DAFTAR PUSTAKA

Sodri. 2013. Pengendalian Miniatur Instalasi Listrik Menggunakan Mikrokontoler Arduino Mega Terintegrasi HMI (Human Machine Interface. [Skipsi, Universitas Indonesia].

Banu P, Wawan. 2017. Si "Imut" Perusak Tanaman Cabai. (http://kaltim.litbang.pertanian.go.id/ind/index.php?option=com_content&view=article&id=900&Itemid=97, diakses 20 maret 2024).

Muliana, Vina A. 2017. 8 Negara Produsen Cabai Terbesar di Dunia, RI Termasuk?. (https://www.liputan6.com/bisnis/read/2819761/8-negara-produsen-cabaiterbesar-di-dunia-ritermasukm, diakses pada 20 Maret 2024).

Detiknews. 2020. Virus Gemini Bikin Ratusan Hektare Lahan Cabai di Malang Gagal Panen. (https://news.detik.com/berita-jawa-timur/d-4487319/virus-gemini-bikin-ratusan-hektare-lahan-cabai-di-malang-gagal-panen, dikakses 20 Maret 2024).

Kompas. 2010. Kena Virus, Tanaman Cabai Gagal Panen. (https://nasional.kompas.com/read/2010/07/20/18164737/~Regional~Sumatera, dikakses 21 Maret 2024).

Hasanuddin. 2022. RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN. Tesis. NTB :Jurusan Teknik Informatika. Universitas Cordova.

Nurfita, Royani Darma. 2018. IMPLEMENTASI DEEP LEARNING BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI. Tesis. Surakarta :Jurusan Teknik Informatika. Universitas Muhammadiyah Surakarta.

Hasanuddin. 2022. RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN. Tesis. NTB :Jurusan Teknik Informatika. Universitas Cordova.

Nizam, Muhammad, dkk. 2022. MIKROKONTROLER ESP 32 SEBAGAI ALAT MONITORING PINTU BERBASIS WEB. Jurnal. Blitar :Jurusan Sistem Komputer. Universitas Islam Balitar Blitar.

Novindri, Graciela Fausten, dkk. 2022. IMPLEMENTASI FLASK PADA SISTEM PENENTUAN MINIMAL ORDER UNTUK TIAP ITEM BARANG DI STRIBUTION CENTER PADA PT XYZ BERBASIS WEBSITE. Jurnal. narang :Jurusan Teknik Informatika. Universitas Kristen Satya Wacana.

Satria, Marius Raka, dkk. 2022. Image Captioning Menggunakan Metode Resnet50 Dan Long Short Term Memory. Jurnal. Bandung :Jurusan Teknik Informatika. Institut Teknologi Nasional.

Muharram, Risnaldy Fatwa, dkk. 2021. IMPLEMENTASI ARTIFICIAL INTELLIGENCE UNTUK DETEKSI MASKER SECARA REALTIME DENGAN TENSORFLOW DAN SSD MOBILENET BERBASIS PYTHON. Jurnal. Jakarta :Jurusan Teknik Informatika. Universitas Indraprasta PGRI.

Wijaya, Alexander Eric, dkk. 2021. IMPLEMENTASI TRANSFER LEARNING PADA CONVOLUTIONAL NEURAL NETWORK UNTUK DIAGNOSIS COVID-19 DAN PNEUMONIA PADA CITRA X-RAY. Tesis. Surabaya :Jurusan Teknik Informatika. Universitas Ma Chung.

Sihombing, R., Erwansyah, K., & Murniyanti, S. (2019). IMPLEMENTASI INTERNET OF THINGS (IOT) PENYIRAM BIBIT MELON DENGAN METODE SIMPLEX BERBASIS NODEMCU. CyberTech, 2(10), 1–8.

Jumasa, H. M., & Saputro, W. T. (2019). Prototipe Penyiram Tanaman Dan Pengukur Kelembaban Tanah Berbasis Arduino Uno. Jurnal INTEK, 2(2), 47–54.

Andrian Eko Widodo, S. N. M. J., Suleman, U. B. S. I., & Mahmud Safudin, U. B. S. I. (2019). PEMANFAATAN ARDUINO UNTUK MENDETEKSI KELEMBABAN TANAH. Jurnal Sains Dan Manajemen, 7(2), 1–5.

Sihombing, R., Erwansyah, K., & Murniyanti, S. (2019). IMPLEMENTASI INTERNET OF THINGS (IOT) PENYIRAM BIBIT MELON DENGAN METODE SIMPLEX BERBASIS NODEMCU. CyberTech, 2(10), 1–8.

L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

LAMPIRAN

Lampiran 1 Program ESP32

```
#include <Arduino.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

String wifiSSID = "Python";
String wifiPassword = "1234arthur";

void connectWifi();
void sendData(int sensorValue, String relayStatus);

#define PinSensor 32
#define PinRelay 4 // Misalnya, relay terhubung ke pin D2

const int Sensor = PinSensor;
const int MODE_AUTO = 1;
const int MODE_MANUAL = 0;
int mode;

String relayStatus = "";

void checkModeFromEndpoint();
void checkRelayCondition();

void setup() {
  Serial.begin(115200);
  pinMode(PinRelay, OUTPUT);
  digitalWrite(PinRelay, LOW);
  connectWifi();
}

void loop() {
  checkModeFromEndpoint();
  int valueSensor = analogRead(Sensor);
  Serial.print("Moisture value: ");
  Serial.println(valueSensor);
  if (mode == MODE_AUTO) {
    if (valueSensor > 2048) {
      digitalWrite(PinRelay, HIGH);
                Status = "ON";
                {
                lWrite(PinRelay, LOW);
                Status = "OFF";
```

```
    }
  } else if (mode == MODE_MANUAL) {
    checkRelayCondition();
  }
  sendData(valueSensor, relayStatus);
  delay(50);
}

void sendData(int sensorValue, String relayStatus) {
  String url = "https://iot-mobile-gizpapwlkq-uc.a.run.app/receive_data";
  HTTPClient http;
  String response;

  DynamicJsonDocument doc(256);
  String jsonParams;
  doc["sensorValue"] = sensorValue;
  doc["relayStatus"] = relayStatus;

  serializeJson(doc, jsonParams);
  Serial.println(jsonParams);

  http.begin(url);
  http.addHeader("Content-Type", "application/json");
  int httpCode = http.POST(jsonParams);
  if (httpCode > 0) {
    response = http.getString();
    Serial.println(response);
  } else {
    Serial.println("Error on HTTP request");
  }
  http.end();
}

void checkModeFromEndpoint() {
  HTTPClient http;
  String serverPath = "https://iot-mobile-gizpapwlkq-uc.a.run.app/check_mode";

  http.begin(serverPath);
  int httpResponseCode = http.GET();

  if (httpResponseCode > 0) {
    String payload = http.getString();
    // Cetak respons payload
    Serial.println(payload);

    DynamicJsonDocument doc(1024);
    deserializeJson(doc, payload);
    modeReceived = doc["mode"];
```

```
    if (modeReceived == 1) {
      mode = MODE_AUTO;
      Serial.println("Switched to Auto Mode");
    } else if (modeReceived == 0) {
      mode = MODE_MANUAL;
      Serial.println("Switched to Manual Mode");
    }
  } else {
    Serial.print("HTTP GET Request failed: ");
    Serial.println(httpResponseCode);
  }
  http.end();
}

void checkRelayCondition() {
  HTTPClient http;
  http.begin("https://iot-mobile-gizpapwlkq-uc.a.run.app/check_kondisi");
  int httpResponseCode = http.GET();

  if (httpResponseCode == 200) {
    String kondisiStr = http.getString();
    Serial.print("Response from server: ");
    Serial.println(kondisiStr);

    // Parse JSON
    StaticJsonDocument<200> jsonDoc;
    DeserializationError error = deserializeJson(jsonDoc, kondisiStr);
    if (error) {
      Serial.print("deserializeJson() failed: ");
      Serial.println(error.c_str());
      return;
    }

    const char* kondisi = jsonDoc["kondisi"];

    if (strcmp(kondisi, "ON") == 0) {
      digitalWrite(PinRelay, HIGH);
      relayStatus = "ON"; // Mengubah nilai variabel relayStatus global
    } else if (strcmp(kondisi, "OFF") == 0) {
      digitalWrite(PinRelay, LOW);
      relayStatus = "OFF"; // Mengubah nilai variabel relayStatus global
    }
```

```
    print("Error accessing endpoint. HTTP response code: ");
    println(httpResponseCode);
    lle HTTP error here if necessary
```

```
  }

  http.end();
}


void connectWifi() {
  Serial.println("Connecting To Wifi");
  WiFi.begin(wifiSSID.c_str(), wifiPassword.c_str());
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("Wifi Connected");
  Serial.println(WiFi.SSID());
  Serial.println(WiFi.RSSI());
  Serial.println(WiFi.macAddress());
  Serial.println(WiFi.localIP());
  Serial.println(WiFi.gatewayIP());
  Serial.println(WiFi.dnsIP());
}
```

Lampiran 2 Program API (Flask)

```
from flask import Flask, request, jsonify
from google.cloud import storage
import datetime

app = Flask(__name__)

data_received = {"sensorValue": None, "relayStatus": None}
mode = 1
kondisi = 0


storage_client = storage.Client()


bucket_name = "iot-mobile-arthur"
bucket = storage_client.bucket(bucket_name)

@app.route('/check_mode', methods=['GET'])
def check_mode():
    return jsonify({'mode': mode})

@app.route('/set_mode/<int:new_mode>', methods=['GET'])
def set_mode(new_mode):
    global mode
    mode = new_mode
    return jsonify({'message': f'Mode has been set to {mode}'}), 200

@app.route('/check_kondisi', methods=['GET'])
def check_kondisi():
    status = 'ON' if kondisi == 1 else 'OFF'
    return jsonify({'kondisi': status})

@app.route('/set_kondisi/<int:new_kondisi>', methods=['GET'])
def set_kondisi(new_kondisi):
    global kondisi
    kondisi = new_kondisi
    return jsonify({'message': f'Mode has been set to {new_kondisi}'}), 200

@app.route('/receive_data', methods=['POST'])
def receive_data():
    global data_received
                est.method == 'POST':
                 = request.json
                ta:
```

```
            timestamp_str = datetime.datetime.now().strftime('%Y-%m-%d
%H:%M:%S')
            data_received["sensorValue"] = data.get('sensorValue')
            data_received["relayStatus"] = data.get('relayStatus')
            response = {"message": "Data received successfully"}


            file_name = "data.txt"
            blob = bucket.blob(file_name)


            existing_content = blob.download_as_string()


            new_content =
f"{existing_content.decode()}{data_received['sensorValue']}
{data_received['relayStatus']} {timestamp_str}\n"


            blob.upload_from_string(new_content)

            return jsonify(response), 200
        else:
            return jsonify({"error": "No JSON data received"}), 400
    else:
        return jsonify({"error": "Only POST requests are allowed"}), 405

@app.route('/check_data', methods=['GET'])
def check_data():
    global data_received
    return jsonify(data_received), 200

if __name__ == '__main__':
    app.run(debug=True)
```

Lampiran 3 Program API (.flaskenv)

```
FLASK_APP=app.py
FLASK_DEBUG=1
```

Lampiran 4 Program API (dockerfile)

```
FROM python:3.10-slim

ENV PYTHONBUFFERED True

ENV APP_HOME /app

WORKDIR $APP_HOME

COPY . ./

RUN pip install -r requirements.txt

CMD exec gunicorn --bind :$PORT --workers 1 --threads 8 --timeout 0 app:app
```

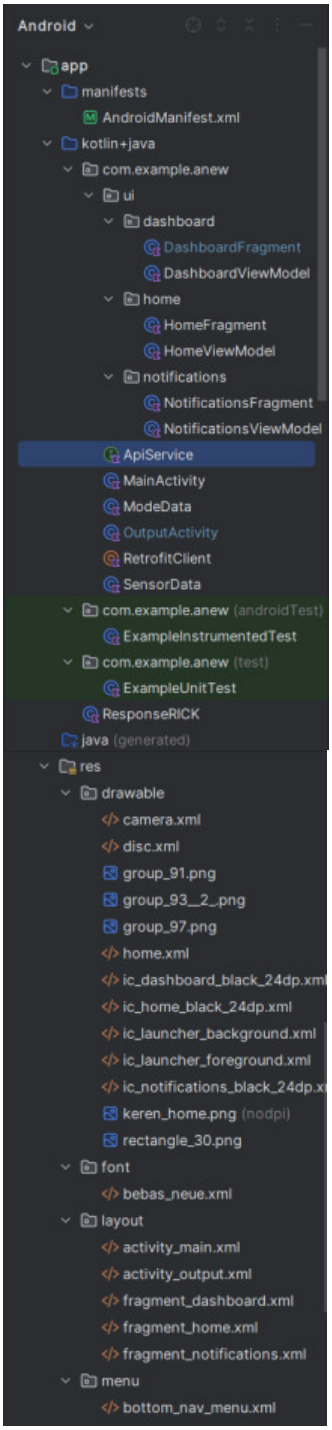
Lampiran 5 Program API (requirements.txt)

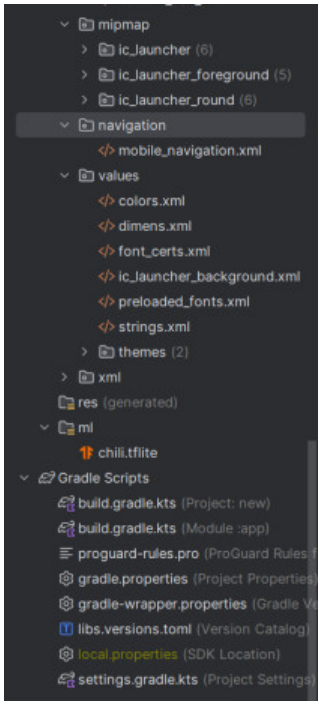
```
bidict==0.23.1
blinker==1.7.0
click==8.1.7
colorama==0.4.6
DateTime==5.4
flask==3.0.2
Flask-SocketIO==5.3.6
gunicorn==21.2.0
h11==0.14.0
importlib-metadata==7.0.1
itsdangerous==2.1.2
Jinja2==3.1.3
MarkupSafe==2.1.5
packaging==24.0
python-dotenv==1.0.1
python-engineio==4.9.0
python-socketio==5.11.1
pytz==2024.1
simple-websocket==1.0.0
werkzeug==3.0.1
wsproto==1.2.0
zipp==3.17.0
zope.interface==6.2
google==3.0.0
google-api-core==2.17.1
google-auth==2.28.2
google-cloud-core==2.4.1
google-cloud-storage==2.16.0
google-crc32c==1.5.0
google-resumable-media==2.7.0
googleapis-common-protos==1.63.0
```

Lampiran 6 Program API (.dockerignore)

```
Dockerfile
README.md
*.pyc
*.pyo
*.pyd
__pycache__
.pytest_cache
```

Lampiran 7 Android Studio Folder





Lampiran 8 Program Android (build.gradle.kt)

```
plugins {
    alias(libs.plugins.androidApplication)
    alias(libs.plugins.jetbrainsKotlinAndroid)
}


android {
    namespace = "com.example.anew"
    compileSdk = 34

    defaultConfig {
        applicationId = "com.example.anew"
        minSdk = 29
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = "1.8"
    }
    buildFeatures {
        viewBinding = true
        mlModelBinding = true
    }
}

    ıcies {

    ıentation(libs.androidx.core.ktx)
    ıentation(libs.androidx.appcompat)
```

```
    implementation(libs.material)
    implementation(libs.androidx.constraintlayout)
    implementation(libs.androidx.lifecycle.livedata.ktx)
    implementation(libs.androidx.lifecycle.viewmodel.ktx)
    implementation(libs.androidx.navigation.fragment.ktx)
    implementation(libs.androidx.navigation.ui.ktx)
    implementation(libs.tensorflow.lite.support)
    implementation(libs.tensorflow.lite.metadata)
    implementation(libs.androidx.activity)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
    //retrofit
    implementation("com.squareup.retrofit2:retrofit:2.9.0")
    implementation("com.squareup.retrofit2:converter-gson:2.9.0")
    //Glide
    implementation ("com.github.bumptech.glide:glide:4.16.0")
    implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.6.2")
    //material
    implementation("com.google.android.material:material:1.9.0")

}
```

Lampiran 9 Program Android (mobile_navigation.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mobile_navigation"
    app:startDestination="@+id/home">

    <fragment
        android:id="@+id/home"
        android:name="com.example.anew.ui.home.HomeFragment"
        android:label="@string/title_home"
        tools:layout="@layout/fragment_home" />

    <fragment
        android:id="@+id/camera"
        android:name="com.example.anew.ui.dashboard.DashboardFragment"
        android:label="@string/title_dashboard"
        tools:layout="@layout/fragment_dashboard" />

    <fragment
        android:id="@+id/disc"
        android:name="com.example.anew.ui.notifications.NotificationsFragment"
        android:label="@string/title_notifications"
        tools:layout="@layout/fragment_notifications" />
</navigation>
```

Lampiran 10 Program Android (bottom_nav_menu.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/home"
        android:icon="@drawable/home"
        android:title="home"/>

    <item
        android:id="@+id/camera"
        android:icon="@drawable/camera"
        android:title="camera" />

    <item
        android:id="@+id/disc"
        android:icon="@drawable/disc"
        android:title="disc" />

</menu>
```

Lampiran 11 Program Android (fragment_notifications.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context=".ui.notifications.NotificationsFragment">
    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
            <ImageView
                android:id="@+id/imageView6"
                android:layout_width="wrap_content"
                android:layout_height="1013dp"
                android:layout_marginTop="30dp"
                android:scaleType="fitStart"
                android:src="@drawable/group_101__2_" />
            <TextView
                android:id="@+id/textView8"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="BERIKUT VIDIO TUTORIALNYA"
                android:layout_marginBottom="60dp"
                android:textStyle="bold"
                android:textColor="#ADE112"
                android:gravity="center"
                android:textSize="27sp"
                />
            <WebView
                android:layout_width="match_parent"
                android:layout_height="200dp"
                android:layout_marginLeft="30dp"
                android:layout_marginRight="30dp"
                android:id="@+id/webView"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintHorizontal_bias="0.0"
                app:layout_constraintStart_toStartOf="parent" />
            nearLayout>
            llView>
            ayout>
```

Lampiran 12 Program Android (fragment_home.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.home.HomeFragment">

    <TextView
        android:id="@+id/text_home"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="111dp"
        android:scaleType="centerCrop"
        android:src="@drawable/group_93__2_"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.666"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="294dp"
        android:layout_height="535dp"
        android:src="@drawable/keren_home"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
            layout_constraintTop_toBottomOf="@+id/imageView" />

            /iew
            oid:id="@+id/textView2"
            oid:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginStart="115dp"
    android:layout_marginTop="44dp"
    android:fontFamily="@font/bebas_neue"


    android:textColor="#FDFDFD"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/imageView2" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="170dp"
    android:layout_marginTop="155dp"
    android:fontFamily="@font/bebas_neue"

    android:textColor="#FDFDFD"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/imageView2" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"

    android:layout_marginBottom="170dp"

    android:fontFamily="@font/bebas_neue"
    android:textColor="#FDFDFD"
    android:textSize="25sp"
    app:layout_constraintBottom_toBottomOf="@+id/imageView2"
    app:layout_constraintEnd_toEndOf="@+id/imageView2"
    app:layout_constraintStart_toStartOf="@+id/imageView2"
    app:layout_constraintTop_toBottomOf="@+id/text_home" />

<TextView
    oid:id="@+id/textView5"
    oid:layout_width="wrap_content"
    oid:layout_height="wrap_content"
    oid:layout_marginStart="190dp"
    oid:layout_marginBottom="58dp"
```

```
    android:fontFamily="monospace"
    android:text="More"
    android:textColor="#AEA9A9"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/imageView2"
    app:layout_constraintEnd_toEndOf="@+id/imageView2"
    app:layout_constraintStart_toStartOf="@+id/imageView2"
    app:layout_constraintTop_toTopOf="@+id/imageView2" />

<Button
    android:id="@+id/button4"
    android:layout_width="81dp"
    android:layout_height="41dp"
    android:backgroundTint="#DAF3A2"
    android:onClick="sendOnRequest"
    android:text="AUTO"
    android:textColor="#FDFDFD"
    android:textSize="12sp"
    app:layout_constraintBottom_toBottomOf="@+id/imageView2"
    app:layout_constraintEnd_toEndOf="@+id/imageView2"
    app:layout_constraintHorizontal_bias="0.162"
    app:layout_constraintStart_toEndOf="@+id/button3"
    app:layout_constraintTop_toTopOf="@+id/imageView"
    app:layout_constraintVertical_bias="0.745" />

<Button
    android:id="@+id/button3"
    android:layout_width="81dp"
    android:layout_height="41dp"
    android:backgroundTint="#DAF3A2"
    android:onClick="sendOffRequest"
    android:text="MANUAL"
    android:textColor="#FDFDFD"
    android:textSize="10sp"
    app:layout_constraintBottom_toBottomOf="@+id/imageView2"
    app:layout_constraintEnd_toStartOf="@+id/textView5"
    app:layout_constraintHorizontal_bias="0.532"
    app:layout_constraintStart_toStartOf="@+id/text_home"
    app:layout_constraintTop_toTopOf="@+id/imageView"
    app:layout_constraintVertical_bias="0.745" />

<Button
    android:id="@+id/button7"
    oid:layout_width="47dp"
    oid:layout_height="32dp"
    oid:backgroundTint="#DAF3A2"
    oid:text="OFF"
    oid:textColor="#FDFDFD"
```

```
        android:textSize="6sp"
        app:layout_constraintBottom_toBottomOf="@+id/imageView2"
        app:layout_constraintEnd_toEndOf="@+id/text_home"
        app:layout_constraintHorizontal_bias="0.666"
        app:layout_constraintStart_toStartOf="@+id/text_home"
        app:layout_constraintTop_toTopOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.926"
        android:onClick="turnOff"/>

    <Button
        android:id="@+id/button6"
        android:layout_width="47dp"
        android:layout_height="32dp"
        android:backgroundTint="#DAF3A2"
        android:text="ON"
        android:textColor="#FDFDFD"
        android:textSize="6sp"
        app:layout_constraintBottom_toBottomOf="@+id/imageView2"
        app:layout_constraintEnd_toEndOf="@+id/text_home"
        app:layout_constraintHorizontal_bias="0.827"
        app:layout_constraintStart_toStartOf="@+id/text_home"
        app:layout_constraintTop_toTopOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.926"
        android:onClick="turnOn"/>

    <ImageView
        android:id="@+id/imageView5"
        android:layout_width="122dp"
        android:layout_height="47dp"
        android:layout_marginTop="328dp"
        android:src="@drawable/rectangle_30"
        app:layout_constraintBottom_toBottomOf="@+id/imageView2"
        app:layout_constraintEnd_toEndOf="@+id/text_home"
        app:layout_constraintHorizontal_bias="0.866"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.0" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Lampiran 13 Program Android (fragment_dashboard.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.dashboard.DashboardFragment">

    <Button
        android:id="@+id/button"
        android:layout_width="360dp"
        android:layout_height="wrap_content"
        android:layout_above="@id/button2"
        android:layout_centerInParent="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="0dp"
        android:backgroundTint="#C4F651"
        android:text="Camera"
        android:textAllCaps="false"
        android:textSize="21sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/button2"
        android:layout_width="360dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerInParent="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="86dp"
        android:backgroundTint="#C4F651"
        android:text="Galeri"
        android:textAllCaps="false"
        android:textSize="21sp"
        android:textStyle="bold" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="370sp"
        android:layout_height="370sp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10sp" />

        /iew
        oid:id="@+id/classified"
        oid:layout_width="wrap_content"
        oid:layout_height="wrap_content"
```

```
        android:layout_below="@+id/imageView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10sp"
        android:text="Classified as:"
        android:textColor="#C4F651"
        android:textSize="24sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/result"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/classified"
        android:layout_centerHorizontal="true"
        android:text=""
        android:textColor="#C30000"
        android:textSize="27sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/result"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10sp"
        android:backgroundTint="#C4F651"
        android:text="Detail"
        android:visibility="invisible" />


</RelativeLayout>
```

Lampiran 14 Program Android (activity_output.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/main"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".OutputActivity">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="24dp"
            android:layout_x="161dp"
            android:layout_y="-35dp"
            android:text="TextView"
            android:textColor="#C4F651"
            android:textSize="48sp"
            android:textStyle="bold"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.501"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/textView6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="50dp"
            android:layout_marginTop="96dp"
            android:layout_marginEnd="50dp"
            android:justificationMode="inter_word"
            android:padding="20dp"

            android:text="Deskripsi"

            )p:layout_constraintEnd_toEndOf="parent"
            )p:layout_constraintHorizontal_bias="0.498"
            )p:layout_constraintStart_toStartOf="parent"
```

```
            app:layout_constraintTop_toBottomOf="@+id/webView" />

        <WebView
            android:id="@+id/webView"
            android:layout_width="0dp"
            android:layout_height="200dp"
            android:layout_marginStart="16dp"
            android:layout_marginTop="24dp"
            android:layout_marginEnd="16dp"
            android:layout_x="0dp"
            android:layout_y="0dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.0"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/textView" />

        <TextView
            android:id="@+id/textView7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="348dp"
            android:text="Tindakan : "
            android:textColor="#C4F651"
            android:textSize="24sp"
            android:textStyle="bold"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

    </androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>
```

Lampiran 15 Program Android (activity_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >



    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        app:itemIconTint="#ADE112"
        app:itemTextColor="#ADE112"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu" />

    <fragment
        android:id="@+id/nav_host_fragment_activity_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@+id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Lampiran 16 Program Android (SensorData.kt)

```
package com.example.anew
import com.google.gson.annotations.SerializedName

data class SensorData(
    @SerializedName("relayStatus") val relayStatus: String,
    @SerializedName("sensorValue") val sensorValue: Int
)
```

Lampiran 17 Program Android (RetrofitClient.kt)

```
package com.example.anew
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

object RetrofitClient {
    private const val BASE_URL = "https://iot-mobile-gizpapwlkq-uc.a.run.app"

    val instance: ApiService by lazy {
        val retrofit = Retrofit.Builder()
            .baseUrl(BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build()

        retrofit.create(ApiService::class.java)
    }
}
```

Lampiran 18 Program Android (OutputActivity.kt)

```
package com.example.anew

import android.os.Bundle
import android.webkit.WebChromeClient
import android.webkit.WebSettings
import android.webkit.WebView
import android.widget.ImageView
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.example.anew.databinding.ActivityOutputBinding

class OutputActivity : AppCompatActivity() {
    private lateinit var binding: ActivityOutputBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityOutputBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Hide the top bar (action bar)
        supportActionBar?.hide()



        // Retrieve the string data passed from the previous activity
        val hasilPrediksi = intent.getStringExtra("hasilPrediksi")


        // Initialize the nama TextView using the binding
        val nama: TextView = binding.textView
        val deskripsi: TextView = binding.textView6
//        val video: WebView = binding.webView


        // Set the text of the nama TextView to the received string data
        nama.text = hasilPrediksi
//        nama.text = "curl leaf" <<--- UBAH INI


        if (hasilPrediksi == "leaf spot") {
            val video =
                """<iframe width="100%" height="100%"
                s://www.youtube.com/embed/t1eIkOhfYbk?si=v2YMVq3iaBDQnSnf"
                uTube video player" frameborder="0" allow="accelerometer; autoplay;
                -write; encrypted-media; gyroscope; picture-in-picture; web-share"
                screen></iframe>"""
                nding.webView.loadData(video, "text/html", "utf-8")
```

```
            binding.webView.settings.javaScriptEnabled = true
            binding.webView.webChromeClient = WebChromeClient()
            binding.webView.settings.cacheMode =
WebSettings.LOAD_NO_CACHE
            binding.webView.settings.domStorageEnabled = true
            deskripsi.text = "Pemangkasan: Potong dan buang daun yang terinfeksi
secara menyeluruh. Jangan lupa untuk membersihkan pisau pemangkas Anda
dengan disinfektan antara setiap pemotongan untuk mencegah penyebaran
infeksi.\n" +
                    "\n" +
                    "Penyiraman: Hindari menyiram tanaman cabai secara berlebihan.
Tanah yang terlalu lembab dapat memperburuk kondisi penyakit. Pastikan tanah
memiliki drainase yang baik dan hanya menyiram tanaman ketika tanah terasa
kering di permukaan.\n" +
                    "\n" +
                    "Pemupukan: Berikan pupuk yang seimbang untuk membantu
meningkatkan kekebalan tanaman. Pupuk yang kaya akan unsur nitrogen dapat
membantu tanaman memproduksi daun baru yang lebih kuat.\n" +
                    "\n" +
                    "Pengendalian hama: Hama seperti kutu daun dan tungau laba-laba
dapat melemahkan tanaman cabai, membuatnya lebih rentan terhadap infeksi
penyakit. Gunakan insektisida organik atau bahan alami lainnya untuk
mengendalikan hama-hama ini.\n" +
                    "\n" +
                    "Penggunaan fungisida: Jika infeksi leaf spot sangat parah, Anda
mungkin perlu menggunakan fungisida yang direkomendasikan untuk melawan
jamur atau bakteri penyebab penyakit. Pastikan untuk mengikuti petunjuk
penggunaan dengan tepat dan menghindari penggunaan yang berlebihan.\n" +
                    "\n" +
                    "Rotasi tanaman: Untuk mencegah penyebaran infeksi, hindari
menanam tanaman cabai di tempat yang sama setiap tahunnya. Rotasi tanaman
dapat membantu mengurangi akumulasi patogen di tanah.\n" +
                    "\n" +
                    "Sanitasi: Pastikan untuk membersihkan semua alat dan peralatan
taman setelah digunakan, terutama jika Anda telah bekerja dengan tanaman yang
terinfeksi. Ini akan membantu mencegah penyebaran penyakit ke tanaman lain.\n"
+
                    "\n" +
                    "Pemantauan: Selalu perhatikan perkembangan tanaman cabai Anda
secara teratur. Jika Anda melihat gejala baru atau memburuknya infeksi, segera
ambil tindakan yang diperlukan."
        }else if (hasilPrediksi == "curl leaf") {
            val video =
                """<iframe width="100%" height="100%"
                s://www.youtube.com/embed/LFPqVu6Hhwc?si=5eMpJ-a6Eqe5sz50"
                uTube video player" frameborder="0" allow="accelerometer; autoplay;
                -write; encrypted-media; gyroscope; picture-in-picture; web-share"
                screen></iframe>"""
```

```
            binding.webView.loadData(video, "text/html", "utf-8")
            binding.webView.settings.javaScriptEnabled = true
            binding.webView.webChromeClient = WebChromeClient()
            binding.webView.settings.cacheMode =
WebSettings.LOAD_NO_CACHE
            binding.webView.settings.domStorageEnabled = true
            deskripsi.text ="Pemeliharaan Rutin: Pastikan tanaman cabai mendapatkan
perawatan rutin seperti penyiraman yang cukup, pemupukan sesuai dengan
kebutuhan tanaman, dan penyiangan gulma.\n" +
                    "\n" +
                    "Penyiraman yang Tepat: Pastikan penyiraman dilakukan secara
teratur tetapi tidak terlalu berlebihan. Tanah harus tetap lembab tetapi tidak
tergenang air. Kebiasaan menyiram tanaman di pagi hari juga bisa membantu
mengurangi risiko penyakit.\n" +
                    "\n" +
                    "Pemupukan Seimbang: Pastikan tanaman cabai mendapatkan nutrisi
yang cukup, terutama nitrogen, fosfor, dan kalium. Namun, hindari pemupukan
berlebihan karena hal ini juga bisa menyebabkan masalah pada tanaman.\n" +
                    "\n" +
                    "Perawatan Tanah: Pastikan tanah tempat tanaman cabai tumbuh
memiliki drainase yang baik. Ini dapat mencegah penumpukan air yang berlebihan
di sekitar akar tanaman.\n" +
                    "\n" +
                    "Pengendalian Hama dan Penyakit: Awasi tanaman cabai Anda untuk
tanda-tanda serangan hama atau penyakit. Jika ditemukan, segera ambil tindakan
untuk mengendalikannya, baik dengan menggunakan insektisida organik atau
fungisida yang sesuai.\n" +
                    "\n" +
                    "Pemangkasan yang Tepat: Jika daun-daun yang terinfeksi terlalu
banyak, lakukan pemangkasan dengan hati-hati. Potong daun-daun yang sudah
terinfeksi dan pastikan untuk membersihkan alat pemangkas sebelum dan setelah
digunakan untuk mencegah penyebaran penyakit.\n" +
                    "\n" +
                    "Perawatan Varietas yang Tahan Penyakit: Jika memungkinkan, pilih
varietas cabai yang tahan terhadap penyakit curl leaf atau penyakit lainnya yang
umum menyerang tanaman cabai di daerah Anda.\n" +
                    "\n" +
                    "Perhatikan Faktor Lingkungan: Beberapa faktor lingkungan seperti
kelembaban udara yang tinggi atau perubahan suhu yang ekstrem juga dapat
menyebabkan curl leaf pada tanaman cabai. Usahakan untuk menciptakan
lingkungan yang optimal bagi tanaman."
            // Code untuk hasilPrediksi "curl leaf"
        } else if (hasilPrediksi == "healthy") {
                ıl video =
                 """<iframe width="100%" height="100%"
                s://www.youtube.com/embed/aA6b0bnzrWo?si=_5Scj4vB86UJPmaa"
                uTube video player" frameborder="0" allow="accelerometer; autoplay;
```

```
clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>"""
        binding.webView.loadData(video, "text/html", "utf-8")
        binding.webView.settings.javaScriptEnabled = true
        binding.webView.webChromeClient = WebChromeClient()
        binding.webView.settings.cacheMode =
WebSettings.LOAD_NO_CACHE
        binding.webView.settings.domStorageEnabled = true
        deskripsi.text ="Untuk menjaga tanaman cabai tetap sehat, berikut adalah
beberapa tindakan yang dapat Anda lakukan:\n" +
            "\n" +
            "Pemilihan Lokasi yang Tepat: Tanaman cabai membutuhkan sinar
matahari penuh. Pilihlah lokasi yang terkena sinar matahari langsung setidaknya
6-8 jam sehari untuk pertumbuhan yang optimal.\n" +
            "\n" +
            "Persiapan Tanah yang Baik: Pastikan tanah memiliki drainase yang
baik, struktur yang longgar, dan kaya akan bahan organik. Anda bisa
menambahkan kompos atau pupuk kandang untuk meningkatkan kesuburan
tanah.\n" +
            "\n" +
            "Penyiraman yang Teratur: Tanaman cabai membutuhkan
penyiraman teratur untuk menjaga tanah tetap lembab, terutama saat tanaman
masih muda atau saat cuaca panas. Namun, hindari penyiraman berlebihan yang
bisa menyebabkan akar membusuk.\n" +
            "\n" +
            "Pemupukan yang Teratur: Berikan pupuk secara teratur untuk
memastikan tanaman mendapatkan nutrisi yang cukup. Pilihlah pupuk yang kaya
akan nitrogen, fosfor, dan kalium, serta mengandung unsur mikro penting seperti
magnesium, kalsium, dan belerang.\n" +
            "\n" +
            "Pemangkasan yang Tepat: Lakukan pemangkasan secara teratur
untuk menghilangkan cabang atau daun yang mati atau terinfeksi penyakit. Ini
juga membantu meningkatkan sirkulasi udara di antara tanaman.\n" +
            "\n" +
            "Pengendalian Hama dan Penyakit: Perhatikan tanda-tanda serangan
hama atau penyakit seperti kutu daun, ulat, atau penyakit jamur. Ambil tindakan
pencegahan seperti menggunakan insektisida organik atau fungisida jika
diperlukan.\n" +
            "\n" +
            "Pemberian Dukungan: Beberapa varietas cabai membutuhkan
dukungan agar tidak roboh ketika buah mulai bertumbuh. Gunakan tiang atau
kawat untuk memberikan dukungan yang cukup kepada tanaman.\n" +
            "\n" +
            "Pemantauan Lingkungan: Perhatikan kondisi lingkungan di sekitar
termasuk suhu udara, kelembaban, dan kualitas udara. Sesuaikan
n Anda sesuai dengan perubahan lingkungan.\n" +
            "\n" +
```

```
                "Perlindungan dari Suhu Ekstrem: Lindungi tanaman cabai dari suhu ekstrem, baik suhu panas maupun dingin. Anda bisa menggunakan mulsa untuk menjaga kelembaban tanah dan menstabilkan suhu tanah.\n" +
                "\n" +
                "Perawatan Rutin: Lakukan perawatan rutin seperti membersihkan gulma, mengontrol pertumbuhan gulma, dan memonitor perkembangan tanaman secara berkala."
            // Code untuk hasilPrediksi "healthy"
        } else if (hasilPrediksi == "whitefly") {
            val video =
                """<iframe width="100%" height="100%" src="https://www.youtube.com/embed/fjASKfr6Y7c?si=lA4FzFHql8kPO7uz" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>"""
            binding.webView.loadData(video, "text/html", "utf-8")
            binding.webView.settings.javaScriptEnabled = true
            binding.webView.webChromeClient = WebChromeClient()
            binding.webView.settings.cacheMode = WebSettings.LOAD_NO_CACHE
            binding.webView.settings.domStorageEnabled = true
            deskripsi.text ="Pemantauan Rutin: Perhatikan tanaman Anda secara rutin untuk mendeteksi adanya tanda-tanda serangan kutu daun putih, seperti adanya sejumlah kecil serangga berwarna putih yang terbang di sekitar tanaman atau meninggalkan jejak-kotoran yang lengket pada daun.\n" +
                "\n" +
                "Pembersihan Daun: Gunakan semprotan air bertekanan rendah untuk membersihkan daun secara teratur. Ini membantu menghilangkan telur dan larva kutu daun putih, serta mengurangi populasi hama.\n" +
                "\n" +
                "Pemangkasan: Jika infestasi cukup parah, pertimbangkan untuk memangkas daun-daun yang terinfeksi secara signifikan. Ini dapat membantu mengurangi jumlah tempat persembunyian bagi kutu daun putih dan memperbaiki sirkulasi udara di sekitar tanaman.\n" +
                "\n" +
                "Aplikasi Insektisida: Gunakan insektisida yang disetujui untuk mengendalikan kutu daun putih. Pilihlah formula yang sesuai untuk tanaman cabai Anda dan ikuti petunjuk penggunaan dengan cermat. Pastikan untuk menyemprotkan insektisida di bawah daun juga, karena itu adalah tempat favorit kutu daun putih bersembunyi.\n" +
                "\n" +
                "Penggunaan Pestisida Alami: Bahan-bahan seperti minyak neem, sabun insektisida, atau ekstrak tanaman tertentu (misalnya, ekstrak bawang putih
                i merah) dapat digunakan sebagai alternatif alami untuk mengendalikan
                ı putih. Namun, pastikan untuk menguji efeknya terlebih dahulu pada
                kecil tanaman dan perhatikan kemungkinan efek samping.\n" +
                "\n" +
```

```
                "Memperhatikan Tanaman Tumbuhan Pendamping: Beberapa
tanaman tertentu, seperti tanaman bunga Calendula atau tanaman aromatik seperti
lavender, bisa membantu menarik serangga pemangsa alami yang dapat
membantu mengendalikan populasi kutu daun putih.\n" +
                "\n" +
                "Perhatikan Lingkungan: Sirkulasi udara yang baik, kelembaban yang
rendah, dan penempatan tanaman yang tidak terlalu rapat dapat membantu
mengurangi kemungkinan serangan kutu daun putih. Hindari over-fertilisasi,
karena kelebihan nutrisi dapat meningkatkan ketertarikan tanaman terhadap
hama.\n" +
                "\n" +
                "Ketekunan dalam Perawatan: Penting untuk konsisten dalam
mengambil tindakan pengendalian dan melakukan pemantauan terus-menerus
untuk memastikan bahwa populasi kutu daun putih tetap terkendali.\n" +
                "\n"
            // Code untuk hasilPrediksi "whitefly"
        } else if (hasilPrediksi == "yellowish") {
            val video =
                """<iframe width="100%" height="100%"
src="https://www.youtube.com/embed/2AvxIVDL090?si=DC3gAvLs8S-YTPE-"
title="YouTube video player" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>"""
            binding.webView.loadData(video, "text/html", "utf-8")
            binding.webView.settings.javaScriptEnabled = true
            binding.webView.webChromeClient = WebChromeClient()
            binding.webView.settings.cacheMode =
WebSettings.LOAD_NO_CACHE
            binding.webView.settings.domStorageEnabled = true
            deskripsi.text ="Pemeriksaan Akar: Periksa akar tanaman untuk
memastikan bahwa tidak terjadi kerusakan atau pembusukan pada akar. Akar yang
sehat penting untuk penyerapan nutrisi yang baik.\n" +
                "\n" +
                "Pemberian Nutrisi: Tanaman cabai mungkin mengalami kekurangan
nutrisi tertentu seperti nitrogen, besi, magnesium, atau zat lainnya yang diperlukan
untuk pertumbuhan dan perkembangan yang sehat. Berikan pupuk yang
mengandung nutrisi yang dibutuhkan tanaman.\n" +
                "\n" +
                "Penyiraman yang Tepat: Pastikan penyiraman dilakukan secara
teratur dan tidak terlalu berlebihan. Terlalu banyak air dapat menyebabkan akar
membusuk dan mengurangi penyerapan nutrisi. Tanaman cabai biasanya
memerlukan penyiraman lebih sering saat suhu tinggi.\n" +
                "\n" +
                "Pengendalian Hama dan Penyakit: Periksa tanaman cabai untuk
ıui apakah terjadi serangan hama seperti kutu daun atau tungau. Serangan
ıat menyebabkan daun menjadi kuning. Jika terjadi serangan, gunakan
engendalian yang sesuai seperti insektisida organik.\n" +
                "\n" +
```

```
                "Perawatan Lingkungan: Pastikan tanaman cabai mendapatkan sinar matahari yang cukup dan tidak terlalu terpapar angin kencang atau suhu yang ekstrem. Lingkungan yang tidak sesuai bisa menyebabkan stres pada tanaman.\n" +
                "\n" +
                "Pemeriksaan pH Tanah: Periksa pH tanah tempat tanaman cabai tumbuh. Tanaman cabai biasanya membutuhkan pH tanah antara 6 hingga 7. Jika pH tanah terlalu rendah atau tinggi, ini dapat mempengaruhi penyerapan nutrisi oleh tanaman.\n" +
                "\n" +
                "Pemangkasan: Lakukan pemangkasan pada bagian tanaman yang mati atau terinfeksi. Hal ini dapat membantu mempercepat pemulihan tanaman dan meningkatkan sirkulasi udara di sekitar tanaman.\n" +
                "\n" +
                "Perawatan Rutin: Lakukan perawatan rutin seperti membersihkan gulma, memastikan tanah tetap subur, dan memantau kondisi tanaman secara berkala."
            // Code untuk hasilPrediksi "yellowish"
        }


    }
}
```

Lampiran 19 Program Android (ModeData.kt)

```
package com.example.anew

import com.google.gson.annotations.SerializedName

data class ModeData(
    @SerializedName("mode") val mode: Int
)
```

Lampiran 20 Program Android (MainActivity.kt)

```
package com.example.anew

import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.Switch
import android.widget.TextView
import com.google.android.material.bottomnavigation.BottomNavigationView
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.LiveData
import androidx.lifecycle.liveData
import androidx.navigation.findNavController
import androidx.navigation.ui.AppBarConfiguration
import androidx.navigation.ui.setupActionBarWithNavController
import androidx.navigation.ui.setupWithNavController
import com.example.anew.databinding.ActivityMainBinding
import okhttp3.ResponseBody
import retrofit2.Response

import retrofit2.Call
import retrofit2.Callback



class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)


        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
//        val textView2 = findViewById<TextView>(R.id.textView2)

        val handler = android.os.Handler()
        handler.postDelayed(object : Runnable {
            override fun run() {
                requestSensorData()
                handler.postDelayed(this, 500) // 5000 milliseconds delay (5 seconds)

            }, 500) // 5000 milliseconds delay (5 seconds)
```

```
        val navView: BottomNavigationView = binding.navView

        val navController =
findNavController(R.id.nav_host_fragment_activity_main)
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        val appBarConfiguration = AppBarConfiguration(
            setOf(
                R.id.home, R.id.camera, R.id.disc
            )
        )
        setupActionBarWithNavController(navController, appBarConfiguration)
        navView.setupWithNavController(navController)



    }


    private fun requestSensorData() {
        // Memanggil Retrofit untuk mendapatkan data sensor
        val sensorCall = RetrofitClient.instance.getSensorData()
        sensorCall.enqueue(object : Callback<SensorData> {
            override fun onResponse(call: Call<SensorData>, response:
Response<SensorData>) {
                if (response.isSuccessful) {
                    // Mendapatkan data sensor
                    val sensorData = response.body()
                    if (sensorData != null) {
                        // Menetapkan data sensor ke TextView yang sesuai
                        val sensorValueTextView =
findViewById<TextView>(R.id.textView2)
                        sensorValueTextView?.text = sensorData.sensorValue.toString()
                        val relayStatusTextView =
findViewById<TextView>(R.id.textView3)
                        relayStatusTextView?.text = sensorData.relayStatus
                    } else {
                        // Handle kasus ketika data sensor kosong (misalnya, tampilkan
pesan kesalahan)
                    }
                } else {
                    // Handle kesalahan dalam respons (misalnya, tampilkan pesan
kesalahan)
                }


            verride fun onFailure(call: Call<SensorData>, t: Throwable) {
                // Handle kegagalan
```

```
            }


        })

        // Memanggil Retrofit untuk mendapatkan data mode
        val modeCall = RetrofitClient.instance.getModeData()
        modeCall.enqueue(object : Callback<ModeData> {
            override fun onResponse(call: Call<ModeData>, response:
Response<ModeData>) {
                if (response.isSuccessful) {
                    // Mendapatkan data mode
                    val modeData = response.body()
                    if (modeData != null) {
                        // Menetapkan teks sesuai dengan mode ke TextView4
                        val modeTextView = findViewById<TextView>(R.id.textView4)
                        if (modeData.mode == 1) {
                            modeTextView?.text = "MODE : AUTO"
                        } else {
                            modeTextView?.text = "MODE : MANUAL"
                        }
                    } else {
                        // Handle kasus ketika data mode kosong (misalnya, tampilkan
pesan kesalahan)
                    }
                } else {
                    // Handle kesalahan dalam respons (misalnya, tampilkan pesan
kesalahan)
                }
            }

            override fun onFailure(call: Call<ModeData>, t: Throwable) {
                // Handle kegagalan
            }
        })
    }

    fun sendOffRequest(view: View) {
        val modeCall = RetrofitClient.instance.setMode(0)
        modeCall.enqueue(object : Callback<ResponseBody> {
            override fun onResponse(call: Call<ResponseBody>, response:
Response<ResponseBody>) {
                if (response.isSuccessful) {
                    // Handle respons sukses (opsional)
                } else {
                    // Handle respons gagal (opsional)
                }
```

```
            override fun onFailure(call: Call<ResponseBody>, t: Throwable) {
                // Handle kegagalan (opsional)
            }
        })
    }

    fun sendOnRequest(view: View) {
        val modeCall = RetrofitClient.instance.setMode(1)
        modeCall.enqueue(object : Callback<ResponseBody> {
            override fun onResponse(call: Call<ResponseBody>, response:
Response<ResponseBody>) {
                if (response.isSuccessful) {
                    // Handle respons sukses (opsional)
                } else {
                    // Handle respons gagal (opsional)
                }
            }
            override fun onFailure(call: Call<ResponseBody>, t: Throwable) {
                // Handle kegagalan (opsional)
            }
        })
    }

    fun turnOff(view: View) {
        val modeCall = RetrofitClient.instance.setKondisi(0)
        modeCall.enqueue(object : Callback<ResponseBody> {
            override fun onResponse(call: Call<ResponseBody>, response:
Response<ResponseBody>) {
                if (response.isSuccessful) {
                    // Handle respons sukses (opsional)
                } else {
                    // Handle respons gagal (opsional)
                }
            }
            override fun onFailure(call: Call<ResponseBody>, t: Throwable) {
                // Handle kegagalan (opsional)
            }
        })
    }

    fun turnOn(view: View) {
        val modeCall = RetrofitClient.instance.setKondisi(1)
        modeCall.enqueue(object : Callback<ResponseBody> {
            override fun onResponse(call: Call<ResponseBody>, response:
<ResponseBody>) {
                if (response.isSuccessful) {
                    // Handle respons sukses (opsional)
                } else {
```

```
                // Handle respons gagal (opsional)
            }
        }
        override fun onFailure(call: Call<ResponseBody>, t: Throwable) {
            // Handle kegagalan (opsional)
        }
    })
}

}
```

Lampiran 21 Program Android (ApiService.kt)

```
package com.example.anew


import okhttp3.ResponseBody
import retrofit2.Call
import retrofit2.http.GET
import retrofit2.http.Path

interface ApiService {
    @GET("check_data")
    fun getSensorData(): Call<SensorData>

    @GET("check_mode")
    fun getModeData(): Call<ModeData>

    @GET("set_mode/{mode}")
    fun setMode(@Path("mode") mode: Int): Call<ResponseBody>

    @GET("set_kondisi/{kondisi}")
    fun setKondisi(@Path("kondisi") mode: Int): Call<ResponseBody>
}
```

Lampiran 22 Program Android (NotificationsViewModel.kt)

```
package com.example.anew.ui.notifications

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.webkit.WebChromeClient
import android.webkit.WebSettings
import android.webkit.WebView
import android.widget.TextView
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import com.example.anew.databinding.FragmentNotificationsBinding

class NotificationsFragment : Fragment() {
    private var _binding: FragmentNotificationsBinding? = null
    private val binding get() = _binding!!
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val notificationsViewModel =
            ViewModelProvider(this).get(NotificationsViewModel::class.java)
        _binding = FragmentNotificationsBinding.inflate(inflater, container, false)
        val root: View = binding.root
        val textView: TextView = binding.textNotifications
        notificationsViewModel.text.observe(viewLifecycleOwner) {
            textView.text = it}
        val webView: WebView = binding.webView
        webView.settings.javaScriptEnabled = true
        webView.webChromeClient = WebChromeClient()
        webView.settings.cacheMode = WebSettings.LOAD_NO_CACHE
        webView.settings.domStorageEnabled = true
        val html = """<iframe width="100%" height="100%"
src="https://www.youtube.com/embed/Ft8e-MZ7-
Rw?si=7sNqhCZBtmDrUOmH" title="YouTube video player" title="YouTube
video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>"""
        webView.loadData(html, "text/html", "utf-8")
        n root}
        le fun onDestroyView() {
        r.onDestroyView()
        ding = null } }
```

Lampiran 23 Program Android (NotificationsFragment.kt)

```
package com.example.anew.ui.notifications

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import com.example.anew.databinding.FragmentNotificationsBinding

class NotificationsFragment : Fragment() {

    private var _binding: FragmentNotificationsBinding? = null

    // This property is only valid between onCreateView and
    // onDestroyView.
    private val binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val notificationsViewModel =
            ViewModelProvider(this).get(NotificationsViewModel::class.java)

        _binding = FragmentNotificationsBinding.inflate(inflater, container, false)
        val root: View = binding.root

        val textView: TextView = binding.textNotifications
        notificationsViewModel.text.observe(viewLifecycleOwner) {
            textView.text = it
        }
        return root
    }

    override fun onDestroyView() {
        super.onDestroyView()
        _binding = null
    }
}
```

Lampiran 24 Program Android (HomeViewModel.kt)

```
package com.example.anew.ui.home

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel

class HomeViewModel : ViewModel() {

    private val _text = MutableLiveData<String>().apply {
        value = ""
    }
    val text: LiveData<String> = _text
}
```

Lampiran 25 Program Android (HomeFragment.kt)

```
package com.example.anew.ui.home

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import com.example.anew.databinding.FragmentHomeBinding

class HomeFragment : Fragment() {

    private var _binding: FragmentHomeBinding? = null
    // This property is only valid between onCreateView and
    // onDestroyView.
    private val binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val homeViewModel =
            ViewModelProvider(this).get(HomeViewModel::class.java)

        _binding = FragmentHomeBinding.inflate(inflater, container, false)
        val root: View = binding.root

        val textView: TextView = binding.textHome
        homeViewModel.text.observe(viewLifecycleOwner) {
            textView.text = it
        }
        return root
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        (activity as AppCompatActivity?)?.supportActionBar?.hide()
    }
            le fun onDestroyView() {
            r.onDestroyView()
             ding = null
```

Lampiran 26 Program Android (DashboardViewModel.kt)

```
package com.example.anew.ui.home

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import com.example.anew.databinding.FragmentHomeBinding

class HomeFragment : Fragment() {

    private var _binding: FragmentHomeBinding? = null
    // This property is only valid between onCreateView and
    // onDestroyView.
    private val binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val homeViewModel =
            ViewModelProvider(this).get(HomeViewModel::class.java)

        _binding = FragmentHomeBinding.inflate(inflater, container, false)
        val root: View = binding.root

        val textView: TextView = binding.textHome
        homeViewModel.text.observe(viewLifecycleOwner) {
            textView.text = it
        }
        return root
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        (activity as AppCompatActivity?)?.supportActionBar?.hide()
    }
            le fun onDestroyView() {
            r.onDestroyView()
             ding = null
```

Lampiran 27 Program Android (DashboardFragment.kt)

```
package com.example.anew.ui.dashboard

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.media.ThumbnailUtils
import android.os.Bundle
import android.provider.MediaStore
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.ImageView
import android.widget.TextView
import androidx.core.view.isVisible
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import com.example.anew.OutputActivity
import com.example.anew.databinding.FragmentDashboardBinding
import com.example.anew.ml.Chili
import org.tensorflow.lite.DataType
import org.tensorflow.lite.support.metadata.schema.ImageSize
import org.tensorflow.lite.support.tensorbuffer.TensorBuffer
import org.w3c.dom.Text
import java.io.IOException
import java.nio.ByteBuffer
import java.nio.ByteOrder
import kotlin.math.min


class DashboardFragment : Fragment() {

    private var _binding: FragmentDashboardBinding? = null

    // This property is only valid between onCreateView and
    // onDestroyView.
    private val binding get() = _binding!!

                nera: Button? = null
                ageView: ImageView? = null
                ult: TextView? = null
                ageSize = 150
                ilPrediksi: String? = null
```

```
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val dashboardViewModel =
            ViewModelProvider(this).get(DashboardViewModel::class.java)


        _binding = FragmentDashboardBinding.inflate(inflater, container, false)
        val root: View = binding.root

        val camera: Button = binding.button
        val gallery: Button = binding.button2
        val detail: Button = binding.button5

        camera.setOnClickListener {
            if (requireContext().checkSelfPermission(Manifest.permission.CAMERA)
== PackageManager.PERMISSION_GRANTED) {
                val cameraIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
                startActivityForResult(cameraIntent, 3)
            } else {
                requestPermissions(arrayOf(Manifest.permission.CAMERA), 100)
            }
        }
        gallery.setOnClickListener {
            val galleryIntent = Intent(
                Intent.ACTION_PICK,
                MediaStore.Images.Media.EXTERNAL_CONTENT_URI
            )
            startActivityForResult(galleryIntent, 1)
        }

        detail.setOnClickListener {
            val intent = Intent(requireContext(), OutputActivity::class.java)
            intent.putExtra("hasilPrediksi", hasilPrediksi)
            startActivity(intent)
        }
        return root
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        val imageView: ImageView = binding.imageView


            r.onActivityResult(requestCode, resultCode, data)


            sultCode == Activity.RESULT_OK) {
```

```
            when (requestCode) {
                3 -> {
                    val imageBitmap = data?.extras?.get("data") as? Bitmap
                    imageBitmap?.let {
                        val dimension = min(it.width, it.height)
                        val thumbnail = ThumbnailUtils.extractThumbnail(it, dimension,
dimension)
                        imageView?.setImageBitmap(thumbnail)


                        val scaledImage = Bitmap.createScaledBitmap(thumbnail,
imageSize, imageSize, false)
                        classifyImage(scaledImage)
                    }
                }
                else -> {
                    val selectedImageUri = data?.data
                    selectedImageUri?.let {
                        try {
                            val image =
MediaStore.Images.Media.getBitmap(requireContext().contentResolver, it)
                            imageView?.setImageBitmap(image)


                            val scaledImage = Bitmap.createScaledBitmap(image,
imageSize, imageSize, false)
                            classifyImage(scaledImage)
                        } catch (e: IOException) {
                            e.printStackTrace()
                        }
                    }
                }
            }
        }
    }

    fun classifyImage(image: Bitmap) {
        try {
            val model = Chili.newInstance(requireContext())
            // Creates inputs for reference.
            val inputFeature0 =
                TensorBuffer.createFixedSize(intArrayOf(1, 150, 150, 3),
DataType.FLOAT32)
            al byteBuffer = ByteBuffer.allocateDirect(4 * imageSize * imageSize *

            teBuffer.order(ByteOrder.nativeOrder())
            al intValues = IntArray(imageSize * imageSize)
            age.getPixels(
```

```
                intValues,
                0,
                image.getWidth(),
                0,
                0,
                image.getWidth(),
                image.getHeight()
            )
            var pixel = 0
            //iterate over each pixel and extract R, G, and B values. Add those values
individually to the byte buffer.
            for (i in 0 until imageSize) {
                for (j in 0 until imageSize) {
                    val `val` = intValues[pixel++] // RGB
                    byteBuffer.putFloat((`val` shr 16 and 0xFF) * (1f / 1))
                    byteBuffer.putFloat((`val` shr 8 and 0xFF) * (1f / 1))
                    byteBuffer.putFloat((`val` and 0xFF) * (1f / 1))
                }
            }
            inputFeature0.loadBuffer(byteBuffer)

            val outputs = model.process(inputFeature0)
            val outputFeature0 = outputs.getOutputFeature0AsTensorBuffer()
            val confidences = outputFeature0.floatArray

            var maxPos = 0
            var maxConfidence = 0f
            for (i in confidences.indices) {
                if (confidences[i] > maxConfidence) {
                    maxConfidence = confidences[i]
                    maxPos = i
                }
            }
            val classes = arrayOf("curl leaf", "healthy", "leaf spot", "whitefly",
"yellowish")
            val result: TextView = binding.result
            hasilPrediksi = classes[maxPos]
            result!!.text = hasilPrediksi
//            result!!.text = "curl leaf" <<--- UBAH INI
            val detail: Button = binding.button5
            detail.setVisibility(View.VISIBLE)
            // Releases model resources if no longer used.
            model.close()
        `   tch (e: IOException) {
            TODO Handle the exception
```

```
    override fun onDestroyView() {
        super.onDestroyView()
        _binding = null
    }
}
```

Lampiran 28 Program Android (AndroidManifes.kt)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission
android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CAMERA" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.New"
        tools:targetApi="31">
        <activity
            android:name=".OutputActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />
    </application>

</manifest>
```

Lampiran 29 Program TensorFlow

```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)

from zipfile import ZipFile
import os
with ZipFile('/content/chili.zip', 'r') as zipObj:
  zipObj.extractall('/content/chili')

import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.python.keras.layers import Dense, Flatten
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam

import pathlib

data_dir = pathlib.Path('/content/chili/chili')

print(data_dir)

healthy = list(data_dir.glob('healthy/*'))
print(healthy[0])
PIL.Image.open(str(healthy[0]))

img_height,img_width=150,150
batch_size=32
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="training",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)

tf.keras.preprocessing.image_dataset_from_directory(
,
on_split=0.2,
```

```
  subset="validation",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)

class_names = train_ds.class_names
print(class_names)

class_namesv = val_ds.class_names
print(class_names)

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
  for i in range(6):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images[i].numpy().astype("uint8"))
    plt.title(class_names[labels[i]])
    plt.axis("off")


resnet_model = Sequential()

pretrained_model= tf.keras.applications.resnet50.ResNet50(include_top=False,
                   input_shape=(150,150,3),
                   pooling='avg',classes=5,
                   weights='imagenet')
for layer in pretrained_model.layers:
        layer.trainable=False

resnet_model.add(pretrained_model)
resnet_model.add(Flatten())
resnet_model.add(Dense(512, activation='relu'))
resnet_model.add(tf.keras.layers.Dropout(0.2))
resnet_model.add(Dense(128, activation='relu'))
resnet_model.add(tf.keras.layers.Dropout(0.1))

resnet_model.add(Dense(5, activation='softmax'))

resnet_model.summary()

resnet_model.compile(optimizer=Adam(lr=0.001),loss='sparse_categorical_crosse
                     etrics=['accuracy'])

                     Callback(tf.keras.callbacks.Callback):
                     efine the correct function signature for on_epoch_end
                     on_epoch_end(self, epoch, logs={}):
```

```
            if logs.get('val_accuracy') is not None and logs.get('val_accuracy') > 0.80:
                print("\nReached!")

                # Stop training once the above condition is met
                self.model.stop_training = True

callbacks = myCallback()

epochs=100
history = resnet_model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs,
  callbacks=[callbacks]

)

fig1 = plt.gcf()
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.axis(ymin=0.4,ymax=1)
plt.grid()
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['train', 'validation'])
plt.show()

converter = tf.lite.TFLiteConverter.from_keras_model(resnet_model)
tflite_model = converter.convert()

# Save the TensorFlow Lite model to a file
with open("chilip150.tflite", "wb") as f:
    f.write(tflite_model)
```