L A M P I R A N

Lampiran 1 - Script Aplikasi

```
import 'dart:io' as io;

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';
import 'package:permission_handler/permission_handler.dart';
import 'package:ultralytics_yolo/ultralytics_yolo.dart';
import 'package:ultralytics_yolo/yolo_model.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatefulWidget {
  const MyApp({super.key});

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  final controller = UltralyticsYoloCameraController();

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: FutureBuilder<bool>(
          future: _checkPermissions(),
```

```
builder: (context, snapshot) {
 final allPermissionsGranted = snapshot.data ?? false;

 if (!allPermissionsGranted) {
  return const Center(child: Text('Permissions not granted'));
 }

 return FutureBuilder<ObjectDetector>(
  future: _initObjectDetectorWithLocalModel(),
  builder: (context, snapshot) {
   if (snapshot.connectionState == ConnectionState.waiting) {
    return const Center(child: CircularProgressIndicator());
   }

   final predictor = snapshot.data;

   if (predictor == null) {
    return const Center(child: Text('Failed to initialize predictor'));
   }

   return Stack(
    children: [
     UltralyticsYoloCameraPreview(
      controller: controller,
      predictor: predictor,
      onCameraCreated: () {
       predictor.loadModel(useGpu: true);
      },
     ),
     StreamBuilder<double?>(
      stream: predictor.inferenceTime,
```

```
                builder: (context, snapshot) {
                  final inferenceTime = snapshot.data;

                  return StreamBuilder<double?>(
                    stream: predictor.fpsRate,
                    builder: (context, snapshot) {
                      final fpsRate = snapshot.data;

                      return Times(
                        inferenceTime: inferenceTime,
                        fpsRate: fpsRate,
                      );
                    },
                  );
                },
              ),
            ],
          );
        },
      );
    },
  ),
  floatingActionButton: FloatingActionButton(
    child: const Icon(Icons.cameraswitch),
    onPressed: () {
      controller.toggleLensDirection();
    },
  ),
 ),
);
}
```

```
Future<ObjectDetector> _initObjectDetectorWithLocalModel() async {
  final modelPath = await _copy('assets/best_float16.tflite');
  final metadataPath = await _copy('assets/metadata.yaml');
  final model = LocalYoloModel(
    id: '',
    task: Task.detect,
    format: Format.tflite,
    modelPath: modelPath,
    metadataPath: metadataPath,
  );

  return ObjectDetector(model: model);
}

Future<String> _copy(String assetPath) async {
  final path = '${(await getApplicationSupportDirectory()).path}/$assetPath';
  await io.Directory(dirname(path)).create(recursive: true);
  final file = io.File(path);
  if (!await file.exists()) {
    final byteData = await rootBundle.load(assetPath);
    await    file.writeAsBytes(byteData.buffer.asUint8List(byteData.offsetInBytes,
        byteData.lengthInBytes));
  }
  return file.path;
}

Future<bool> _checkPermissions() async {
  var cameraStatus = await Permission.camera.status;

  if (!cameraStatus.isGranted) {
```

```
      final newStatus = await Permission.camera.request();
      cameraStatus = newStatus;
    }

    return cameraStatus.isGranted;
  }

}

class Times extends StatelessWidget {
  const Times({
    super.key,
    required this.inferenceTime,
    required this.fpsRate,
  });

  final double? inferenceTime;
  final double? fpsRate;

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Align(
        alignment: Alignment.bottomCenter,
        child: Container(
            margin: const EdgeInsets.all(20),
            padding: const EdgeInsets.all(20),
            decoration: const BoxDecoration(
              borderRadius: BorderRadius.all(Radius.circular(10)),
```

```
import tensorflow as tf
import numpy as np
import cv2
import os

# Path model
saved_model_dir = "runs/export/saved_model"

# Representative dataset generator
def representative_data_gen():
    folder = "dataset/"
    for name in os.listdir(folder):
        img = cv2.imread(os.path.join(folder, name))
        img = cv2.resize(img, (640, 640))  # Sesuaikan ukuran input
        img = img / 255.0
        img = img.astype(np.float32)
        yield [np.expand_dims(img, axis=0)]

# Converter
converter = tf.lite.TFLiteConverter.from_saved_model(saved_model_dir)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
converter.representative_dataset = representative_data_gen
converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS_INT8]
converter.inference_input_type = tf.uint8
converter.inference_output_type = tf.uint8
```

```
# Konversi
tflite_model = converter.convert()

# Simpan
with open("best_int8.tflite", "wb") as f:
    f.write(tflite_model)
from ultralytics import YOLO

model = YOLO('yolov8n.pt')  # Model pretrained

# Parameter training
results = model.train(
    data='dataset/dataset.yaml',
    epochs=100,
    batch=96,
    imgsz=640,
    name='custom_train',
    optimizer='AdamW',
    lr0=0.001,
    augment=True,
    hsv_h=0.015,  # Augmentasi HSV-Hue
    hsv_s=0.7,    # Augmentasi HSV-Saturation
    hsv_v=0.4,    # Augmentasi HSV-Value
    flipud=0.5,   # Flip up-down
```

```
    fliplr=0.5,   # Flip left-right
    mosaic=1.0    # Mosaic augmentation
)

# 5. Evaluasi Model
metrics = model.val()
print(f"mAP50-95: {metrics.box.map}")
print(f"mAP50: {metrics.box.map50}")

# 6. Inference dengan Model Terlatih
model.predict(
    source='image.jpg',
    conf=0.5,
    save=True,
    show_labels=True,
    show_conf=True
)
```

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : DIAS RAFIKIL	Pembimbing I : Muh. Basri, S.T.,M.T.
NIM : 218280094	Pembimbing II : Ir. Untung Suwardoyo, S.Kom., M.T., IPP.
Judul Skripsi : PENDETEKSI NOMINAL MATA UANG KERTAS MENGGUNAKAN KAMERA	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 · Pakai kamera Live		Konsultasi 1 Pakaki Polaroid	
Konsultasi 2 · Jelaskan Cara deteksi nominal (berdasarkan apa?)		Konsultasi 2 Pakaki Rumus pakai.	
Konsultasi 3 Pergelar di Gab IV.		Konsultasi 3 → Pakaki alur pakai → Bab dgn kapala dan abstrak ada lalu.	
Konsultasi 4 Acc Judul		Konsultasi 4 Pakaki Bab 4, kapala dan abstrak.	
Konsultasi 5		Konsultasi 5 → Pandan → Lengkapi	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik