DAFTAR PUSTAKA

1) Cita Aisah Nurbani,dkk, *Measurement of Cholesterol Levels through Eye based on Co-Occurrence Matrix on Android*, Telkom University Bandung, Indonesia, IEEE, 2019

2) Thu TN Dinh, dkk, *Cholesterol Levels and Methods of Determining Cholesterol in Meat and Poultry*, CRFSFS c 2011 Institute of FoodTechnologists.

3) Dra. Asmar Yulastri, M.Pd, pengembangan rumah makan masakan padang dalam menuju perlindungan hak kekayaan intelektual (hki), 2008

4) Takumi Ege Keiji Yanai,dkk, *Estimating Food Calories for Multiple-dish Food Photos*,2017 4th IAPR Asian conference on Pattern, IEEE Explore.

5) Liang, dkk, *Deep Learning-Based Food Calorie Estimation Method in Dietary Assessment*. IEEE, 2017.

6) Parth Poply, dkk, *An Instance Segmentation approach to Food Calorie Estimation using Mask R-CNN*, SPML 2020, Oktober 22–24, 2020, Beijing, China© 2020

7) Parth polly, dkk, *Image segmentation for calorie estimatioan of multiple-dish food items*,IEEE,2021

8) Yuanyuan Wang,dkk, *Combining single shot multibox detector with transfer learning for ship detection using Sentinel-1 images*, IEEE, 2017

9) Nan Ge, dkk, *Evaluation of the Two-Dimensional Temperature Field and Instability of a Dual-Jet DC Arc Plasma Based on the Image Chain Coding Technique*,IEEE,2011

10) Eduardo Aguilar, dkk, *Grab, Pay and Eat: Semantic Food Detection for Smart Restaurants*,IEEE,2020

11) Ubiratan Ramos, dkk, *Adaptable Architecture for the Development of Computer Vision Systems in FPGA*, IEEE, 2020

12) Guoyan Yu, dkk, *An adaptive dead fish detection approach using SSD-MobileNet*, IEEE, 2020.

13) Smit Trambadia, *Food Detection on Plate Based on HSV Color Model,* 2016 Online International Conference on Green Engineering and Technologies (IC-GET)

14) Takumi Ege, dkk, *Image-Based Estimation of Real Food Size for Accurate Food Calorie Estimation*,IEEE,2019.

15) Haoyu Hu, dkk, *Image Based Food Calories Estimation Using Various Models of Machine Learning*, IEEE, 2021

16) Boya, dkk, *Diseases and Pests Identification of Lycium Barbarum Using SE-MobileNet V2 Algorithm*, International Symposium on Computational Intelligence and Design (ISCID),IEEE,2020.

17) Aifian Adi Sufian Chan,Dkk. *Face Detection in Still Image using SSD MobileNet V2 and Geometrical Algorithm*, 2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST),IEEE 2023

18) Dmitri Truhachev, dkk, *Code Design Based on Connecting Spatially Coupled Graph Chains*, IEEE,2019

19) Yihuan Liao, dkk, *Connecting Spatially Coupled LDPC Code Chains for Bit-Interleaved Coded Modulation*, IEEE, 2021

20) R. Dinesh Kumar, dkk, *Recognition of food type and calorie estimation using neural network*, The Journal of Supercomputing, IEEE, 2016

21.) Gianluigi Ciocc, dkk, *Evaluating CNN-Based Semantic Food Segmentation Across Illuminants*,IEEE, 2019.

22). Cahyo Adhi Hartanto; Adi Wibowo, *Development of Mobile Skin Cancer Detection using Faster R-CNN and MobileNet v2 Model*, IEEE, 2020.

23). Annisaa' F. Nurfirdausi, *Implementation of Single Shot Detector (SSD) MobileNet V2 on Disabled Patient's Hand Gesture Recognition as a Notification System*, IEEE,2021

24). Yun Fan; Hualu Liu, *Double Constacyclic Codes Over Two Finite Commutative Chain Rings*, IEEE, 2023

ıamadhab Dalai , Kishore Kumar Senapati, *A Heuristic Grid Area Based Segmentation Approach for Weight Estimation of an Object from Image*, I2CT, 2018

LAMPIRAN

Lampiran 1. Script pada deteksi objek

```
#wajib di eksekusi, supaya data di google drive
bisa terbaca
#mount drive
from google.colab import drive
drive.mount('/content/drive')
```

```
#wajib di eksekusi, dipakai oleh script lain
sebagai shorcut ke folder MyDrive

# this creates a symbolic link so that now the
path /content/gdrive/My Drive/ is equal to
/mydrive
!ln -s /content/drive/MyDrive /drive
!ls /drive
```

```
#wajib di eksekusi, karena ini adalah model
tensorflow yang dibutuhkan untuk training hingga
ke pengenalan obyek

# clone the tensorflow models on the colab cloud
vm
!git clone --q
https://github.com/tensorflow/models.git
# navigate to /models/research folder to compile
protos
%cd models/research
# Compile protos.
!protoc object_detection/protos/*.proto --
python_out=.
# Install TensorFlow Object Detection API.
!cp object_detection/packages/tf2/setup.py .
!python -m pip install .
```

```
t/models/research
ing /content/models/research
ring metadata (setup.py) ... done
ing avro-python3 (from object-detection==0.1)
```

```
  Downloading avro-python3-1.10.2.tar.gz (38 kB)
  Preparing metadata (setup.py) ... done
Collecting apache-beam (from object-detection==0.1)
  Downloading apache_beam-2.48.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (14.3 MB)

Collecting crcmod<2.0,>=1.7 (from apache-beam->object-
detection==0.1)
  Downloading crcmod-1.7.tar.gz (89 kB)
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 6.0
    Uninstalling PyYAML-6.0:
      Successfully uninstalled PyYAML-6.0
  Attempting uninstall: pyparsing
    Found existing installation: pyparsing 3.1.0
    Uninstalling pyparsing-3.1.0:
      Successfully uninstalled pyparsing-3.1.0
  Attempting uninstall: numpy
    Found existing installation: numpy 1.22.4
    Uninstalling numpy-1.22.4:
      Successfully uninstalled numpy-1.22.4
Successfully installed apache-beam-2.48.0 avro-python3-1.10.2
colorama-0.4.6 crcmod-1.7 dill-0.3.1.1 dnspython-2.3.0 docopt-
0.6.2 fastavro-1.7.4 fasteners-0.18 hdfs-2.7.0 immutabledict-
2.2.4 lvis-0.5.3 numpy-1.23.5 object-detection-0.1 objsize-
0.6.1 orjson-3.9.1 portalocker-2.7.0 pymongo-4.4.0 pyparsing-
2.4.7 pyyaml-5.4.1 sacrebleu-2.2.0 sentencepiece-0.1.99
seqeval-1.2.2 tensorflow-addons-0.20.0 tensorflow-model-
optimization-0.7.5 tensorflow-text-2.12.1 tensorflow_io-0.32.0
tf-models-official-2.12.0 typeguard-2.13.3 zstandard-0.21.0
```

```
import pandas as pd
import numpy as np

namaFile =
pd.read_csv('/content/drive/MyDrive/customTF2/da
ta/train_labels.csv')
namaFile
```

	filename	width	height	class	xmin	ymin	xmax	ymax
0	DSC00017.jpg	1200	800	Cassava leaves	440	416	553	588
1	DSC00017.jpg	1200	800	Green chili	663	478	744	559
2	DSC00017.jpg	1200	800	Rice	533	359	708	530
3	DSC00017.jpg	1200	800	Cakedel	552	527	677	632
4	DSC00019.jpg	800	1200	Green chili	344	395	470	506
...	...	...	...	...	...	...	...	...
2274	DSC04984.jpg	1200	675	Cassava leaves	298	460	481	626
2275	DSC04985.jpg	675	1200	Rice	233	492	396	674
2276	DSC04985.jpg	675	1200	Rice	242	719	389	880
2277	DSC04985.jpg	675	1200	Rice	0	474	88	650
2278	DSC04985.jpg	675	1200	Rice	0	688	67	832

2279 rows × 8 columns

```
namaFile =
pd.read_csv('/content/drive/MyDrive/customTF2/data/test_labels.csv')
namaFile
```

	filename	width	height	class	xmin	ymin	xmax	ymax
0	DSC00408.jpg	1200	800	Rendang	567	136	832	357
1	DSC00408.jpg	1200	800	Rice	443	241	762	532
2	DSC00408.jpg	1200	800	Hot spicy	395	466	627	636
3	DSC00408.jpg	1200	800	Jackfruit vegetable	670	374	913	671
4	DSC00178.jpg	1200	800	Balado eggs	462	127	622	276
...	...	...	...	...	...	...	...	...
434	DSC00411.jpg	1200	800	Rice	437	237	735	527
435	DSC00279.jpg	1200	800	Hot spicy	480	118	662	286
436	DSC00279.jpg	1200	800	Cassava leaves	646	128	871	434
437	DSC00279.jpg	1200	800	Rice	492	238	721	494
438	DSC00279.jpg	1200	800	Goulash	579	419	830	719

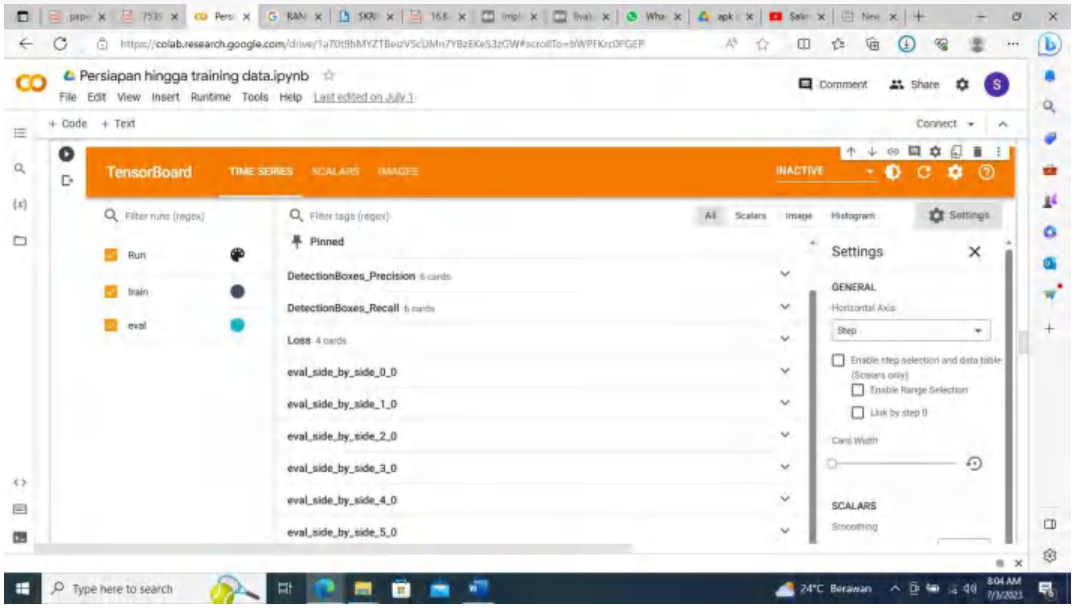
439 rows × 8 columns

```
#tidak wajib di eksekusi. perintah dibawah hanya untuk melihat apakah model tensorflow sudah berjalan sebagaimana mestinya atau belum

!python
_detection/builders/model_builder_tf2_test
```

```
#wajib dieksekusi kalo ingin melihat visualisasi
dari model yang sedang dilatih

%load_ext tensorboard
%tensorboard --logdir
'/drive/customTF2/training'
```




```
#wajib di eksekusi
%cd /content/models/research/object_detection

#Run the command below from the
content/models/research/object_detection
directory
"""
PIPELINE_CONFIG_PATH=path/to/pipeline.config
MODEL_DIR=path to training checkpoints directory
NUM_TRAIN_STEPS=50000
SAMPLE_1_OF_N_EVAL_EXAMPLES=1
python model_main_tf2.py -- \
--model_dir=$MODEL_DIR --
num_train_steps=$NUM_TRAIN_STEPS \
--
    e_1_of_n_eval_examples=$SAMPLE_1_OF_N_EVAL_
    LES \
    eline_config_path=$PIPELINE_CONFIG_PATH \
    logtostderr
```

```
"""
!python model_main_tf2.py --
pipeline_config_path=/content/drive/MyDrive/cust
omTF2/data/ssd_mobilenet_v2_320x320_coco17_tpu-
8.config --
model_dir=/content/drive/MyDrive/customTF2/train
ing --alsologtostderr
#ini digunakan untuk mengkonversi model yang
telah di training menjadi frozen inference graph
sehingga bisa digunakan untuk
#memprediksi gambar yang kita sediakan

!python exporter_main_v2.py \
--
trained_checkpoint_dir=/content/drive/MyDrive/cu
stomTF2/training \
--
pipeline_config_path=/content/drive/MyDrive/cust
omTF2/data/ssd_mobilenet_v2_320x320_coco17_tpu-
8.config \
--output_directory
/content/drive/MyDrive/customTF2/data/inference

# Jalankan ini supaya label pada obyek yang
dikenali menggunakan "font Arial"
%cd /content/models/research/object_detection

# Different font-type and font-size for labels
text
!wget
https://freefontsdownload.net/download/160187/ar
ial.zip
!unzip arial.zip -d .
%cd utils/
!sed -i "s/font =
font.truetype('arial.ttf', 24)/font =
font.truetype('arial.ttf', 24)/"
lization_utils.py
```

```
%cd ..
```

```
--2023-06-06 00:55:40--
https://freefontsdownload.net/download/160187/arial.zip
Resolving freefontsdownload.net (freefontsdownload.net)...
172.67.180.27, 104.21.75.182, 2606:4700:3036::6815:4bb6, ...
Connecting to freefontsdownload.net
(freefontsdownload.net)|172.67.180.27|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 172821 (169K) [application/force-download]
Saving to: 'arial.zip'
arial.zip           100%[===================>] 168.77K  --.-
KB/s    in 0.09s
2023-06-06 00:55:40 (1.94 MB/s) - 'arial.zip' saved
[172821/172821]
Archive: arial.zip
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@##################################@
@#         Downloaded from         #@
@#                                 #@
@#    www.FreeFontsDownload.net    #@
@#                                 #@
@# -------------- More site ------------ #@
@#       https://funnytv.net       #@
@#                                 #@
@#                                 #@
@##################################@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  inflating: ./www.freefontsdownload.net.url
 extracting: ./arial.png
  inflating: ./arial.ttf
    ting: ./freefontsdownload.txt
    t/models/research/object_detection/utils
    t/models/research/object_detection
```

```
#Script ini dipakai untuk uji coba mengenali
obyek pada gambar menu makanan nasi padang

#Loading the saved_model
import tensorflow as tf
import time
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from PIL import Image
from google.colab.patches import cv2_imshow
from object_detection.utils import
label_map_util
from object_detection.utils import
visualization_utils as viz_utils
import cv2

IMAGE_SIZE = (12, 8) # Output display size as
you want
import matplotlib.pyplot as plt
PATH_TO_SAVED_MODEL="/drive/customTF2/data/infer
ence/saved_model"
print('Loading model...', end='')

# Load saved model and build the detection
function
detect_fn=tf.saved_model.load(PATH_TO_SAVED_MODE
L)
print('Done!')

#Loading the label_map
category_index=label_map_util.create_category_in
dex_from_labelmap("/drive/customTF2/data/label_m
ap.pbtxt",use_display_name=True)
#category_index=label_map_util.create_category_i
    rom_labelmap([path_to_label_map],use_displ
    e=True)
```

```
def load_image_into_numpy_array(path):

    return np.array(Image.open(path))

image_path =
"/content/drive/MyDrive/customTF2/Baru/IMG202305
22145842.jpg"


image_temporary = "/resize_tmp.jpg"
img=cv2.imread(image_path)
img_75 = cv2.resize(img, None, fx = 0.40, fy =
0.40)
cv2.imwrite(image_temporary,img_75)


#image_path =
"/drive/customTF2/data/images/DSC00029.jpg"
#print('Running inference for {}...
'.format(image_path), end='')

image_np =
load_image_into_numpy_array(image_temporary)

# The input needs to be a tensor, convert it
using `tf.convert_to_tensor`.
input_tensor = tf.convert_to_tensor(image_np)
# The model expects a batch of images, so add an
axis with `tf.newaxis`.
input_tensor = input_tensor[tf.newaxis, ...]

detections = detect_fn(input_tensor)

# All outputs are batches tensors.
# Convert to numpy arrays, and take index [0] to
    e the batch dimension.
    ce only interested in the first
    etections.
```

```
num_detections =
int(detections.pop('num_detections'))
detections = {key: value[0,
:num_detections].numpy()
               for key, value in
detections.items()}
detections['num_detections'] = num_detections

# detection_classes should be ints.
detections['detection_classes'] =
detections['detection_classes'].astype(np.int64)

image_np_with_detections = image_np.copy()

viz_utils.visualize_boxes_and_labels_on_image_ar
ray(
      image_np_with_detections,
      detections['detection_boxes'],
      detections['detection_classes'],
      detections['detection_scores'],
      category_index,
      use_normalized_coordinates=True,
      max_boxes_to_draw=200,
      min_score_thresh=.6, # Adjust this value
to set the minimum probability boxes to be
classified as True
      agnostic_mode=False)
%matplotlib inline
plt.figure(figsize=IMAGE_SIZE, dpi=200)
plt.axis("off")
plt.imshow(image_np_with_detections)
plt.show()
```

Lampiran 2. Script Android

#Chain Code : #

```
class ChainCodeHelper {

    // Chain code untuk perpindahan ke tetangga dalam grid 8-arah
    private val chainCode = arrayOf(
        intArrayOf(-1, 0), // Barat
        intArrayOf(-1, 1), // Barat laut
        intArrayOf(0, 1),   // Utara
        intArrayOf(1, 1),   // Timur laut
        intArrayOf(1, 0),   // Timur
        intArrayOf(1, -1), // Tenggara
        intArrayOf(0, -1), // Selatan
        intArrayOf(-1, -1)  // Barat daya
    )


    private fun calculateAreaInPixelsFromPoint(point: Point, bitmap:
Bitmap, visited: Array<BooleanArray>): Double {
        val stack = Stack<Point>()
        stack.push(point)
        visited[point.x][point.y] = true
        var luas = 0
        while (!stack.isEmpty()) {
            val currentPoint = stack.pop()
            // Loop melalui chain code
            for (code in chainCode) {
                val neighborX = currentPoint.x + code[0]
                val neighborY = currentPoint.y + code[1]

                if (isValid(neighborX, neighborY, bitmap.width,
bitmap.height)
                    && bitmap.getPixel(neighborX, neighborY) ==
Color.BLACK
                    && !visited[neighborX][neighborY]
                ) {
                    stack.push(Point(neighborX, neighborY))
                    visited[neighborX][neighborY] = true
                    // Tambahkan panjang langkah (dalam satuan pixel)
                    luas++
                }
            }
        }
        return luas * 0.026;
    }

    private fun isValid(x: Int, y: Int, width: Int, height: Int):
Boolean {
        return x >= 0 && x < width && y >= 0 && y < height
```

```
#Coding menghitung berat :#
override fun onResults(
  results: MutableList<Detection>?,
  inferenceTime: Long,
  imageHeight: Int,
  imageWidth: Int
) {
    activity?.runOnUiThread {

fragmentCameraBinding.bottomSheetLayout.inferenceTimeVal.text =
                        String.format("%d ms", inferenceTime)

fragmentCameraBinding.bottomSheetLayout.layoutBaladoEggs.setVisibi
lity(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutCakedel.setVisibilit
y(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutCassavaLeaves.setVis
ibility(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutCassavaLeavesWithCoc
onutMilk.setVisibility(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutChickenGoulash.setVi
sibility(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutChickenPop.setVisibi
lity(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutFriedChicken.setVisi
bility(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutFriedFish.setVisibil
ity(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutFriedTempeTofu.setVi
sibility(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutGoulash.setVisibilit
y(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutGreenChili.setVisibi
lity(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutHotSpicy.setVisibili
ty(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutJackfruitVegetable.s
        ility(View.GONE)

        CameraBinding.bottomSheetLayout.layoutOmelet.setVisibility
        ONE)

        :CameraBinding.bottomSheetLayout.layoutRendang.setVisibilit
```

```
y(View.GONE)

fragmentCameraBinding.bottomSheetLayout.layoutRice.setVisibility(V
iew.GONE)

        var tebal: Float = 3.14f;
        var densitas: Float = 3.14f;
        var volume: Float = 3.14f;


        var massa: Float = 3.14f;
        var berat: Float = 3.14f;
        var kolesterol: Float = 3.14f;

        if (results != null) {
            if(results.isNotEmpty()){
                for (result in results) {

                    if(result.categories[0].label.contains("Balado
eggs",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutBaladoEggs.setVisibi
lity(View.VISIBLE)
                        tebal   =  3.1f
                        densitas = 1.74f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.98f

fragmentCameraBinding.bottomSheetLayout.baladoEggsLabel.text =
"Telur Balado  "+luas2

fragmentCameraBinding.bottomSheetLayout.baladoEggsVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Cakedel",ignoreCase =
true)){

fragmentCameraBinding.bottomSheetLayout.layoutCakedel.setVisibilit
y(View.VISIBLE)
                        tebal   =  2.4f
                        densitas = 1.19f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.1f

fragmentCameraBinding.bottomSheetLayout.cakedelLabel.text =
"Perkedel"

fragmentCameraBinding.bottomSheetLayout.cakedelVal.text =
        format("%.2f || %.2f", berat, kolesterol);
                    }

        lt.categories[0].label.contains("Cassava leaves",ignoreCase
        {
```

```
fragmentCameraBinding.bottomSheetLayout.layoutCassavaLeaves.setVis
ibility(View.VISIBLE)
                        berat = 100f
                        kolesterol = 0f

fragmentCameraBinding.bottomSheetLayout.cassavaLeavesLabel.text =
"Daun Singkong"

fragmentCameraBinding.bottomSheetLayout.cassavaLeavesVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Cassava leaves with
coconut milk",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutCassavaLeavesWithCoc
onutMilk.setVisibility(View.VISIBLE)
                        berat = 100f
                        kolesterol = 0f

fragmentCameraBinding.bottomSheetLayout.cassavaLeavesLabel.text =
"Sayur Singkong"

fragmentCameraBinding.bottomSheetLayout.cassavaLeavesVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Chicken goulash
thigh",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutChickenGoulash.setVi
sibility(View.VISIBLE)
                        tebal  =  3.7f
                        densitas = 1.62f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.186956522f

fragmentCameraBinding.bottomSheetLayout.chickenGoulashLabel.text =
"Ayam Gulai Paha"

fragmentCameraBinding.bottomSheetLayout.chickenGoulashVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Chicken
goulash",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutChickenGoulash.setVi
sibility(View.VISIBLE)
                        tebal  =  3.8f
                        densitas = 1.62f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.186956522f

:CameraBinding.bottomSheetLayout.chickenGoulashLabel.text =
```

```
"Ayam Gulai Dada"

fragmentCameraBinding.bottomSheetLayout.chickenGoulashVal.text
=String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Chicken pop
thigh",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutChickenPop.setVisibi
lity(View.VISIBLE)
                        tebal  =  3.8f
                        densitas = 1.62f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.571428571f

fragmentCameraBinding.bottomSheetLayout.chickenPopLabel.text =
"Ayam Pop Paha"

fragmentCameraBinding.bottomSheetLayout.chickenPopVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }


if(result.categories[0].label.contains("Chicken pop",ignoreCase =
true)){

fragmentCameraBinding.bottomSheetLayout.layoutChickenPop.setVisibi
lity(View.VISIBLE)
                        tebal  =  3.8f
                        densitas = 1.62f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.571428571f

fragmentCameraBinding.bottomSheetLayout.chickenPopLabel.text =
"Ayam Pop Dada"

fragmentCameraBinding.bottomSheetLayout.chickenPopVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }


                    if(result.categories[0].label.contains("Fried
fish",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutFriedFish.setVisibil
ity(View.VISIBLE)
                        tebal = 2.2f
                        densitas = 1.9f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.64f

:CameraBinding.bottomSheetLayout.friedFishLabel.text =
```

```
"Ikan Goreng"

fragmentCameraBinding.bottomSheetLayout.friedFishVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }
                    if(result.categories[0].label.contains("Fried
tempe/Tofu",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutFriedTempeTofu.setVi
sibility(View.VISIBLE)
                        tebal   =  1.3f
                        densitas = 2.01f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.0f

fragmentCameraBinding.bottomSheetLayout.friedTempeTofuLabel.text =
"Tempe/Tahu"

fragmentCameraBinding.bottomSheetLayout.friedTempeTofuVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Goulash",ignoreCase =
true)){

fragmentCameraBinding.bottomSheetLayout.layoutGoulash.setVisibilit
y(View.VISIBLE)
                        tebal   =  4.2f
                        densitas = 4.6f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.335f

fragmentCameraBinding.bottomSheetLayout.goulashLabel.text =
"Kikil"

fragmentCameraBinding.bottomSheetLayout.goulashVal.text
=String.format("%.2f || %.2f", berat, kolesterol);
                    }
                    if(result.categories[0].label.contains("Green
chili",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutGreenChili.setVisibi
lity(View.VISIBLE)
                        berat = 20f
                        kolesterol = 10f

fragmentCameraBinding.bottomSheetLayout.greenChiliLabel.text =
"Sambal Ijo"

    :CameraBinding.bottomSheetLayout.greenChiliVal.text =
    format("%.2f || %.2f", berat, kolesterol);
                    }
                    if(result.categories[0].label.contains("Hot
    ignoreCase = true)){
```

```
fragmentCameraBinding.bottomSheetLayout.layoutHotSpicy.setVisibili
ty(View.VISIBLE)
                        berat = berat + 0f
                        kolesterol = berat * 0f

fragmentCameraBinding.bottomSheetLayout.hotSpicyLabel.text =
"Sambal Merah"

fragmentCameraBinding.bottomSheetLayout.hotSpicyVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Jackfruit
vegetable",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutJackfruitVegetable.s
etVisibility(View.VISIBLE)
                        berat = 130f
                        kolesterol = 0f

fragmentCameraBinding.bottomSheetLayout.jackfruitVegetableLabel.te
xt = "Sayur Nangka"

fragmentCameraBinding.bottomSheetLayout.jackfruitVegetableVal.text
= String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Omelet",ignoreCase =
true)){
                        tebal  =  3.8f
                        densitas = 3.2f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 2.4f

fragmentCameraBinding.bottomSheetLayout.layoutOmelet.setVisibility
(View.VISIBLE)

fragmentCameraBinding.bottomSheetLayout.omeletLabel.text = "Telur
dadar"

fragmentCameraBinding.bottomSheetLayout.omeletVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Rendang",ignoreCase =
true)){
                        tebal  =  1.2f
                        densitas = 1.9f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 1.34f

    CameraBinding.bottomSheetLayout.layoutRendang.setVisibilit
    /ISIBLE)

    :CameraBinding.bottomSheetLayout.rendangVal.text =
```

```
String.format("%.2f || %.2f", berat, kolesterol);
                    }

if(result.categories[0].label.contains("Rice",ignoreCase = true)){

                        tebal  =  6.8f
                        densitas = 1.46f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.0f

fragmentCameraBinding.bottomSheetLayout.layoutRice.setVisibility(V
iew.VISIBLE)

fragmentCameraBinding.bottomSheetLayout.riceLabel.text = "Nasi  "
+luas1

fragmentCameraBinding.bottomSheetLayout.riceVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }
                    if(result.categories[0].label.contains("Fried
chicken thigh",ignoreCase = true)){
                        tebal  =  1.9f
                        densitas = 1.62f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.95f

fragmentCameraBinding.bottomSheetLayout.layoutRice.setVisibility(V
iew.VISIBLE)

fragmentCameraBinding.bottomSheetLayout.riceLabel.text = "Ayam
Goreng Paha"

fragmentCameraBinding.bottomSheetLayout.riceVal.text =
String.format("%.2f || %.2f", berat, kolesterol);
                    }

                    if(result.categories[0].label.contains("Fried
chicken",ignoreCase = true)){

fragmentCameraBinding.bottomSheetLayout.layoutFriedChicken.setVisi
bility(View.VISIBLE)
                        tebal  =  1.9f
                        densitas = 1.62f
                        volume = luas * tebal
                        massa = densitas * volume
                        berat = massa * 9.8f
                        kolesterol = berat * 0.95f

fragmentCameraBinding.bottomSheetLayout.friedChickenLabel.text =
        oreng Dada"

        CameraBinding.bottomSheetLayout.friedChickenVal.text =
        ormat("%.2f || %.2f", berat, kolesterol);
                    }
                }
```

```
                }
            }

fragmentCameraBinding.bottomSheetLayout.totalVal.text=berat.toStri
ng()+" mg"

            // Pass necessary information to OverlayView for drawing
on the canvas
            fragmentCameraBinding.overlay.setResults(
                results ?: LinkedList<Detection>(),
                imageHeight,
                imageWidth
            )
            // Force a redraw
            fragmentCameraBinding.overlay.invalidate()
        }
```