

## **LAPORAN TUGAS RANCANG KECERDASAN BUATAN H**

*Program Penerapan Metode Klasifikasi untuk Mengenali atau Mendeteksi Object  
di dalam Gambar, Video dan WebCam menggunakan OpenCV Python dengan  
Metode Yolo*



Ditulis Oleh :

Mario Indra Widaja H	672020013
Krisna Adi Saputra	672020178
Alief Yuwastika Firmandicky	672020316

**FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN SATYA WACANA  
SALATIGA**

**2022**

**BAB 1**

**Pendahuluan**

**1. Latar Belakang**

Computer vision merupakan salah satu pengembangan teknologi informasi yang sering dijumpai dalam kehidupan kita sehari hari dan dalam semua bentuk aktivitas. Salah satu bentuk contohnya adalah “scan barcode” , dimana kode dari sebuah barcode dapat dibaca hanya dengan dilihatkan pada scanner sehingga akan muncul data dan informasi yang termuat dalam barcode. Contoh yang sering ditemui saat di mini market, sering dijumpai barcode yang terdapat di setiap label makanan atau minuman , dimana label tersebut menyimpan data informasi mengenai harga dan data dari makanan atau minuman tersebut.

Dengan jaman yang semakin berkembang, muncul konsep teknologi yang lebih cepat dalam membaca dan menyimpan data dan informasi dan ditampilkan dalam berbagai media. Contoh dari penerapan computer vision hampir berlaku pada semua bidang kehidupan seperti pada bidang matematika yang menjadi bukti kecepatan pembacaan data sekaligus data diolah menjadi hasil akhir, lalu pada pengolahan di bidang kecerdasan buatan / *artificial intelligence* dimana sebuah robot mampu melakukan perintah dan menjadikannya sebagai dasar Gerakan mengikuti alur data pada objek data. Kemudian computer vision juga digunakan pada bidang seperti industri , medis, otomotif , transportasi dan sebagainya.

## **2. Tujuan**

Tujuan dari pembuatan Tugas Rancang ini adalah membuat suatu program dalam sebuah mesin/komputer yang mampu membuktikan bahwa komputer dapat mengelola dan mengidentifikasi objek dalam suatu media gambar maupun video menggunakan Yolo V3 dan opencv serta mampu membuktikan bahwa hasil output dari identifikasi komputer sesuai dengan objek sebenarnya.

## **BAB II**

### **Landasan Teori**

#### **1. Kecerdasan Buatan**

Kecerdasan buatan atau Artificial Intelligence (AI) merupakan simulasi dari kecerdasan yang dimiliki oleh manusia dengan model di mesin dan diprogram agar bisa berpikir seperti manusia. Menurut Mc Leod dan Schell, kecerdasan buatan adalah aktivitas penyediaan mesin seperti komputer dengan kemampuan untuk menampilkan perilaku yang dianggap sama cerdasnya dengan jika kemampuan tersebut ditampilkan oleh manusia. Dapat disimpulkan bahwa kecerdasan buatan merupakan sistem komputer yang bisa melakukan pekerjaan-pekerjaan yang umumnya memerlukan tenaga manusia atau kecerdasan manusia untuk menyelesaikan pekerjaan tersebut dengan data untuk dijadikan sumber pengetahuan.

Dalam kecerdasan buatan dibutuhkan pengalaman dan data supaya kecerdasannya bisa lebih baik lagi. Poin penting dalam proses AI adalah *learning*, *reasoning* dan *self correction*. Dimana kecerdasan buatan tidak perlu selalu disuruh untuk belajar oleh manusia , melainkan kecerdasan buatan akan belajar dengan mandiri berdasarkan pengalaman saat digunakan manusia. Semakin banyak pengetahuan dan pengalaman yang didapatkan maka akan semakin baik dan akurat kecerdasan buatan terebut.

#### **2. Computer Vision**

Computer Vision merupakan kemampuan dari pengembangan teknologi informasi dengan membaca data melalui objek dengan perangkat terhubung, lalu

akan memunculkan perintah maupun fungsi tertentu akan suatu aktivitas sesuai data objek.

Dengan adanya computer vision , kehidupan semakin dipermudah karena objek yang dilihat dapat lebih akurat dan jelas, contohnya kamera , ketika melihat sasaran objek yang dituju, dapat menampilkan secara detail bentuknya. Lalu kualitas gambar yang ditangkap dapat ditingkatkan kualitasnya supaya terlihat jernih dan rapi. Selain itu penggunaan dari computer vision dalam kehidupan sehari hari memberikan kemudahan dalam menganalisis suatu objek serta dapat dijadikan suatu data akan objek tersebut.

### **3. Deteksi Objek**

Ketika kita melihat gambar atau video, kita dapat dengan mudah menemukan dan mengidentifikasi objek yang kita amati dalam beberapa saat. Deteksi objek meneruskan kecerdasan yang diamati ke komputer untuk diidentifikasi. Dalam aktivitas sehari - hari , deteksi objek sering digunakan pada saat pengawasan video CCTV , pengambilan gambar, kamera belakang atau samping mobil dan lainnya . Pastinya untuk mendeteksi objek dibutuhkan berbagai macam algoritma, dan pada kesempatan ini kelompok kami akan mendalami algoritma Yolo V3.

### **4. Algoritma Yolo V3**

Kata “Yolo” memiliki singkatan “ You Only Look Once” merupakan salah satu algoritma untuk mendeteksi objek real-time tercepat (45 frame per detik) dibandingkan dengan keluarga R-CNN seperti R-CNN, Fast R-CNN, Faster R-CNN, dan sebagainya.

Perbedaan algoritma R-CNN dengan Yolo terletak pada pendekatannya dimana algoritma R-CNN menggunakan wilayah untuk melokalisasi objek dalam gambar yang berarti model diterapkan ke beberapa wilayah , objek akan terdeteksi apabila memiliki wilayah skor tinggi. Sedangkan algoritma Yolo mengikuti pendekatan yang sama sekali berbeda dengan menerapkan jaringan syaraf ke seluruh gambar untuk memprediksi kotak pembatas dan probabilitasnya.

Pendekatan algoritma Yolo dalam mendeteksi objek terbagi menjadi dua opsi , opsi pertama dengan menggunakan model pra-terlatih dan opsi kedua dengan melatih detektor objek khusus dari awal. Dalam tugas rancang ini , kelompok kami akan melihat pembuatan detektor objek dengan membahas semua langkah proses dimulai dari anotasi data hingga akhir.

## **BAB III**

### **Pembahasan**

#### **1. Penjelasan Numpy dan OpenCV**

##### **A. Numpy**

Numpy merupakan singkatan dari Numerical Python. NumPy adalah salah satu library Python berguna untuk proses komputasi numerik. NumPy memiliki kemampuan untuk membuat objek N-dimensi array. Dilengkapi dengan sumber daya yang berguna untuk membantu data scientist mengubah python menjadi alat analisis dan pemodelan ilmiah yang kuat.

Selain itu NumPy juga digunakan sebagai wadah multi-dimensi serbaguna untuk berbagai jenis data generik. NumPy terintegrasi dengan bahasa pemrograman lain seperti C/C++, fleksibel perpustakaan NumPy memungkinkan untuk mudah dan cepat bergabung dengan berbagai database dan tools.

Yang biasa digunakan adalah NumPy Array, NumPy Array mempunyai keunggulan dari pada list pada Python adalah konsumsi memory yang lebih kecil dan runtime yang lebih cepat. NumPy juga memudahkan Aljabar Linier, terutama operasi pada Vector dan Matriks.

##### **B. OpenCV**

OpenCV(Open Sourch Computer Vision Library) adalah library open source dikembangkan oleh intel berfokus untuk menyederhanakan programing citra digital. OPenCV mempunyai beberapa fitur seperti : pengenalan wajah, pelacakan wajah, dan jenis metode AI. Dan menyediakan algoritma sederhana terkait Computer Vision untuk low level API. OpenCV termasuk open source computer vision library untuk bahasa pemrograman C/C++, dan telah dikembangkan ke python, java, matlab.

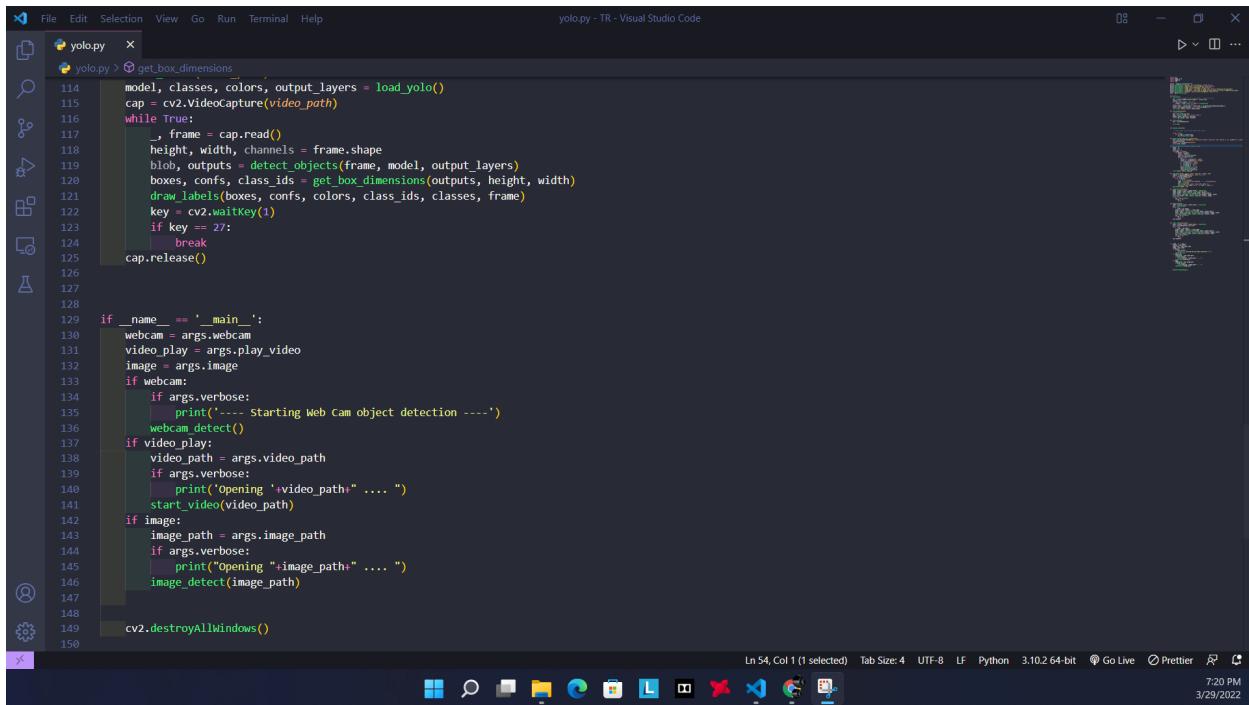
## 2. Desain Program

```
File Edit Selection View Go Run Terminal Help
yolo.py > ...
1 import cv2
2 import numpy as np
3 import argparse
4 import time
5
6 parser = argparse.ArgumentParser()
7 parser.add_argument('--webcam', help="True/False", default=False)
8 parser.add_argument('--play_video', help="True/False", default=False)
9 parser.add_argument('--image', help="True/False", default=False)
10 parser.add_argument('--video_path', help="Path of video file", default="videos/car_on_road.mp4")
11 parser.add_argument('--image_path', help="Path of image to detect objects", default="Images/bicycle.jpg")
12 parser.add_argument('--verbose', help="To print statements", default=True)
13 args = parser.parse_args()
14
15 #Load yolo
16 def load_yolo():
17     net = cv2.dnn.readNet("yolov3-tiny.weights", "yolov3-tiny.cfg")
18     net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
19     classes = []
20     with open("coco.names", "r") as f:
21         classes = [line.strip() for line in f.readlines()]
22     layer_names = net.getLayerNames()
23     output_layers = [layer_name for layer_name in net.getUnconnectedOutLayersNames()]
24     colors = np.random.uniform(0, 255, size=(len(classes), 3))
25     return net, classes, colors, output_layers
26
27 def load_image(img_path):
28     # image loading
29     img = cv2.imread(img_path)
30     img = cv2.resize(img, None, fx=0.4, fy=0.4)
31     height, width, channels = img.shape
32     return img, height, width, channels
33
34 def start_webcam():
35     cap = cv2.VideoCapture(0)
36
37     return cap
Ln 12, Col 50 Tab Size: 4 UTF-8 LF Python 3.10.2 64-bit ⚡ Go Live ⚡ Prettier ⚡ 7:16 PM 3/29/2022
```

```
File Edit Selection View Go Run Terminal Help
yolo.py > ...
38
39
40 def display_blob(blob):
41     """
42         Three images each for RED, GREEN, BLUE channel
43     """
44     for b in blob:
45         for n, imgh in enumerate(b):
46             cv2.imshow(str(n), imgh)
47
48 def detect_objects(img, net, outputLayers):
49     blob = cv2.dnn.blobFromImage(img, scaleFactor=0.00392, size=(320, 320), mean=(0, 0, 0), swapRB=True, crop=False)
50     net.setInput(blob)
51     outputs = net.forward(outputLayers)
52     return blob, outputs
53
54 def get_box_dimensions(outputs, height, width):
55     boxes = []
56     confs = []
57     class_ids = []
58     for output in outputs:
59         for detect in output:
60             scores = detect[5:]
61             class_id = np.argmax(scores)
62             conf = scores[class_id]
63             if conf > 0.3:
64                 center_x = int(detect[0] * width)
65                 center_y = int(detect[1] * height)
66                 w = int(detect[2] * width)
67                 h = int(detect[3] * height)
68                 x = int(center_x - w/2)
69                 y = int(center_y - h / 2)
70                 boxes.append([x, y, w, h])
71                 confs.append(float(conf))
72                 class_ids.append(class_id)
73
74     return boxes, confs, class_ids
Ln 12, Col 50 Tab Size: 4 UTF-8 LF Python 3.10.2 64-bit ⚡ Go Live ⚡ Prettier ⚡ 7:17 PM 3/29/2022
```

```
File Edit Selection View Go Run Terminal Help
yolo.py - TR - Visual Studio Code
yolo.py > get_box_dimensions
74
75 def draw_labels(boxes, confs, colors, class_ids, classes, img):
76     indexes = cv2.dnn.NMSBoxes(boxes, confs, 0.5, 0.4)
77     font = cv2.FONT_HERSHEY_PLAIN
78     for i in range(len(boxes)):
79         if i in indexes:
80             x, y, w, h = boxes[i]
81             label = str(classes[class_ids[i]]) + " " + str(confs)[1:5]
82             color = colors[i]
83             cv2.rectangle(img, (x,y), (x+w, y+h), color, 2)
84             cv2.putText(img, label, (x, y - 5), font, 1, color, 1)
85     cv2.imshow("Image", img)
86
87 def image_detect(img_path):
88     model, classes, colors, output_layers = load_yolo()
89     image, height, width, channels = load_image(img_path)
90     blob, outputs = detect_objects(image, model, output_layers)
91     boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
92     draw_labels(boxes, confs, colors, class_ids, classes, image)
93     while True:
94         key = cv2.waitKey(1)
95         if key == 27:
96             break
97
98 def webcam_detect():
99     model, classes, colors, output_layers = load_yolo()
100    cap = start_webcam()
101    while True:
102        _, frame = cap.read()
103        height, width, channels = frame.shape
104        blob, outputs = detect_objects(frame, model, output_layers)
105        boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
106        draw_labels(boxes, confs, colors, class_ids, classes, frame)
107        key = cv2.waitKey(1)
108        if key == 27:
109            break
110    cap.release()
111
112
113 def start_video(video_path):
114     model, classes, colors, output_layers = load_yolo()
115     cap = cv2.VideoCapture(video_path)
116     while True:
117         _, frame = cap.read()
118         height, width, channels = frame.shape
119         blob, outputs = detect_objects(frame, model, output_layers)
120         boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
121         draw_labels(boxes, confs, colors, class_ids, classes, frame)
122         key = cv2.waitKey(1)
123         if key == 27:
124             break
125     cap.release()
126
127
128 if __name__ == '__main__':
129     webcam = args.webcam
130     video_play = args.play_video
131     image = args.image
132     if webcam:
133         if args.verbose:
134             print('---- Starting Web Cam object detection ----')
135         webcam.detect()
136     if video_play:
137         video_path = args.video_path
138         if args.verbose:
139             print('Opening ' + video_path + '....')
140         start_video(video_path)
141     if image:
142         image_path = args.image_path
143         if args.verbose:
144             print('Opening ' + image_path + '....')
145         image_detect(image_path)
146
147
Ln 54, Col 1 (1 selected) Tab Size: 4 UTF-8 LF Python 3.10.2 64-bit Go Live Prettier 7:18 PM 3/29/2022
```

```
File Edit Selection View Go Run Terminal Help
yolo.py - TR - Visual Studio Code
yolo.py > get_box_dimensions
110
111
112
113 def start_video(video_path):
114     model, classes, colors, output_layers = load_yolo()
115     cap = cv2.VideoCapture(video_path)
116     while True:
117         _, frame = cap.read()
118         height, width, channels = frame.shape
119         blob, outputs = detect_objects(frame, model, output_layers)
120         boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
121         draw_labels(boxes, confs, colors, class_ids, classes, frame)
122         key = cv2.waitKey(1)
123         if key == 27:
124             break
125     cap.release()
126
127
128 if __name__ == '__main__':
129     webcam = args.webcam
130     video_play = args.play_video
131     image = args.image
132     if webcam:
133         if args.verbose:
134             print('---- Starting Web Cam object detection ----')
135         webcam.detect()
136     if video_play:
137         video_path = args.video_path
138         if args.verbose:
139             print('Opening ' + video_path + '....')
140         start_video(video_path)
141     if image:
142         image_path = args.image_path
143         if args.verbose:
144             print('Opening ' + image_path + '....')
145         image_detect(image_path)
146
147
Ln 54, Col 1 (1 selected) Tab Size: 4 UTF-8 LF Python 3.10.2 64-bit Go Live Prettier 7:19 PM 3/29/2022
```



```
File Edit Selection View Go Run Terminal Help
yolo.py - TR - Visual Studio Code
yolo.py > get_box_dimensions
114     model, classes, colors, output_layers = load_yolo()
115     cap = cv2.VideoCapture(video_path)
116     while True:
117         _, frame = cap.read()
118         height, width, channels = frame.shape
119         blob, outputs = detect_objects(frame, model, output_layers)
120         boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
121         draw_labels(boxes, confs, colors, class_ids, classes, frame)
122         key = cv2.waitKey(1)
123         if key == 27:
124             break
125     cap.release()
126
127
128 if __name__ == '__main__':
129     webcam = args.webcam
130     video_play = args.play_video
131     image = args.image
132     if webcam:
133         if args.verbose:
134             print('----- Starting Web Cam object detection -----')
135             webcam_detect()
136     if video_play:
137         video_path = args.video_path
138         if args.verbose:
139             print('Opening ' + video_path + ' .... ')
140             start_video(video_path)
141     if image:
142         image_path = args.image_path
143         if args.verbose:
144             print('Opening ' + image_path + ' .... ')
145             image_detect(image_path)
146
147
148
149     cv2.destroyAllWindows()
150
```

Ln 54, Col 1 (1 selected) Tab Size: 4 UTF-8 LF Python 3.10.2 64-bit ⚙ Go Live ⚙ Prettier ⚙ 7:20 PM 3/29/2022

### 3. Analisa dan Diskusi terkait dengan performa Implementasi Program.

Kami akan menunjukkan dan menjelaskan beberapa program atau tools yang harus diinstal sebelum melakukan program untuk objek image detection, video detection, dan webcam.

Berikut Programnya yang harus dilakukan terlebih dahulu:

```
1 import cv2
2 import numpy as np
3 import argparse
4 import time
5
```

Code diatas bisa disebut sebagai Module. Yang harus diinstal pertama kali adalah OpenCV dan numpy yang dapat dilakukan melalui cmd (Command Prompt). Pada baris pertama untuk mengimpor OpenCV, baris kedua untuk mengimpor numpy, baris ketiga untuk mengimport argparse untuk video, dan yang terakhir baris keempat untuk mengimport time atau waktu untuk deteksi objek.

```
#Load yolo
def load_yolo():
    net = cv2.dnn.readNet("yolov3-tiny.weights", "yolov3-tiny.cfg")
    net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
    classes = []
    with open("coco.names", "r") as f:
        classes = [line.strip() for line in f.readlines()]
    #layers_names = net.getLayerNames()
    output_layers = [layer_name for layer_name in net.getUnconnectedOutLayersNames()]
    colors = np.random.uniform(0, 255, size=(len(classes), 3))
    return net, classes, colors, output_layers
```

Lalu ada Algoritma Load Yolo def load\_yolo(). Terdapat net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg") untuk mengakses dan membuat bobot di file yolov3, file **coco.names** berisi nama-nama objek berbeda yang telah dlatih. Lalu kami menyimpan nya dalam list yang disebut classes. Untuk melanjutkan forward pass menggunakan modul cv2, perlu memasukkan nama layer yang outputnya akan dihitung. net.getUnconnectedOutLayersNames() mengembalikan indeks lapisan keluaran jaringan.

```
def load_image(img_path):
    # image loading
    img = cv2.imread(img_path)
    img = cv2.resize(img, None, fx=1.0, fy=1.0)
    height, width, channels = img.shape
    return img, height, width, channels
```

Lalu pada Algoritma diatas berguna untuk menerima jalur gambar sebagai parameter, membaca gambar, mengubah ukurannya, dan mengembalikannya. Contoh pada img = cv2.resize(img, None, fx=1.0, fy=1.0) berfungsi untuk mengubah ukuran pada fx dan fy.

Untuk memprediksi objek dengan jaringan saraf, perlu memproses data dan modul cv2. Memberi dua fungsi untuk tujuan ini : blobFromImage dan blobFromImages. Fungsi-fungsi ini melakukan penskalaan, pengurangan rata-rata, dan pertukaran saluran yang opsional.

```

def detect_objects(img, net, outputLayers):
    blob = cv2.dnn.blobFromImage(img, scaleFactor=0.00392, size=(320, 320), mean=(0, 0, 0), swapRB=True, crop=False)
    net.setInput(blob)
    outputs = net.forward(outputLayers)
    return blob, outputs

```

Disini terdapat `detect_objects` menggunakan `blobFromImage` berfungsi untuk menerima gambar atau bingkai dari aliran video atau webcam, model dan lapisan keluaran sebagai parameter. Dapat dilihat pada `scalefactor=0.00392` menggunakan **skala 0.00392**, Oleh karena itu dapat menskalakan piksel gambar ke kisaran 0 hingga 1 dan mengurnya ke nilai [0,0,0]. Fungsi `forward()` dari modul cv2 mengembalikan daftar bersarang yang berisi informasi tentang semua objek yang terdeteksi yang mencakup koordinat x dan y dari pusat objek yang terdeteksi, tinggi dan lebar kotak pembatas, keyakinan dan skor untuk semua kelas objek yang terdaftar di `coco.names`. Kelas dengan skor tertinggi dianggap sebagai kelas prediksi.

```

def get_box_dimensions(outputs, height, width):
    boxes = []
    confs = []
    class_ids = []
    for output in outputs:
        for detect in output:
            scores = detect[5:]
            class_id = np.argmax(scores)
            conf = scores[class_id]
            if conf > 0.3:
                center_x = int(detect[0] * width)
                center_y = int(detect[1] * height)
                w = int(detect[2] * width)
                h = int(detect[3] * height)
                x = int(center_x - w/2)
                y = int(center_y - h / 2)
                boxes.append([x, y, w, h])
                confs.append(float(conf))
                class_ids.append(class_id)
    return boxes, confs, class_ids

```

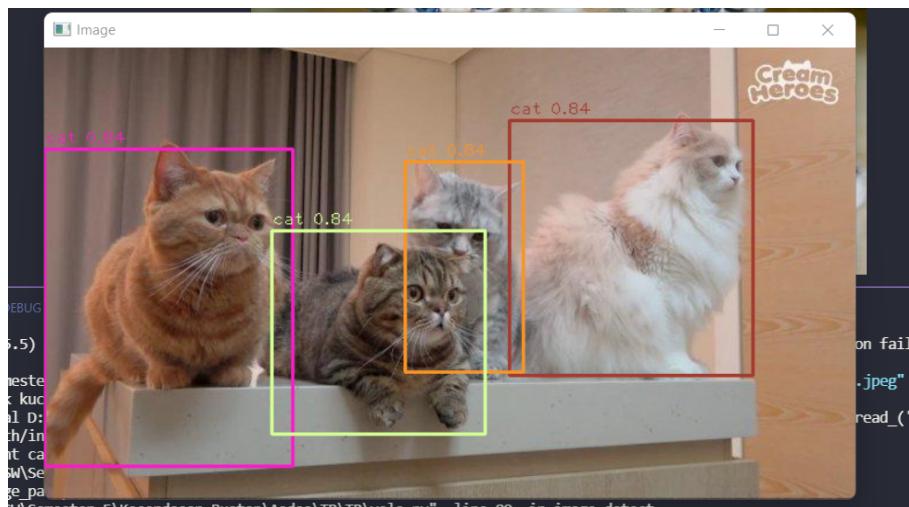
Dalam fungsi `get_box-dimensions()` urutan yang disebut `scores` dibuat menyimpan yang sesuai dengan setiap objek. Selanjutnya mengidentifikasi indeks `class` dengan skor tertinggi menggunakan `np.argmax()` sehingga mendapatkan

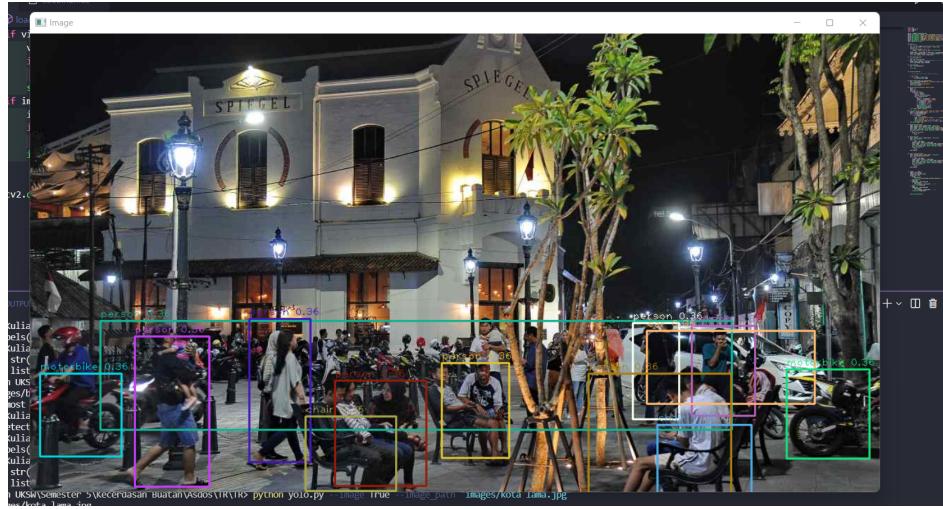
nama kelas yang sesuai dengan indeks dari daftar `classs` yang telah dibuat di `load_yolo()`. Setelah memilih kotak yang diprediksi keyakinan 0.3 atau 30%, angka ini bisa diganti sesuka hati namun semakin besar angka tersebut makan kotak tidak muncul.

```
def draw_labels(boxes, confs, colors, class_ids, classes, img):
    indexes = cv2.dnn.NMSBoxes(boxes, confs, 0.5, 0.4)
    font = cv2.FONT_HERSHEY_PLAIN
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]]) + " " + str(confs)[1:5]
            color = colors[i]
            cv2.rectangle(img, (x,y), (x+w, y+h), color, 2)
            cv2.putText(img, label, (x, y - 5), font, 1, color, 1)
    cv2.imshow("Image", img)
```

Fungsi `draw_labels()` ini untuk menggambar kotak pembatas dan menambahkan label objek ke dalamnya. Memiliki simpul dari kotak pembatas yang diprediksi dan `class_id`. Untuk menghilangkan beberapa kotak yang double, dapat memberikan nilai ambang dan batas ambang sebagai parameter untuk memilih satu kotak pembatas. Dari kisaran 0 hingga 1, kami memilih nilai tengah seperti 0,4 atau 0,5 seperti pada codingan diatas untuk memastikan mendeteksi objek tumpah tindih atau double tetapi tidak mendapatkan banyak kotak pembatas untuk objek yang sama. Fungsi ini dapat disatukan dengan fungsi yang lain seperti fungsi `image detection`, dan `video detection` yang akan dijelaskan selanjutnya.

#### A. Image Detection





Pada gambar di atas melakukan pendektsian objek, dimana objek tersebut yang berada di gambar terdapat terdetksi oleh sistem yang dimana data objek tersebut telah menyimpan beberapa nama barang atau benda di dalam suatu file sehingga sistem menyesuaikan objek yang ada di gambar dengan nama-nama objek yang ada di dalam file. Dan pendektsian tersebut tidak ada kesalahan sehingga pendektsian dapat bilang sempurna.

Berikut program Image Detection :

```
def image_detect(img_path):
    model, classes, colors, output_layers = load_yolo()
    image, height, width, channels = load_image(img_path)
    blob, outputs = detect_objects(image, model, output_layers)
    boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
    draw_labels(boxes, confs, colors, class_ids, classes, image)
    while True:
        key = cv2.waitKey(1)
        if key == 27:
            break
```

Didalam fungsi `image_detect()` dapat parameter `img_path` yang berfungsi untuk memanggil gambar yang ingin di detek objek, selain itu dalam parameter ini terdapat fungsi-fungsi yang lainnya untuk menjalankan dan mendetek objek pada gambar, yang sudah dijelaskan di atas.

```

parser = argparse.ArgumentParser()
parser.add_argument('--webcam', help="True/False", default=False)
parser.add_argument('--play_video', help="True/False", default=False)
parser.add_argument('--image', help="True/False", default=False) ←
parser.add_argument('--video_path', help="Path of video file", default="videos/car_on_road.mp4")
parser.add_argument('--image_path', help="Path of image to detect objects", default="Images/bicycle.jpg") ←
parser.add_argument('--verbose', help="To print statements", default=True)
args = parser.parse_args()

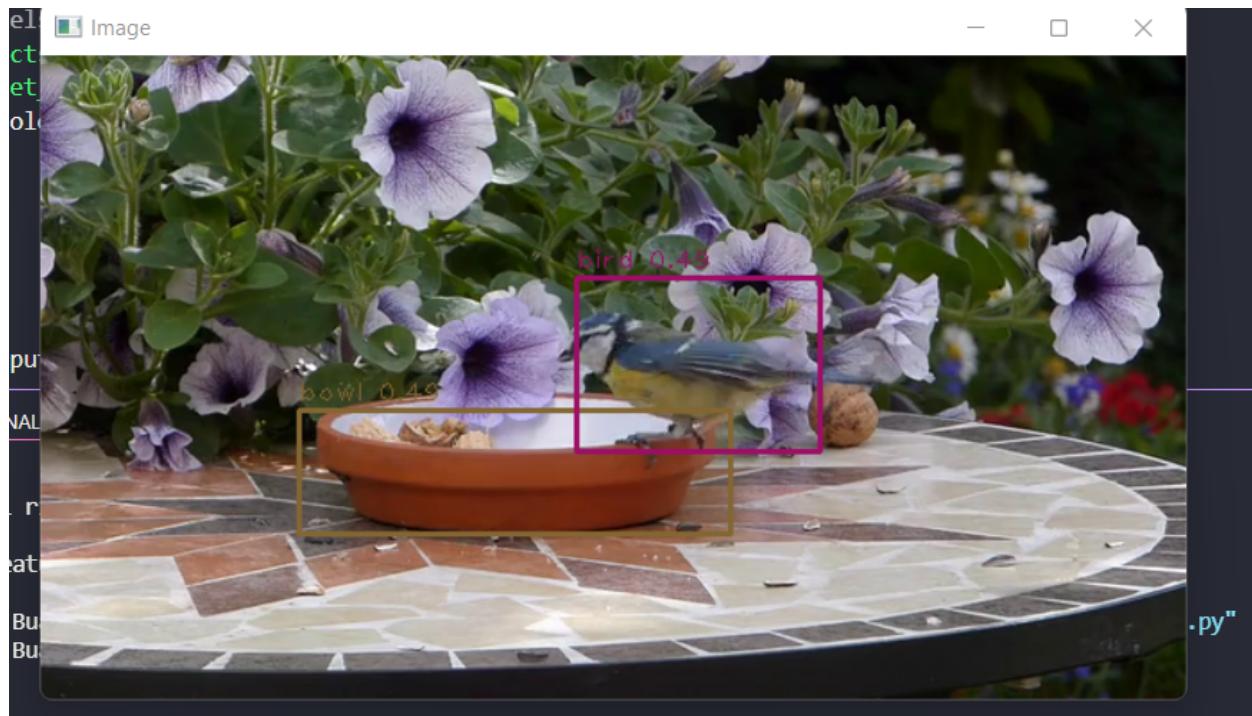
```

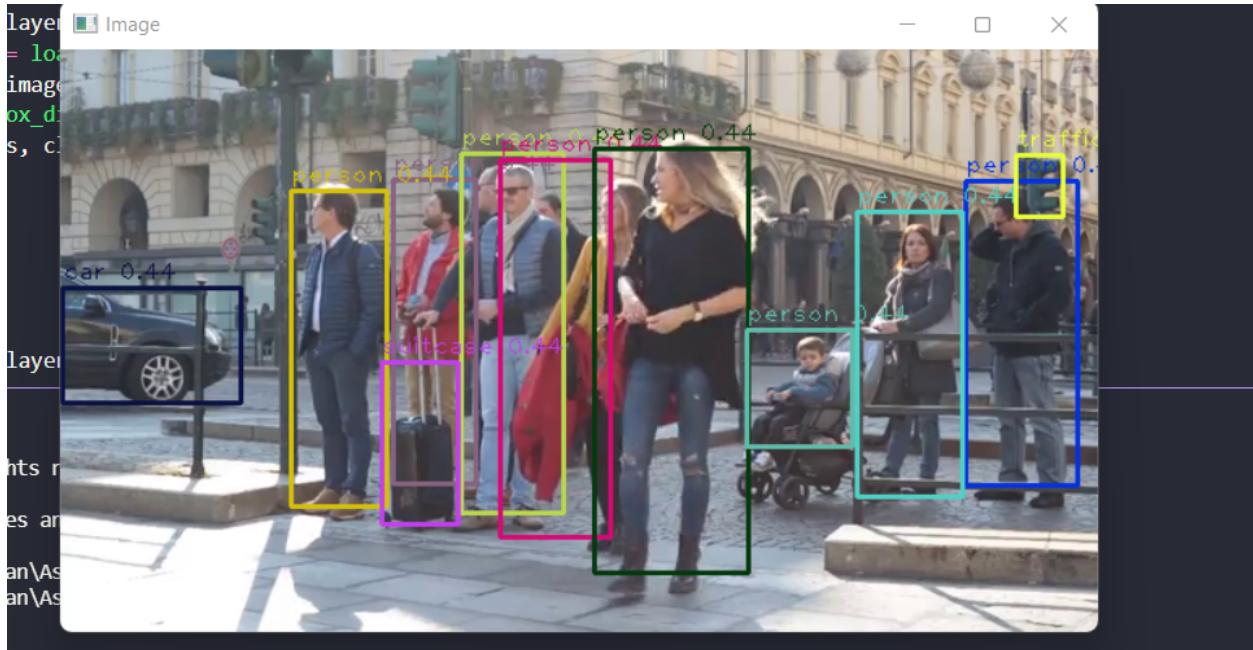
Pada anak panah merah tersebut menunjukkan command yang digunakan untuk memanggil saat di terminal namun harus berada lokasi folder python berada. Penulisannya seperti ini :

`python nama_file.py --image nilai_boolean(True) --image_path "folder/namafile.format"`

```
PS D:\Kuliah UKSW\Semester 5\Kecerdasan Buatan\Asdos\TR\TR> python yolo.py --image True --image_path "images/kota lama.jpg"
```

## B. Video Detection





```
def start_video(video_path):
    model, classes, colors, output_layers = load_yolo()
    cap = cv2.VideoCapture(video_path)
    while True:
        _, frame = cap.read()
        height, width, channels = frame.shape
        blob, outputs = detect_objects(frame, model, output_layers)
        boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
        draw_labels(boxes, confs, colors, class_ids, classes, frame)
        key = cv2.waitKey(1)
        if key == 27:
            break
    cap.release()
```

Didalam fungsi `start_video()` terdapat parameter `video_path` yang berfungsi untuk memanggil video yang ingin di detek objek, selain itu dalam parameter ini terdapat fungsi-fungsi yang lainnya untuk menjalankan dan mendetek objek pada video, yang sudah dijelaskan di atas.

```

parser = argparse.ArgumentParser()
parser.add_argument('--webcam', help="True/False", default=False)
parser.add_argument('--play_video', help="True/False", default=False) ←
parser.add_argument('--image', help="True/False", default=False)
parser.add_argument('--video_path', help="Path of video file", default="videos/car_on_road.mp4") ←
parser.add_argument('--image_path', help="Path of image to detect objects", default="Images/bicycle.jpg")
parser.add_argument('--verbose', help="To print statements", default=True)
args = parser.parse_args()

```

Pada anak panah merah tersebut menunjukkan command yang digunakan untuk memanggil saat di terminal namun harus berada lokasi folder python berada.

Penulisannya seperti ini :

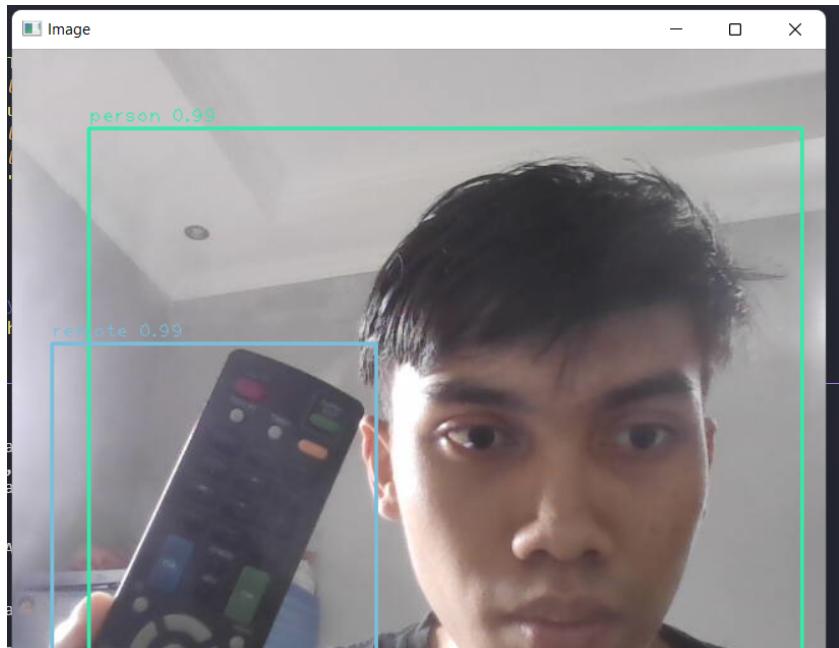
python      nama\_file.py      -play\_video      nilai\_boolean(True)      -video\_path  
 "folder/namaFile.format"

```

PS D:\Kuliah UKSW\Semester 5\Kecerdasan Buatan\Asdos\TR\TR> python -u "d:\Kuliah UKSW\Semester 5\Kecerdasan Buatan\Asdos\TR\TR\yolo.py"
PS D:\Kuliah UKSW\Semester 5\Kecerdasan Buatan\Asdos\TR\TR> python yolo.py --play_video True --video_path "videos/bird.mp4" []

```

### C. WebCam



```

def webcam_detect():
    model, classes, colors, output_layers = load_yolo()
    cap = start_webcam()
    while True:
        _, frame = cap.read()
        height, width, channels = frame.shape
        blob, outputs = detect_objects(frame, model, output_layers)
        boxes, confs, class_ids = get_box_dimensions(outputs, height, width)
        draw_labels(boxes, confs, colors, class_ids, classes, frame)
        key = cv2.waitKey(1)
        if key == 27:
            break
    cap.release()

```

Didalam fungsi `webcam_detect()` berfungsi untuk memanggil video yang ingin di detek objek, selain itu dalam parameter ini terdapat fungsi-fungsi yang lainnya untuk menjalankan dan mendetek objek pada video webcam, yang sudah dijelaskan di atas.

```

parser = argparse.ArgumentParser()
parser.add_argument('--webcam', help="True/False", default=False) ←
parser.add_argument('--play_video', help="True/False", default=False)
parser.add_argument('--image', help="True/False", default=False)
parser.add_argument('--video_path', help="Path of video file", default="videos/car_on_road.mp4")
parser.add_argument('--image_path', help="Path of image to detect objects", default="images/bicycle.jpg")
parser.add_argument('--verbose', help="To print statements", default=True)
args = parser.parse_args()

```

Pada anak panah merah tersebut menunjukkan command yang digunakan untuk memanggil saat di terminal namun harus berada lokasi folder python berada. Penulisannya seperti ini :

python nama\_file.py --webcam nama\_boolean(True)

```

PS D:\Kuliah UKSW\Semester 5\Kecerdasan Buatan\Asdos\TR\TR> python -u "d:\Kuliah UKSW\Semester 5\Kecerdasan Buatan\Asdos\TR\TR\yolo.py"
PS D:\Kuliah UKSW\Semester 5\Kecerdasan Buatan\Asdos\TR\TR> python yolo.py --webcam True

```

## **BAB IV**

### **PENUTUP**

#### **1. Kesimpulan**

Berdasarkan analisa yang telah kami rangkum, Kecerdasan buatan merupakan suatu aspek ilmu pengetahuan yang patut untuk ditekuni pada era modern ini karena penerapan program sangat luas kaitanya dengan kehidupan manusia, seperti contohnya pendekripsi objek yang merupakan suatu hal yang lumrah pada manusia kini komputer juga mampu mendeteksi dan mengklasifikasi objek pada gambar hingga sebuah video, bahkan secara real time menggunakan webcam.

Pada program kali ini penerapan metode pendekatan pembuatan detektor objek dari awal sampai akhir untuk mengenali atau mendeteksi objek di dalam gambar sukses diujikan, dimana objek yang dideteksi dapat menghasilkan informasi mengenai jenis objek yang diuji.

Penklasifikasian objek menggunakan OpenCV python dan juga YOLO ini dapat dikatakan baik walaupun masih terdapat kekurangan seperti beberapa objek dalam foto tidak terdeteksi yang dikarenakan kurangnya cahaya maupun tertutup sebagian oleh objek di depannya. Akan tetapi, Pengembangan program ini merupakan pencapaian yang baik untuk perkembangan teknologi kedepan.

## DAFTAR PUSTAKA

- Rosebroock, Adrian. 2018. “YOLO object detection with OpenCV”, <https://pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv>, diakses pada 29 Maret 2022 pukul 20.31.
- Bansal, Nandini. 2020. “Object Detection using YoloV3 and OpenCV”, <https://towardsdatascience.com/object-detection-using-yolov3-and-opencv-19ee0792a420>, diakses pada tanggal 29 Maret 2022 pukul 21.15.
- Prasetyo, Eko. 2020. “Tutorial 2 Deteksi Obyek dengan YOLO dan OpenCV” [Video], <https://www.youtube.com/watch?v=gy73C4khvgs&t=132s>, diakses pada tanggal 28 Maret 2022 pukul 20.15.
- Giovi, Fashalli. 2020. “Deteksi Objek Otomatis secara Real-Time dengan YOLO dan OpenCV” [Video], [https://www.youtube.com/watch?v=GRa40tX0U\\_s&t=1s](https://www.youtube.com/watch?v=GRa40tX0U_s&t=1s), diakses pada tanggal 28 Maret 2022 pukul 20.15.
- Firnas, I. (2021). Apa itu Computer Vision ? Pengertian, Fungsi, & Contoh. <https://ibnufirnas.com/pengertian-computer-vision/>
- Takdirillah, R. (2020). Apa Itu Kecerdasan Buatan? Berikut Pengertian dan Contohnya - Dicoding Blog. <https://www.dicoding.com/blog/kecerdasan-buatan-adalah/>
- Purwanto. (2018). 3 Library Python Terbaik Untuk Data Science. Belajarpython. <https://belajarpython.com/2018/09/3-library-python-terbaik-untuk-data-science.html>, diakses pada 30 Maret 2022 pukul 10.01.
- Uswatun, L. (2021). Python : Kenali NumPy Array dalam Python. DQLab. <https://www.dqlab.id/kenali-numpy-array-dalam-python>, diakses pada 30 Maret 2022 pukul 10.28.
- Sidharta, H. A. (2017). INTRODUCTION TO OPEN CV. Binus UNiversity. <https://binus.ac.id/malang/2017/10/introduction-to-open-cv/>, diakses pada 30 Maret 2022 pukul 11.07