



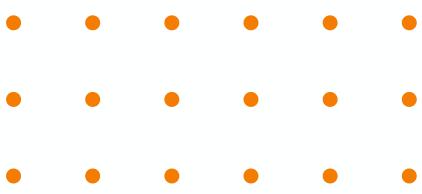
WARZONE

# FreshFruit

**KOMPRES 15**

Laboratorium Informatika Universitas Gunadarma





# Warzone

*"Menggunakan AI dan pengolahan data untuk meningkatkan kualitas dan keberlanjutan industri buah melalui deteksi kesegaran jeruk."*



**Haikal Abizar**  
**Mobile Developer**  
50421583



**Fathan Shani P. A**  
**Machine Learning**  
50421488



**Suryo Aji W**  
**Mobile Developer**  
51421449



**Wardiansyah F**  
**Machine Learning**  
51421514

# Kenapa Jeruk?

No	PROVINSI	2015	2016	2017	2018	2019	RATA-RATA	SHARE (%)
1	Jawa Timur	495.555	857.315	918.822	937.575	1.015.097	844.873	37,17
2	Sumatera Utara	485.962	467.746	453.530	410.938	298.234	423.282	18,62
3	Bali	129.440	84.260	169.002	225.584	349.775	191.612	8,43
4	Kalimantan Barat	148.766	125.257	127.490	144.016	140.593	137.224	6,04
5	Kalimantan Selatan	112.986	108.859	126.997	147.263	141.792	127.580	5,61
	Provinsi lainnya	483.366	495.022	499.469	645.044	617.995	548.179	24,12
	<b>Indonesia</b>	<b>1.856.076</b>	<b>2.138.459</b>	<b>2.295.310</b>	<b>2.510.420</b>	<b>2.563.486</b>	<b>2.272.752</b>	<b>100,00</b>

Sumber : BPS dan Ditjen Hortikultura



## Popularitas dan Produksi Tinggi:

- Jeruk adalah salah satu buah paling populer di dunia, dengan produksi global mencapai lebih dari 75 juta ton pada tahun 2020 (FAO).



## Permintaan Tinggi di Indonesia:

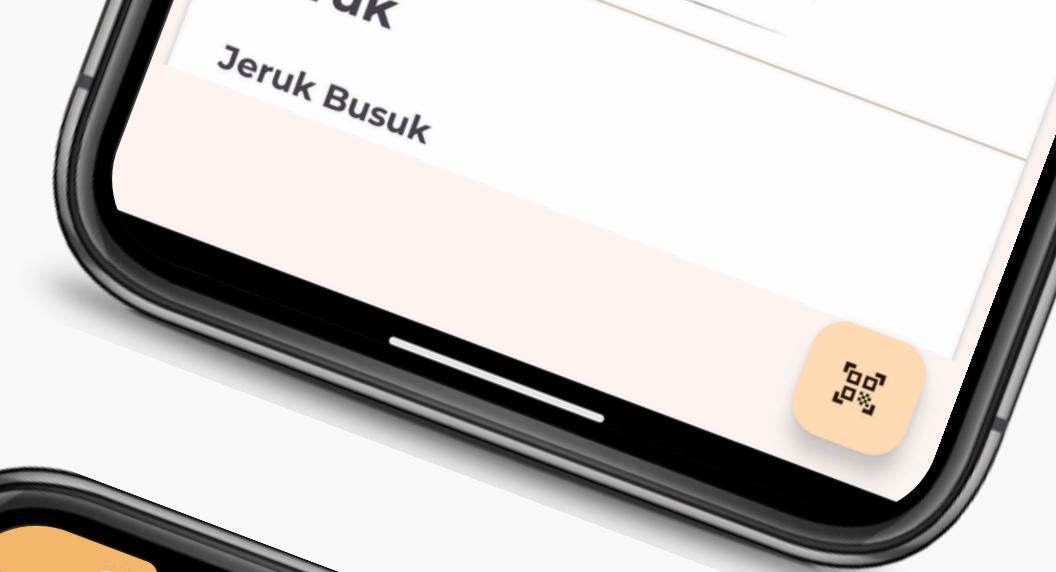
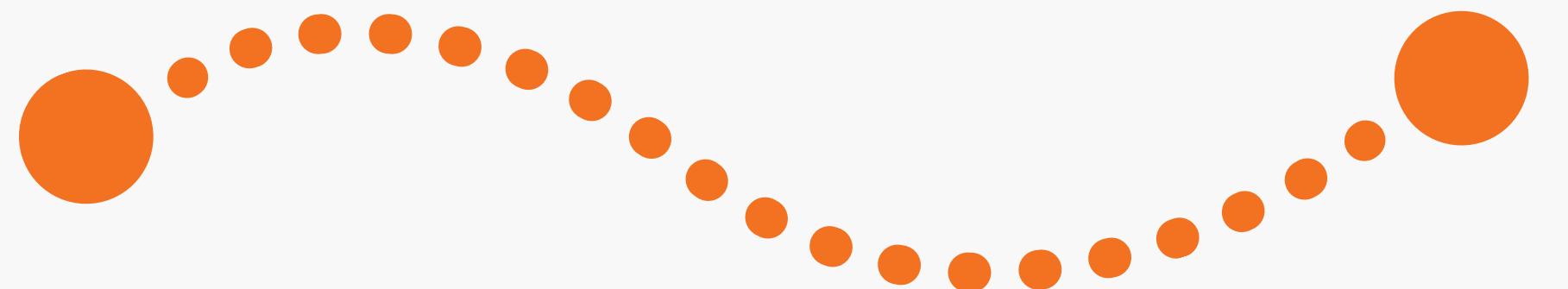
- Jeruk sangat diminati di Indonesia, namun banyak konsumen sering kesulitan memilih jeruk yang segar dan berkualitas baik di pasar tradisional.



## Pentingnya Pengolahan Data dan AI:

- Teknologi AI kami membantu memastikan kesegaran jeruk, mengurangi pemborosan, dan meningkatkan kepuasan konsumen.

# FreshFruit



# Pembuatan Model

kaggle

 roboflow

 YOLOv8

 PyTorch

 TensorFlow Lite

# Anotasi Data

FATHAN SHANI PUTRA A

Fresh Oranges: Job 1

Progress  
733 Images

Instructions  
No specific instructions were added when this job was assigned

Assignment  
Fathan Shani Putra A  
Labeler

Timeline

Fathan Shani Putra A added 733 to the dataset and set the job status to "completed".  
7/8/2023, 15:32:01

Fathan Shani Putra A created this Job and assigned it to fathanshani01@gmail.com.  
7/7/2023, 23:29:35

Unannotated 0

Annotated 733

TRAIN TEST TRAIN TRAIN TEST TRAIN

TEST TEST VALID TEST TRAIN

TEST TRAIN TRAIN TRAIN TRAIN

TRAIN TRAIN TRAIN TRAIN TRAIN

TRAIN TRAIN TRAIN TRAIN TRAIN

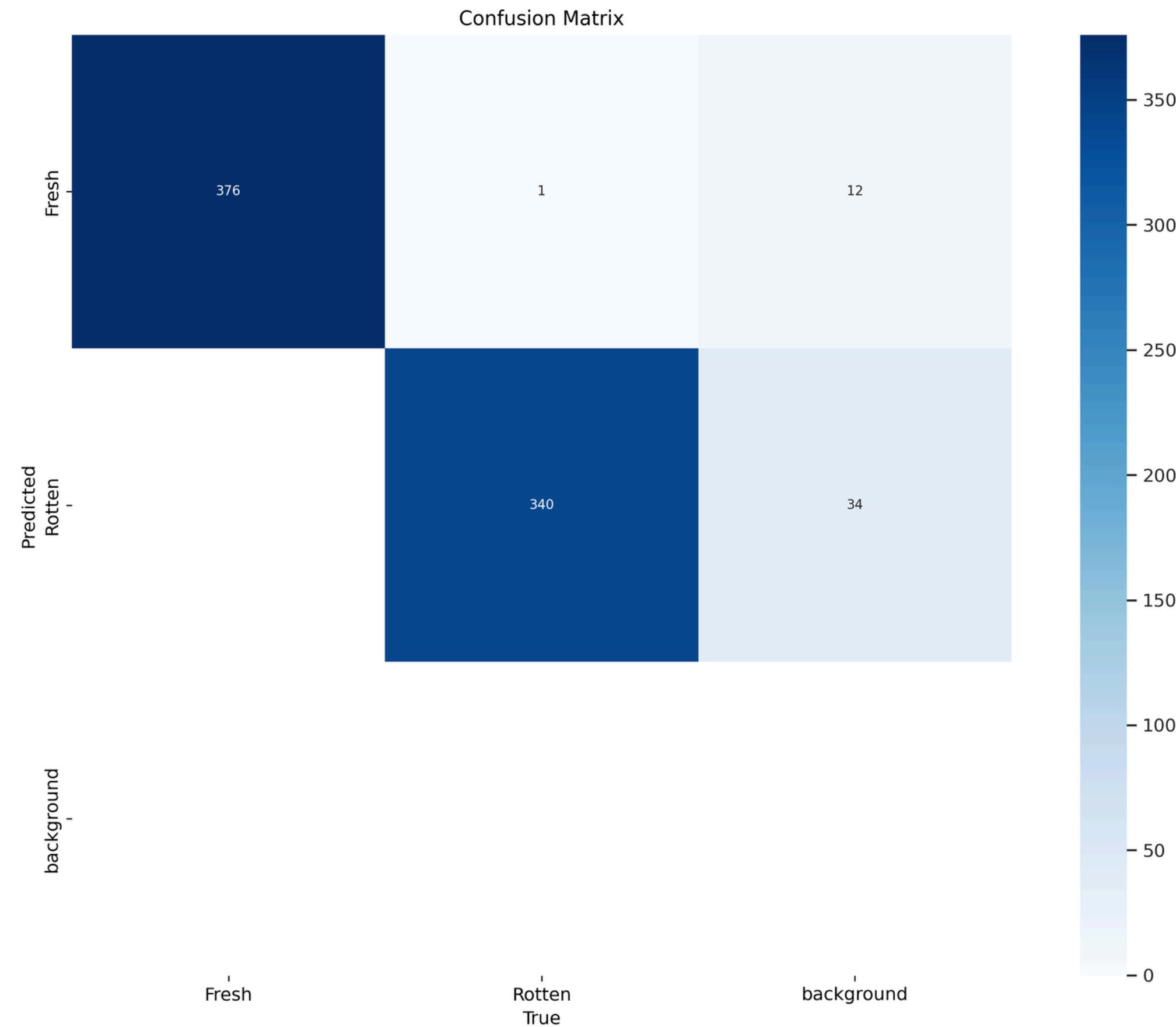
VALID

# Model Training

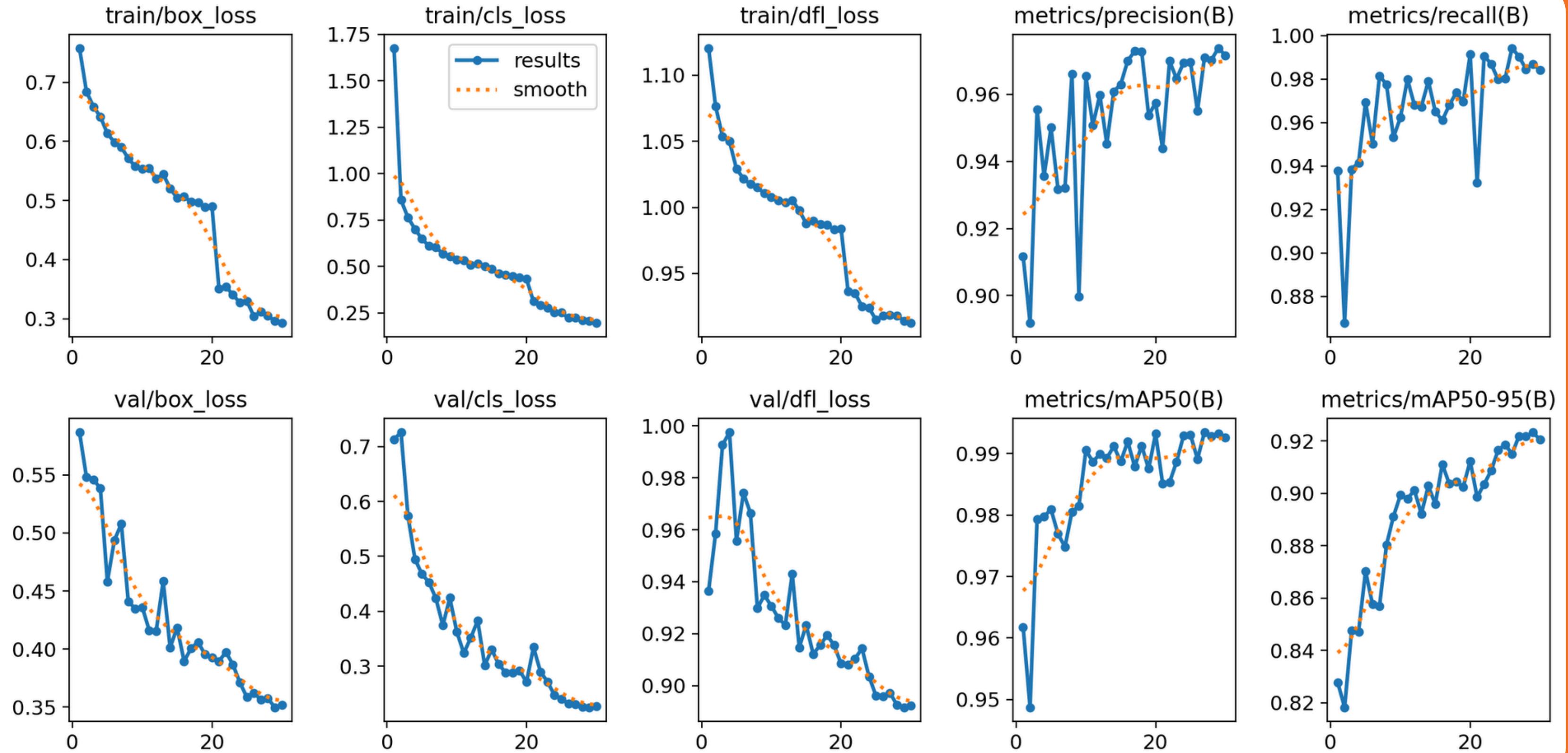
```
!mkdir {HOME}/datasets  
%cd {HOME}/datasets  
  
!pip install roboflow  
  
from roboflow import Roboflow  
rf = Roboflow(api_key="xxxxxxxxxxxxxxxxxx")  
project = rf.workspace("fathan-shani-putra-a-pw2zv").project("oranges-model")  
version = project.version(4)  
dataset = version.download("yolov8")
```

```
[ ] %cd {HOME}  
  
!yolo task=detect mode=train model=yolov8n.pt data={dataset.location}/data.yaml epochs=30 imgsze=160 plots=True  
→ 0% 0/134 [00:00<?, ?it/s]/usr/local/lib/python3.10/dist-packages/torch/autograd/graph.py:744: UserWarning: Plan failed with  
return Variable._execution_engine.run_backward( # Calls into the C++ engine to run the backward pass  
    1/30    0.627G    0.7582    1.677    1.121      46    160: 100% 134/134 [00:22<00:00,  5.88it/s]  
        Class   Images Instances Box(P) R mAP50 mAP50-95): 100% 20/20 [00:05<00:00,  3.96it/s]  
          all     612      717    0.912    0.938    0.962    0.828  
  
    Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size  
    2/30    0.512G    0.6845    0.8582    1.077      48    160: 100% 134/134 [00:17<00:00,  7.56it/s]  
        Class   Images Instances Box(P) R mAP50 mAP50-95): 100% 20/20 [00:04<00:00,  4.92it/s]  
          all     612      717    0.892    0.868    0.949    0.818  
  
    Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size  
    3/30    0.516G    0.659     0.7645    1.054      55    160: 100% 134/134 [00:17<00:00,  7.56it/s]  
        Class   Images Instances Box(P) R mAP50 mAP50-95): 100% 20/20 [00:04<00:00,  4.31it/s]  
          all     612      717    0.956    0.939    0.979    0.848
```

# Model Training



# Model Training



# Hasil Prediksi



# Ekspor Model ke TFLite

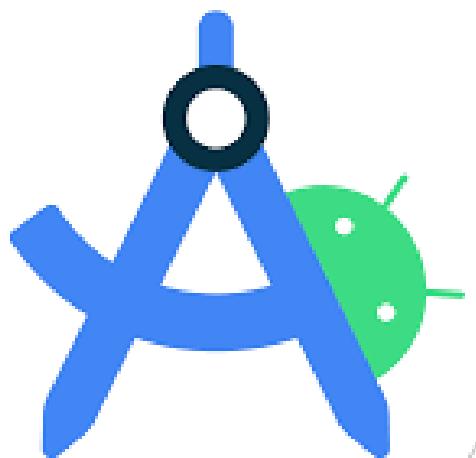
```
[ ] # Export model to tflite

from ultralytics import YOLO

# Load a model
model = YOLO('runs/detect/train/weights/best.pt') # load a custom trained model

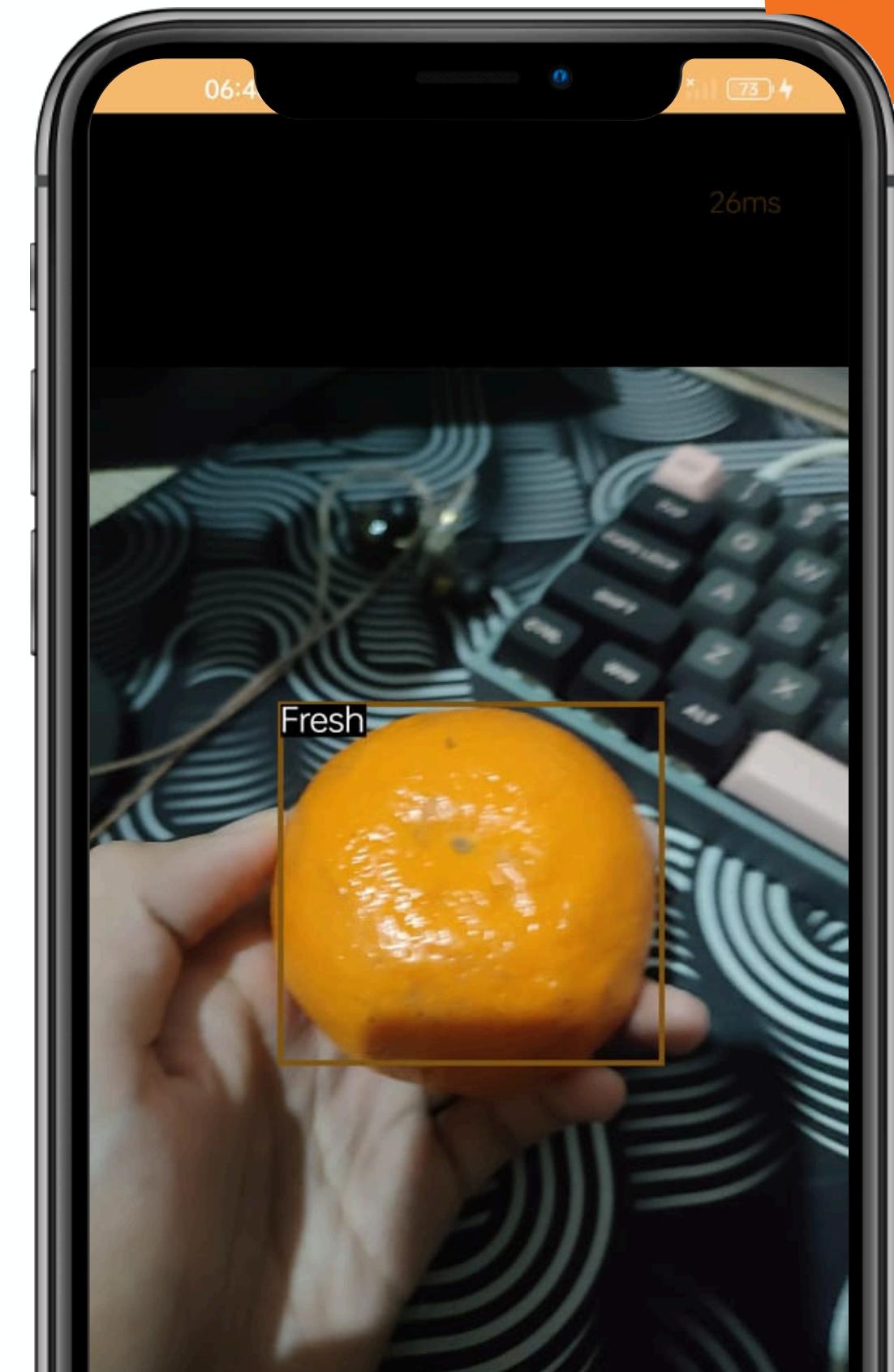
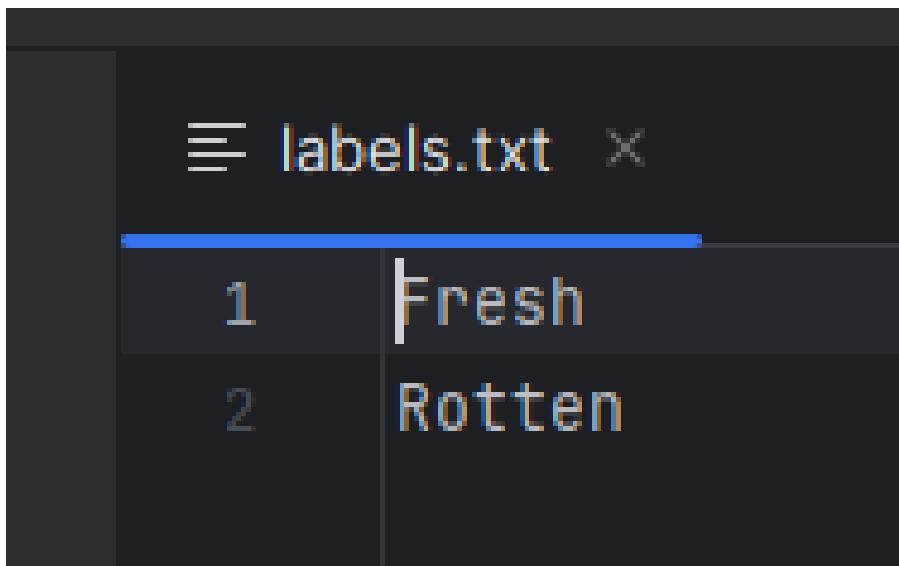
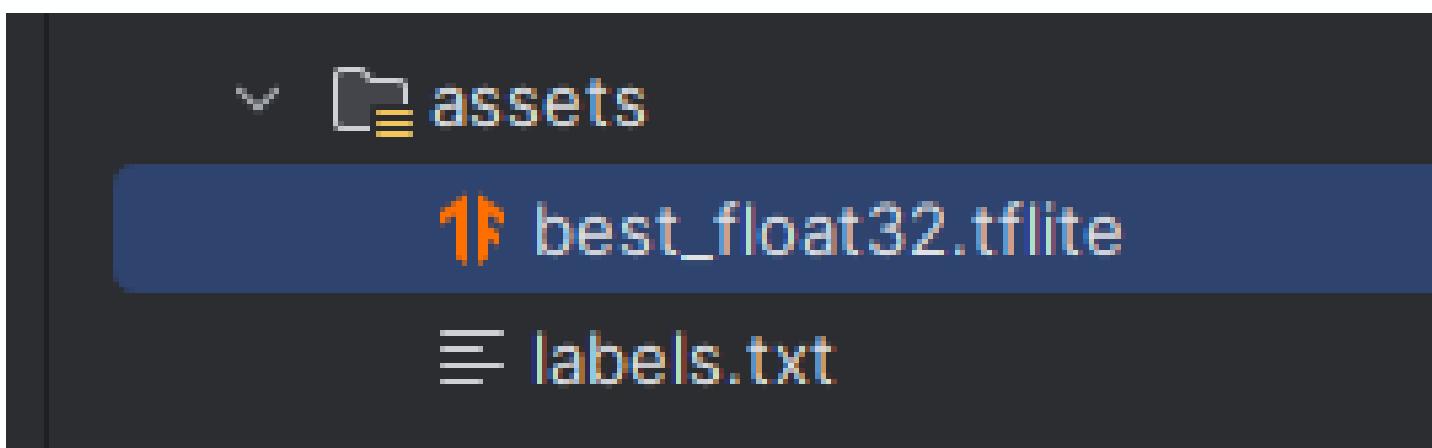
# Export the model
model.export(format='tflite')
```

# Android Implementation



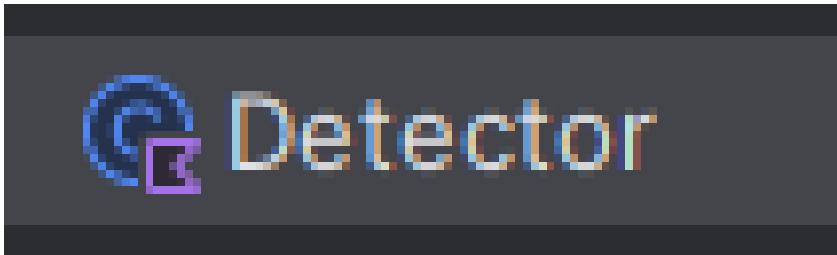
# Implementasi

1. Import Model TfLite dan Buat label untuk binding box



# Implementasi

## 2. Membuat class detector



```
fun detect(frame: Bitmap) {
    interpreter ?: return
    if (tensorWidth == 0) return
    if (tensorHeight == 0) return
    if (numChannel == 0) return
    if (numElements == 0) return

    var inferenceTime = SystemClock.uptimeMillis()

    val resizedBitmap = Bitmap.createScaledBitmap(frame, tensorWidth, tensorHeight, filter: false)

    val tensorImage = TensorImage(DataType.FLOAT32)
    tensorImage.load(resizedBitmap)
    val processedImage = imageProcessor.process(tensorImage)
    val imageBuffer = processedImage.buffer

    val output = TensorBuffer.createFixedSize(intArrayOf(1, numChannel, numElements), OUTPUT_IMAGE_TYPE)
    interpreter?.run(imageBuffer, output.buffer)

    val bestBoxes = bestBox(output.floatArray)
    inferenceTime = SystemClock.uptimeMillis() - inferenceTime

    if (bestBoxes == null) {
        detectorListener.onEmptyDetect()
        return
    }

    detectorListener.onDetect(bestBoxes, inferenceTime)
}
```

# Implementasi

3.

## Membuat class OverlayView



```
override fun draw(canvas: Canvas) {
    super.draw(canvas)

    results.forEach { it: BoundingBox
        val left = it.x1 * width
        val top = it.y1 * height
        val right = it.x2 * width
        val bottom = it.y2 * height

        canvas.drawRect(left, top, right, bottom, boxPaint)
        val drawableText = it.clsName

        textBackgroundPaint.getTextBounds(drawableText, start: 0, drawableText.length, bounds)
        val textWidth = bounds.width()
        val textHeight = bounds.height()
        canvas.drawRect(
            left,
            top,
            right: left + textWidth + BOUNDING_RECT_TEXT_PADDING,
            bottom: top + textHeight + BOUNDING_RECT_TEXT_PADDING,
            textBackgroundPaint
        )
        canvas.drawText(drawableText, left, y: top + bounds.height(), textPaint)
    }
}
```

# Implementasi

## 4. Membuat CameraActivity



```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityCameraBinding.inflate(layoutInflater)
    setContentView(binding.root)

    detector = Detector(baseContext, Constants.MODEL_PATH, Constants.LABELS_PATH, detectorListener: this)
    detector.setup()

    if (allPermissionsGranted()) {
        startCamera()
    } else {
        ActivityCompat.requestPermissions(activity: this, REQUIRED_PERMISSIONS, REQUEST_CODE_PERMISSIONS)
    }

    cameraExecutor = Executors.newSingleThreadExecutor()
}
```

# FreshFruit

