame = DrawText(game_frame, "Computer Card",
13        (15, int(fheight * 0.05)), (255, 0, 0))
14        game_frame = DrawText(game_frame, "Player Card", (15,
15        int(fheight * 0.5)), (0, 255, 0))
16
17        ##### Menampilkan Kartu #####
18        card_size = (96, 144) # ukuran kartu pada game frame
19        for i, (text, _) in enumerate(opened_card):
20            if i % 2 == 0: # player
21                y_offset = int(fheight * 0.55)
22                x_offset = 20 + i * 50
23            else: # computer
24                y_offset = int(fheight * 0.1)
25                x_offset = 20 + (i-1) * 50
26
27            card_img = card_images[text]
28            game_frame = DrawResizedCard(game_frame, card_img
29            , (x_offset, y_offset), card_size)
30
31            ##### Perhitungan Score #####
32            if len(player_card) >= 1 and len(computer_card) >= 1:
33                player_value = getValue()[player_card[-1][0]]
34                computer_value = getValue()[computer_card[-1][0]]

```



```

33         if player_value > computer_value:
34             score_player += 1
35         elif computer_value > player_value:
36             score_computer += 1
37         else:
38             pass # Nilai kartu sama tidak ada penambahan
39     score
40
41     game_frame = DrawText(game_frame, f"Score Player: {
42     score_player}", (350, int(fheight * 0.5)), (0, 255, 0))
43     game_frame = DrawText(game_frame, f"Score Computer: {
44     score_computer}", (350, int(fheight * 0.05)), (255, 0, 0))
45
46     cv2.imshow('GAME', game_frame)

```

- Pada bagian berikut apabila di tekan tombol a maka akan menampilkan pemenang dari game.

```

1     elif key == ord('a'):
2         winner = ""
3         if score_player > score_computer:
4             winner = "Player wins!"
5         elif score_computer > score_player:
6             winner = "Computer wins!"
7         else:
8             winner = "It's a tie!"
9
10        print(winner)
11        # Mengosongkan frame
12        game_frame = np.zeros_like(frame)
13        # Menampilkan pemenang pada konsol
14        game_frame = DrawText(game_frame, winner, (int(fwidth *
15        0.45), int(fheight * 0.5)), (0, 0, 255))
16        cv2.imshow('GAME', game_frame)

```

## 5 Hasil Deteksi Kartu dan Game

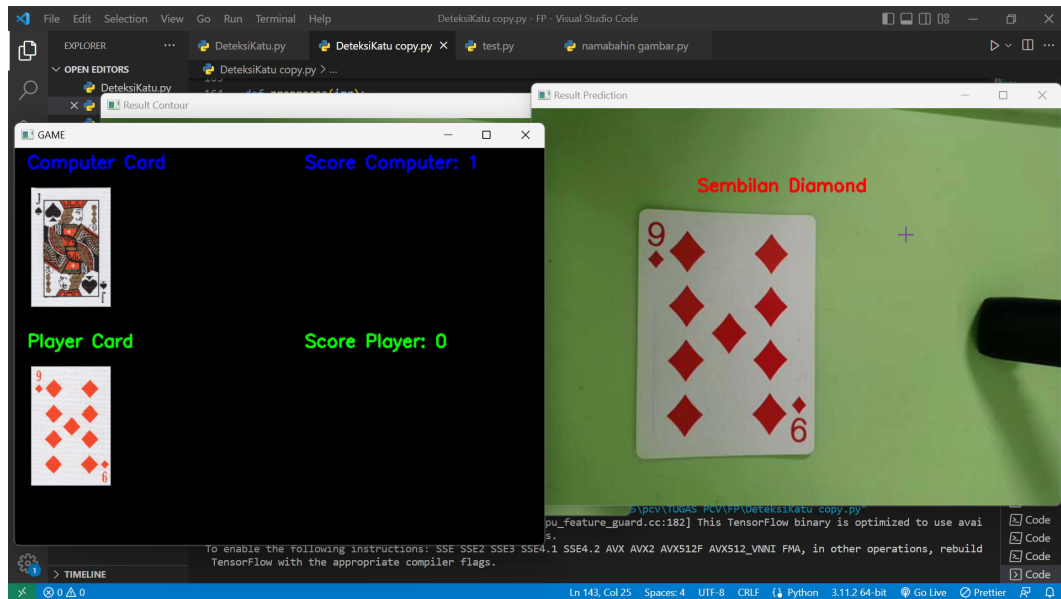


Figure 1: Deteksi Kartu dan Menampilkan dalam game

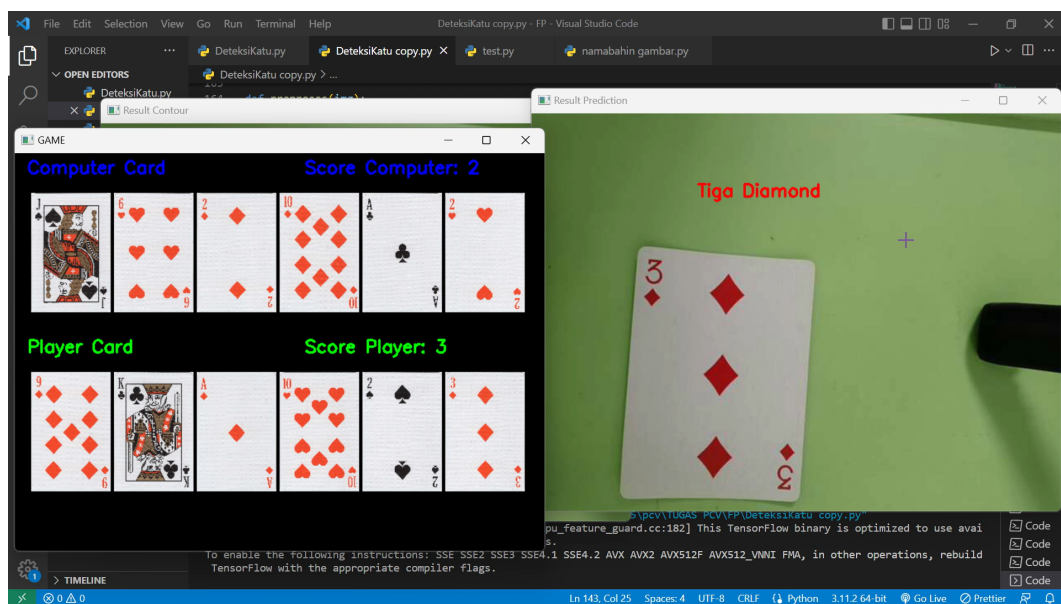


Figure 2: Game saat telah dibuka semua kartu

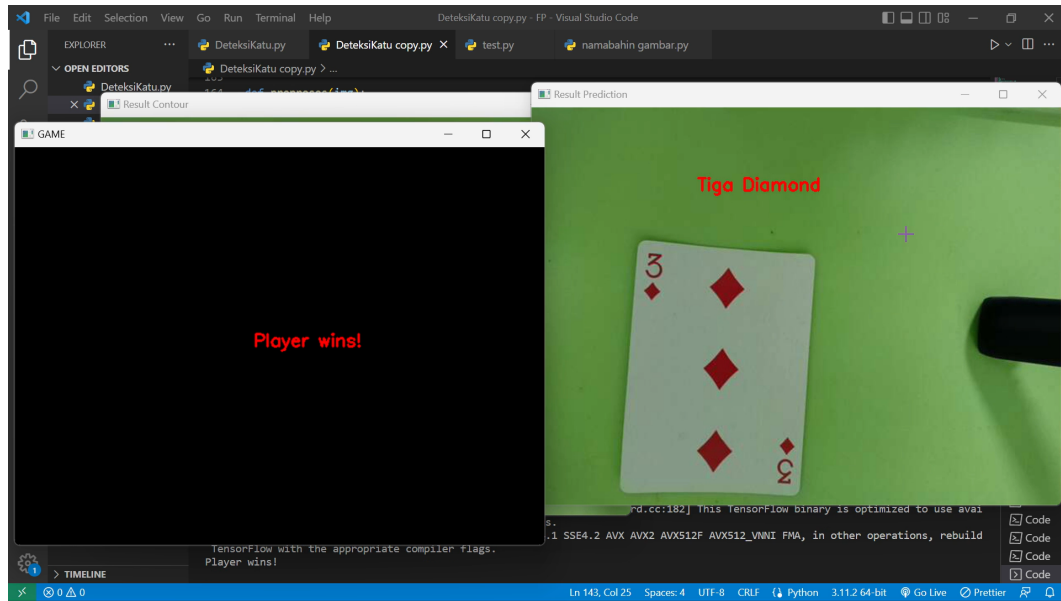


Figure 3: Menampilkan Pemenang