

WANDA (WhatsApp Tanpa Derita)

logbook



Dinda Majesty

1.18.4.011

Tri Angga Dio Simamora

1.18.4.047

Informatics Research Center

Applied Bachelor Program of Informatics Engineering

Bandung

2019

‘Jika Kamu tidak dapat menahan lelahnya belajar,
Maka kamu harus sanggup menahan perihnya Kebodohan.’
Imam Syafi’i

Contents

1	Pertemuan 1	1
1.1	Issues #1	1
1.2	Issues #2	1
1.3	Issues #3	2
1.4	Issues #4	2
1.5	Issues #5	2
1.6	Issues #6	3
1.7	Issues #7	3
1.8	Issues #8	4
1.9	Issues #9	4
1.10	Issues #10	4
2	Pertemuan 2	5
2.1	Issues #11	5
2.2	Issues #12	5
2.3	Issues #13	5
2.4	Issues #14	6
2.5	Issues #15	6
2.6	Issues #16	6
2.7	Issues #17	6
2.8	Issues #18	6
2.9	Issues #19	7
2.10	Issues #20	7
3	Pertemuan 3	8
3.1	Issues #21	8
3.2	Issues #22	8
3.3	Issues #23	8

3.4	Issues #24	9
3.5	Issues #25	9
3.6	Issues #26	9
3.7	Issues #27	9
3.8	Issues #28	10
3.9	Issues #29	10
3.10	Issues #30	11
4	Pertemuan 4	12
4.1	Issues #31	12
4.2	Issues #32	12
4.3	Issues #33	12
4.4	Issues #34	12
4.5	Issues #35	13
4.6	Issues #36	13
4.7	Issues #37	13
4.8	Issues #38	13
4.9	Issues #39	13
4.10	Issues #40	13
5	Pertemuan 5	14
5.1	Issues #41	14
5.2	Issues #42	14
5.3	Issues #43	15
5.4	Issues #44	15
5.5	Issues #45	16
5.6	Issues #46	16
5.7	Issues #47	17
5.8	Issues #48	17
5.9	Issues #49	17
5.10	Issues #50	17
6	Pertemuan 6	18
6.1	Issues #51	18
6.2	Issues #52	18
6.3	Issues #53	18
6.4	Issues #54	19

6.5	Issues #55	20
6.6	Issues #56	20
6.7	Issues #57	21
6.8	Issues #58	21
6.9	Issues #59	21
6.10	Issues #60	21
7	Pertemuan 7	22
7.1	Issues #61	22
7.2	Issues #62	22
7.3	Issues #63	22
7.4	Issues #64	22
7.5	Issues #65	23
7.6	Issues #66	23
7.7	Issues #67	23
7.8	Issues #68	23
7.9	Issues #69	24
7.10	Issues #70	24
8	Pertemuan 8	25
8.1	Issues #71	25
8.2	Issues #72	25
8.3	Issues #73	25
8.4	Issues #74	25
8.5	Issues #75	25
8.6	Issues #76	26
8.7	Issues #77	26
8.8	Issues #78	26
8.9	Issues #79	26
8.10	Issues #80	26

Chapter 1

Pertemuan 1

1.1 Issues #1

Pada *issues #1 (Selenium Can't Close Current Tab)* permasalahannya yaitu tab yang telah terbuka tidak mau ditutup dengan cara *shortcut key* yang pada umumnya digunakan yaitu (CTRL + W). Pemecahan masalahnya yaitu dengan menggunakan *method*

```
self.driver.switch_to_window(self.driver.window_handles[1])
```

sehingga tab tersebut akan pindah sesuai dengan *list* yang kita pilih. Setelah *switch* jalankan method

```
self.driver.close()
```

untuk mengclose tab pada *list* yang kita pilih lalu, lakukan *switch* lagi untuk kembali ke halaman awal.

1.2 Issues #2

Pada *issues #2 (Detecting The New Message on Whatsapp Web with Selenium)* yaitu bagaimana caranya kita mendeteksi bahwa ada pesan yang masuk pada *WhatsApp*.

```
try:
```

```
    self.chat = self.driver.find_elements_by_class_name("P6z4j")[0]
    self.chat.click()
    self.chat.click()
    self.chat.click()
```

```
except:
```

```
    print("no message")
```

Pendeteksian pesan ada atau tidaknya yaitu pada *"P6z4j"* yaitu *class name* pada *HTML* dan *list* yang ke 0, karena 0 merupakan pesan yang muncul pertama maka kita akan mengklik pesan yang pertama kali muncul. Setelah, terdeteksi ada pesan masuk maka *bot* akan mengklik 3 kali dimaksudkan agar *bot* benar-benar mengklik pesan tersebut. *Try and except* digunakan dikarenakan program berjalan secara terus-menerus sehingga *bot* akan mencoba mencari apakah ditemukan *class* namanya, jika tidak ditemukan maka akan mencetak tulisan *"no message"*.

1.3 Issues #3

Pada *issues #3 (Detecting A Value Message from Sender to ChatBot)* yaitu kita akan mengambil sebuah *value* dari isi *chat* yang *user* kirimkan ke *bot*.

```
try:
    self.span = self.driver.find_elements_by_xpath('(./span)')[-10].text
except:
    print("sender not sending some message")
```

Berdasarkan beberapa percobaan yang kita lakukan, pesan terakhir terdapat polanya yaitu pesan selalu berapa di *tag span* dengan *list -10*. Setelah kita temukan polanya, kita transformasikan dari *object selenium* menjadi sebuah *text* atau *string* sehingga *(.text)* digunakan untuk mentransformasikannya.

1.4 Issues #4

Pada *issues #4 (Typing and Sending Message with Whatsapp Web Selenium (Python))* mengetik dan mengirim pesan adalah fitur penting di Chatbot, jadi setelah chatbot menerima pesan dan membaca pesan, chatbot harus menanggapi obrolan. Jadi chatbot akan mengetik dan mengirim secara otomatis dengan parameter pesan apa yang harus dikirim oleh chatbot.

1.5 Issues #5

Pada *issues #5 (The Program Will Wait Until User Scan Barcode)* fungsi ini digunakan untuk agar *program* menunggu admin sampai memindai kode *barcode*, agar program tidak *error* sampai kode *barcode* muncul dan admin memindainya.

```
def waitLogin(self):
    self.target = '"_3RWII"'
    self.x_arg = '//div[contains(@class, ' + self.target + ')]'
    self.wait = WebDriverWait(self.driver, 600)
    self.wait.until(EC.presence_of_element_located((By.XPATH, self.x_arg)))
```

1.6 Issues #6

Pada *issues #6 (Error: TypeError)* *TypeError* terjadi ketika sebuah parameter yang diberikan tidak sesuai dengan fungsi yang kita jalankan, contohnya ***len(42)*** ini akan menyebabkan *TypeError* dikarenakan fungsi *len()* untuk menghitung berapa panjang dari sebuah variabel yang ditampung selain numerik. Dalam kasus ini, *TypeError* yang terjadi adalah

```
#class object
class Chatbot (object):
    def __init__(self, filename):
        self.fileName = filename
```

```
#initiation
```

```
#error
```

```
run = chatbot.Chatbot()
```

```
#fixed
```

```
run = chatbot.Chatbot("data")
```

dikarenakan saat inisiasi *object* harus ada parameter yang diberikan pada *class Chatbot*.

1.7 Issues #7

Pada *issues #7 (Error: NoSuchElementException)* *NoSuchElementException* adalah *error* yang terjadi ketika selenium ingin mencari sebuah object dari HTML tetapi tidak dapat ditemukan object tersebut. Pemecahan masalahnya adalah dengan cara menggunakan metode *Try and Except*

1.8 Issues #8

Pada *issues #8 (Error: ListIndexOutOfRangeException)* *ListIndexOutOfRangeException* adalah *error* yang terjadi ketika indeksnya tidak ada atau diluar jangkauan.

1.9 Issues #9

Pada *issues #9 (Error: FileNotFoundException)* *FileNotFoundException* adalah *error* yang terjadi ketika parameter file yang kita berikan tetapi program tidak dapat menemukan file yang dijadikan parameter tersebut. Pemecahan masalahnya adalah dengan cara mengecek apakah file tersebut sudah benar dalam hal penamaan, lokasi, dan apakah file tersebut ada.

1.10 Issues #10

Pada *issues #10 (Error: WebDriverException)* *WebDriverException* adalah *error* yang disebabkan ketika kita ingin menjalankan sebuah selenium driver tetapi internet kita dalam keadaan tidak tersambung atau terputus, jadi pastikan internet tersambung.

Chapter 2

Pertemuan 2

2.1 Issues #11

Pada *issues #11 (Movie Schedule)* chatbot akan mencari jadwal film dengan parameter yang diberikan yaitu kata kunci, nama kota, nama bioskop, nama lokasi.

```
if "bioskop" in self.message:
    self.movieSchedule(self.message)
```

Jika kata kunci "bioskop" terdeteksi pada pesan yang dikirimkan oleh pengirim pesan maka chatbot akan menjalankan method `movieSchedule` yang berisi variabel nama kota, nama bioskop, dan nama lokasi yang akan diinputkan berdasarkan pesan yang dikirimkan oleh pengirim pesan.

2.2 Issues #12

Pada *issues #12 (Searching Product on Tokopedia E-Commerce Site)* chatbot akan mencari produk yang diinginkan oleh pengirim pesan pada website Tokopedia. Namun, kami menggantinya menjadi website tokoperhutani.

2.3 Issues #13

Pada *issues #13 (Chatbot Send Link The Product for Confirmation)* chatbot akan mengirimkan link dari produk yang telah ditemukan oleh chatbot kepada pengirim pesan untuk melakukan konfirmasi apakah benar produk tersebut yang sedang dicari oleh pengirim pesan. Fitur ini kami ganti menjadi chatbot yang dapat mengirimkan link pembayaran tokoperhutani.

2.4 Issues #14

Pada *issues #14 (The Chatbot Sort The Price)* chatbot akan mencari produk yang diinginkan oleh pengirim pesan dan mengurutkan harga produk, agar pengirim pesan mendapatkan produk yang diinginkan dengan harga yang murah. Fitur ini telah ditiadakan karena pada tokoperhutani tidak ada fitur untuk mengurutkan harga.

2.5 Issues #15

Pada *issues #15 (The bot will automatically click the product after sort from cheap to expensive.)* chatbot akan mengklik produk yang telah diurutkan harganya, sehingga chatbot akan memilih produk dengan harga yang murah. Fitur ini telah ditiadakan karena pada tokoperhutani kami hanya menerapkan chatbot yang dapat mengklik produk sesuai dengan kode produk yang telah dikirimkan oleh pengirim pesan dan melakukan checkout serta mengirimkan link pembayaran kepada pengirim pesan.

2.6 Issues #16

Pada *issues #16 (The bot will ask you to check the link and confirmation is that product that you want or not. if yes "agree" if not "disagree".)* chatbot akan mengirimkan link dan pesan agar pengirim pesan melakukan pengecekan terhadap link produk yang telah dikirimkan oleh chatbot dengan tujuan untuk meminta persetujuan pengirim pesan untuk melakukan pembelian terhadap produk tersebut. Fitur ini telah ditiadakan dan diganti menjadi website tokoperhutani.

2.7 Issues #17

Pada *issues #17 (The bot will ask you what payment methods that you want to use. example: BNI, BCA, etc)* chatbot akan mengirimkan pesan yang berisi pertanyaan tentang metode pembayaran apa yang diinginkan oleh pengirim pesan, sehingga chatbot dapat memilihkan metode pembayaran yang sesuai dengan yang diinginkan pengirim pesan. Fitur ini telah diganti menjadi metode pembayaran pada tokoperhutani.

2.8 Issues #18

Pada *issues #18 (Send The Payment Number)* chatbot akan mengirimkan kode pembayaran kepada pengirim pesan. Fitur ini diganti menjadi chatbot yang mengirimkan

link pembayaran tokoperhutani kepada pengirim pesan.

2.9 Issues #19

Pada *issues #19 (Deadline Payment)* *chatbot* akan memberitahukan kepada pengirim pesan deadline pembayaran dari produk yang telah dipesan. Fitur ini telah ditiadakan.

2.10 Issues #20

Pada *issues #20 (Amount The Products)* *chatbot* akan mengirimkan pemberitahuan tentang jumlah barang yang akan dibeli oleh pengirim pesan. Fitur ini telah ditiadakan.

Chapter 3

Pertemuan 3

3.1 Issues #21

Pada *issues #21 (Login Variable) chatbot* membutuhkan dua variabel agar bisa login website tokoperhutani variabel yaitu username dan password. Variabel ini akan secara otomatis terisi pada bagian username dan password dan akan secara otomatis klik tombol login. pengisian username dan password serta klik tombol login dilakukan secara otomatis oleh chatbot.

3.2 Issues #22

Pada *issues #22 (send_keys Method)* dengan menggunakan *method send_keys chatbot* dapat mengisi username dan password pada kolom username dan password hanya dengan mengirimkan username dan password yang telah diinputkan pada variabel `usEmail` dan `usPass`.

```
usEmail = "email"  
usPass = "password"
```

```
self.driver.find_element_by_id("email").send_keys(usEmail)  
self.driver.find_element_by_id("password").send_keys(usPass)
```

3.3 Issues #23

Pada *issues #23 (.click Method on Pyhton Selenium)* `.click` digunakan untuk mengklik sebuah tombol yang terdapat pada website. *chatbot* akan mengklik tombol login secara otomatis menggunakan `method .click`

```
self.driver.find_elements_by_class_name("le-button")[0].click()
```

3.4 Issues #24

Pada *issues #24 (daftarBelanjaan Variable)* variabel `daftarBelanjaan` berisi list kode produk yang akan dibeli oleh pengirim pesan, variabel memiliki tipe data list agar pengirim pesan dapat melakukan pembelian terhadap satu atau lebih item produk.

```
daftarBelanjaan = ['193150214695', '193150214696', '193150214751', '193150215151',  
                  '193150215166', '193150215173', '193150215178', '193150215190',  
                  '193150215192', '193150215398', '193150215511', '193150215524',  
                  '193150214698', '193150215373']
```

3.5 Issues #25

Pada *issues #25 (varTableData Variable)* variabel ini berguna untuk menampung semua item data yang terdapat pada website tokoperhutani.

```
varTableData = self.driver.find_elements_by_xpath("//table[@id='example'  
                                                    and @class='display select now.'  
                                                    /tbody/tr")
```

3.6 Issues #26

Pada *issues #26 (Variable List for Counting items)* variabel ini berguna untuk menampung hasil perhitungan jumlah data yang terdapat pada website tokoperhutani. Variabel `forCounting` memiliki tipe data list, karena jumlah data yang terdapat pada website tokoperhutani sangat banyak.

```
forCounting = []
```

3.7 Issues #27

Pada *issues #27 (for i in varTableData to Looping)* perulangan ini digunakan untuk melakukan perhitungan terhadap item yang terdapat pada website tokoperhutani.

```
for i in asd:  
    abc = i.text[9:22]  
    waduwek = abc.splitlines()  
    forCounting.append(waduwek)
```

```
itungan = 10 - len(forCounting)
```

3.8 Issues #28

Pada *issues #28 (while cariData to While Loop)* pengecekan terhadap item yang terdapat pada daftarBelanjaan dengan item data yang ada pada website tokoperhutani. Apabila item data pada daftarBelanjaan ada pada varTableData maka item data tersebut akan diklik untuk dimasukkan kedalam keranjang belanjaan.

```
while cariData:
varTableData = self.driver.find_elements_by_xpath("//table[@id='example' and @class='table']")

for i in varTableData:
    itungan += 1
    abc = i.text[9:22]
    waduwek = abc.splitlines()

    if listOfOrder[0] in daftarBelanjaan:
        i.click()
        wektow = daftarBelanjaan.index(listOfOrder[0])
        daftarBelanjaan.pop(wektow)

    if len(daftarBelanjaan) == 0:
        cariData = False
        self.driver.find_elements_by_class_name("le-button")[1].click()
    if itungan == 10 and len(daftarBelanjaan) >= 1:
        print("masih ada belanjaan lanjut")
        itungan = 0
        self.driver.find_element_by_id("example_previous").click()
```

3.9 Issues #29

Pada *issues #29 (for i in varTableData to Loop The Second Data)* perulangan ini bertujuan untuk melakukan pengecekan dan perhitungan terhadap item yang terdapat pada website tokoperhutani.

```
for i in varTableData:
    itungan += 1
    abc = i.text[9:22]
    waduwek = abc.splitlines()
```

3.10 Issues #30

Pada *issues #30* (*if listOfOrder[0] in daftarBelanjaan Variable*) pengecekan ini digunakan untuk membandingkan apakah item yang terdapat pada daftarBelanjaan ada pada varTableData, jika ada maka item tersebut akan diklik.

```
if listOfOrder[0] in daftarBelanjaan:
    i.click()
    wektow = daftarBelanjaan.index(listOfOrder[0])
    daftarBelanjaan.pop(wektow)
```


Chapter 4

Pertemuan 4

4.1 Issues #31

Pada *issues #31 (Google Maps Feature)* chatbot akan menggunakan fitur *google maps* sehingga algoritmanya yaitu user mengirim location lalu mengetikkan kata kunci "gmaps (tujuan)" sehingga chatbot akan mengirimkan link yang akan menunjukkan jalan ketempat yang ingin dituju.

4.2 Issues #32

Pada *issues #32 (Use Try and Except to Maps)* dengan menggunakan metode *Try and Except* untuk mencari kelas yang berisi lokasi, jika ada, fungsi gmaps akan dijalankan.

4.3 Issues #33

Pada *issues #33 (gmaps.click)* fungsi ini digunakan untuk mengklik *maps* yang user kirim kepada chatbot.

4.4 Issues #34

Pada *issues #34 (Switch BrowserTab)* setelah fungsi *gmaps.click()* dijalankan maka akan membuka tab baru dan mengharuskan untuk menggunakan metode *switch_window* untuk mengganti tab pada browser.

4.5 Issues #35

Pada *issues #35 (searchboxinput)* di *field "searchboxinput"* ada value lokasi koordinat lokasi user, kita harus ambil koordinat tersebut untuk kita tampung di variabel koordinat yang fungsinya untuk *direction*.

4.6 Issues #36

Pada *issues #36 (taparea Class Name)* sebuah *class name* pada *HTML* yang dapat digunakan pada selenium untuk diproses seperti klik.

4.7 Issues #37

Pada *issues #37 (tactile_searchbox_input)* sebuah nama kelas pada *field* yang ada di *HTML* pada halaman *google maps*. Pada field ini akan diisi oleh variabel koordinat yang kita dapatkan dari nama kelas "searchboxinput".

4.8 Issues #38

Pada *issues #38 (currenturl)* sebuah variabel yang akan diisi oleh *url* setelah bot mendapatkan *direction* pada proses google maps.

4.9 Issues #39

Pada *issues #39 (destination Variable)* sebuah variabel string yang kita dapatkan dari pesan user setelah kata kunci "gmaps".

4.10 Issues #40

Pada *issues #40 (sb_cb50 id name)* adalah sebuah nama id yang ada di *HTML* yang berfungsi untuk menghapus sebuah field pada "searchbox".

Chapter 5

Pertemuan 5

5.1 Issues #41

Pada *issues #41* (*Method retrievePicture*) sebuah fungsi untuk mendownload gambar dari WhatsApp.

```
def retrievePicture(self):  
    self.driver.find_elements_by_class_name("_18vxA")[-1].click()  
    sleep(1)  
  
    self.driver.find_element_by_css_selector("span[data-icon='download']").click()  
    sleep(1)  
  
    self.driver.find_element_by_css_selector("span[data-icon='x-viewer']").click()  
    sleep(1)
```

5.2 Issues #42

Pada *issues #42* (*Method getName*) sebuah fungsi untuk mengambil nama dari pengirim jika individual, dan akan mengambil nama grup di pengirim berada dalam sebuah grup.

```
def getName(self):  
    try:  
        self.driver.find_element_by_class_name("_3fs0K").click()  
        sleep(1)  
  
        self.driver.find_element_by_class_name("_2vJ0g").click()
```

```

        sleep(1)

        name = self.driver.find_elements_by_class_name("_F7Vk")[1].text
        sleep(1)

        self.driver.find_element_by_css_selector("span[data-icon='x-viewer']").click()
        sleep(1)
    except Exception as e:
        print(e)
        print("Grup")

        name = self.driver.find_elements_by_class_name("_3u328")[0].text
        sleep(1)

    return name

```

5.3 Issues #43

Pada *issues #43* (*Method renamePicture*) sebuah fungsi untuk mengubah nama berkas yang telah kita download.

```

def renamePicture(self, fileName):
    dir_name = "/Users/trian/Downloads/"
    list = os.listdir(dir_name)

    print(list)

    for item in list:
        if item.endswith(".jpeg"):
            os.rename(os.path.join(dir_name, item), os.path.join(dir_name,
                                                                    fileName + ".jpeg"))

```

5.4 Issues #44

Pada *issues #44* (*Method deletePicture*) sebuah fungsi untuk menghapus berkas foto yang ada dikomputer.

```
def deletePicture(self):
    dir_name = "/Users/trian/Downloads/"
    list = os.listdir(dir_name)

    for item in list:
        if item.endswith(".jpeg"):
            os.remove(os.path.join(dir_name, item))
```

5.5 Issues #45

Pada *issues #45* (*Method sendPicture*) sebuah fungsi untuk mengupload dan mengirimkan berkas foto dari komputer ke WhatsApp.

```
def sendPicture(self, phoneNumber, filePath):
    self.driver.get("https://web.whatsapp.com/send?phone=" + phoneNumber)

    self.waitLogin()
    sleep(3)

    self.driver.find_element_by_css_selector("span[data-icon='clip']").click()
    sleep(2)

    path = r"C:\Users\trian\Downloads"
    nameFile = filePath + ".jpeg"

    result = os.path.join(path, nameFile)

    self.driver.find_element_by_css_selector("input[type='file']").send_keys(result)
    sleep(1)

    self.driver.find_element_by_css_selector("span[data-icon='send-light']").click()
    sleep(1)
```

5.6 Issues #46

Pada *issues #46* (*os.path.join*) sebuah fungsi untuk menggabungkan 2 path menjadi satu, contoh: `path1 = "/users/pisanggoreng/"` dan `path2 = "Downloads/"` maka

ketika kita lakukan fungsi `os.path.join` maka hasilnya akan seperti `"/users/pisang-goreng/Downloads/"`

5.7 Issues #47

Pada *issues #47* (`os.remove`) sebuah fungsi untuk menghapus sebuah file maupun folder pada komputer.

5.8 Issues #48

Pada *issues #48* (`os.listdir`) sebuah fungsi untuk menampilkan sebuah folder/directory yang berada pada path.

5.9 Issues #49

Pada *issues #49* (`os.rename`) sebuah fungsi untuk mengubah nama file atau folder.

5.10 Issues #50

Pada *issues #50* (`item.endswith(".jpeg")`) sebuah fungsi untuk memberitahu bahwa hanya akan dieksekusi jika file tersebut memiliki ekstensi `.jpeg`

Chapter 6

Pertemuan 6

6.1 Issues #51

Pada *issues #51* (*Method cocoNamesLoad*) sebuah fungsi untuk mengeloaad sebuah data dari file `coco.names`

```
def cocoNamesLoad(self):
    listClass = []

    with open("coco.names", "r") as daftarNama:
        for i in daftarNama.readlines():
            cocoNames = i.strip()
            listClass.append(cocoNames)

    return listClass
```

6.2 Issues #52

Pada *issues #52* (*with open("coco.names", "r") as daftarNama*) fungsi ini dijalankan untuk mengeloaad file file "coco.names" dan kita hanya dalam mode "r" yaitu membaca, dan kita masukkan ke variabel `daftarNama`.

6.3 Issues #53

Pada *issues #53* (*listClass = []*) sebuah variabel list yang akan menampung data data dengan cara `listClass.append(data)`

6.4 Issues #54

Pada *issues #54* (*Method loadYolo*) sebuah fungsi untuk mengeloa sebuah data dari yolo dan memproses gambar inputan sehingga menghasilkan list object yang ada pada gambar inputan.

```
def loadYolo(self, coconames, fileName):
    model = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")

    layerNames = model.getLayerNames()

    outputLayer = []

    for i in model.getUnconnectedOutLayers():
        outputLayer.append(layerNames[i[0] - 1])

    path = r"C:\Users\trian\Downloads"
    nameFile = fileName + ".jpeg"

    result = os.path.join(path, nameFile)

    img = cv2.imread(result)

    width, height, channels = img.shape

    blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)

    model.setInput(blob)
    outs = model.forward(outputLayer)

    boxes = []
    class_ids = []
    confidences = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
```



```

confidence = scores[class_id]

if confidence > 0.5:
    center_x = int(detection[0] * width)
    center_y = int(detection[1] * height)

    w = int(detection[2] * width)
    h = int(detection[3] * height)

    x = int(center_x - w / 2)
    y = int(center_y - h / 2)

    boxes.append([x, y, w, h])
    class_ids.append(class_id)
    confidences.append(float(confidence))

objectNames = []
for i in range(len(boxes)):
    label = coconames[class_ids[i]]

    if label in objectNames:
        print("sudah ada")
    else:
        objectNames.append(label)

return objectNames

```

6.5 Issues #55

Pada *issues #55* (*model = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")*) kode ini digunakan untuk membaca data set yolo dan konfigurasi neural network dari yolo.

6.6 Issues #56

Pada *issues #56* (*img = cv2.imread(result)*) kode ini digunakan untuk menampilkan gambar dengan *result = path foto*.

6.7 Issues #57

Pada *issues #57 (Redundant Object)* ketika ada 2 object yang sama dalam 1 foto maka yolo akan memproses 2 kali dan output 2 kali sehingga ketika kita masukkan ke list maka akan ada 2 string berobject sama. Maka dari itu kode ini digunakan untuk mengecek apakah sudah ada string tersebut pada list yang sudah berisi string dari output object yang diproses. Jika sudah ada maka string tersebut tidak akan di masukkan ke dalam list.

```
if label in objectNames:
    print("sudah ada")
else:
    objectNames.append(label)
```

6.8 Issues #58

Pada *issues #58 (if "yolo" in self.message)* perngecekan ini dilakukan jika ada kata yolo yang user kirimkan ke chatbot maka fungsi yolo akan dijalankan.

6.9 Issues #59

Pada *issues #59 (Code to Run Yolo)* kode untuk menjalankan yolo.

```
objectNames = self.listToString(self.loadYolo(self.cocoNamesLoad(), name))
```

6.10 Issues #60

Pada *issues #60 (ChatBot Type the Reply Message)* kode yang digunakan untuk membalas pesan user bahwa pada foto yang terakhir dikirim ada object apa saja.

```
self.typeAndSendMessage("Difoto terakhir yang dikirim ada object: " + objectNames)
```

Chapter 7

Pertemuan 7

7.1 Issues #61

Pada *issues #61* (*def faceRecognition(self, fileName):*) adalah sebuah fungsi untuk mengolah faceRecognition dengan inputan berupa path gambar.

7.2 Issues #62

Pada *issues #62* (*face_recognition.load_image_file()*) adalah sebuah fungsi untuk memload gambar.

```
angga_picture = face_recognition.load_image_file("angga.jpg")
```

7.3 Issues #63

Pada *issues #63* (*face_recognition.face_encodings()*) adalah sebuah fungsi untuk mende-code gambar kita menjadi sebuah bilangan numerik yang berada pada suatu list.

```
angga_encoding = face_recognition.face_encodings(angga_picture)[0]
```

7.4 Issues #64

Pada *issues #64* (*known_face_encodings =*

) adalah sebuah variabel yang berisi list encoding dari foto berisi muka yang telah di encode.

```
known_face_encodings = [  
    rolyy_encoding,  
    angga_encoding  
]
```

7.5 Issues #65

Pada *issues #65* (*face_recognition.face_locations()*) adalah sebuah fungsi untuk menemukan lokasi wajah (manusia) pada suatu gambar.

```
#test_image = gambar/foto
```

```
face_locations = face_recognition.face_locations(test_image)
```

7.6 Issues #66

Pada *issues #66* (*face_recognition.compare_faces()*) adalah untuk membandingkan 2 wajah yang sudah diencoding apakah sama atau tidak dengan parameter tambahan yaitu tolerance.

```
results = face_recognition.compare_faces(known_face_encodings, face_encoding, tol
```

7.7 Issues #67

Pada *issues #67* (*tolerance=0.45*) adalah sebuah toleransi perbandingan dengan wajah yang kita bandingkan, jika range untuk toleransi dari 0 sampai dengan 1 semakin besar toleransi maka kemungkinan wajah yang sama akan semakin besar (tidak akurat). Semakin kecil toleransi (sangat akurat) tetapi terlalu kecil akan menyebabkan wajah tidak ditemukan dikarenakan keketatan dalam pemilihan wajah dengan tingkat toleransi yang kecil.

7.8 Issues #68

Pada *issues #68* (*if "face" in self.message:*) untuk mengecek apakah ada kata face dalam chat yang dikirimkan user kepada chatbot.

7.9 Issues #69

Pada *issues #69* (*zip()*) digunakan untuk menggabungkan 2 list menjadi 1 dengan aturan penyesuaian list yang ada jika list 0 maka akan di gabungkan dengan list yang bernilai 0 juga. contoh,

#In

```
a = [1, 2]
```

```
b = [a, b]
```

```
zip(a, b)
```

#Out

```
[(1,a), (2,b)]
```

7.10 Issues #70

Pada *issues #70* (*unsupported operand type(s) for -: 'list' and 'tuple'*) disebabkan dikarenakan perbedaan data antara yang satu list dengan yang satu lagi bertipe data tuple. Pemecahan masalahnya adalah dengan cara mengecek isi kedua datanya.

Chapter 8

Pertemuan 8

8.1 Issues #71

Pada *issues #71 (getcoordinate changed variable names)* adalah pergantian variabel dari asalnya "abc" menjadi "coordinate".

8.2 Issues #72

Pada *issues #72* adalah (*change variable name from itungan to count*) yang akan diisi tipe data *integer*.

8.3 Issues #73

Pada *issues #73* adalah (*change variable name from waduwek to splitting*) yang akan diisi dengan tipe data *list*.

8.4 Issues #74

Pada *issues #74* adalah (*change variable name from abc to getNumberOfOrder*) yang akan diisi dengan angka *order* pada fitur order perhutani.

8.5 Issues #75

Pada *issues #75 (change variable name from ask to tableDataofOrder)* pergantian variabel untuk menampung nilai *list* dari kode angka orderan diwebsite perhutani.

8.6 Issues #76

Pada *issues #76* (*change variable name from nontonBioskop to keyWatch*) yang berisi tipe data *list* dari *rule* yang sudah ditentukan.

8.7 Issues #77

Pada *issues #77* (*change variable name from namaKota to cityName*) yang berisi tipe data *list* dari nama kota bioskop.

8.8 Issues #78

Pada *issues #78* (*change variable name form namaLokasi to locationName*) yang berisi tipe data *list* dari nama lokasi bioskop.

8.9 Issues #79

Pada *issues #79* (*change variable name from namaBioskop to cinemaName*) yang berisi tipe data *list* dari nama bioskop.

8.10 Issues #80

Pada *issues #80* (*translate from indonesian to english*) adalah menerjemahkan dari Bahasa Indonesia ke Bahasa Inggris.