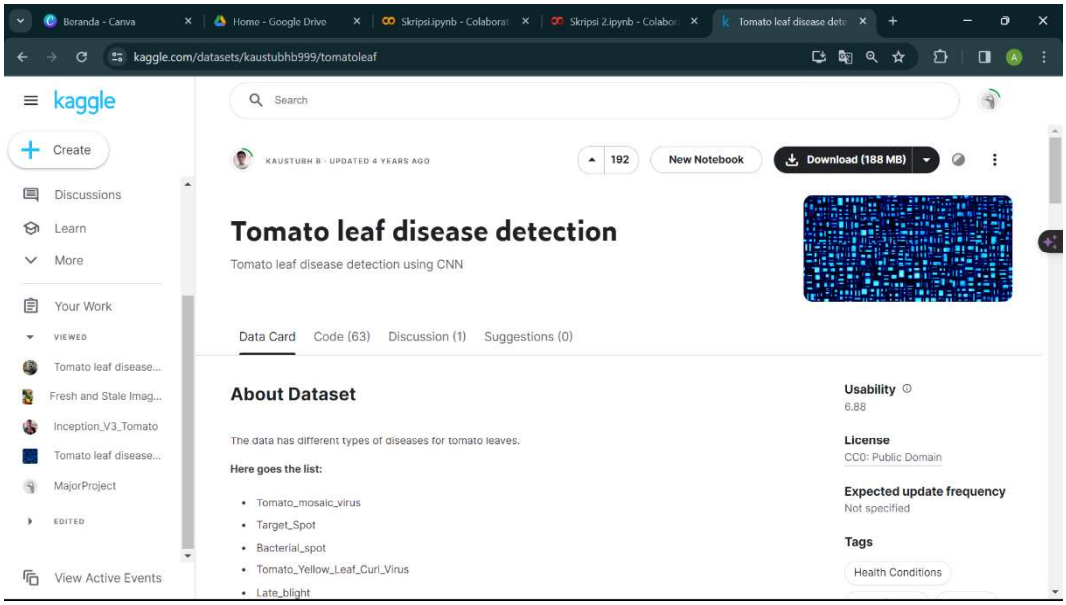
LAMPIRAN

Lampiran 1 Sumber *Dataset*



Lampiran 2 Kode Program Import Library dan *Dataset*

```
# install library
! pip install -q kaggle
# import library
import numpy as np
import pandas as pd
import seaborn as sns
import zipfile
from PIL import Image
from PIL import ImageEnhance
from skimage.io import imread
from google.colab import files
import matplotlib.pyplot as plt
import os, random, pathlib, warnings, itertools, math
warnings.filterwarnings("ignore")

import tensorflow as tf
import tensorflow.keras.backend as K
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score, f1_score
from tensorflow.keras import layers
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras import models
from tensorflow.keras.models import Model
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.applications.inception_v3 import
InceptionV3,preprocess_input
from tensorflow.keras.layers import Dense, Flatten,
GlobalAveragePooling2D, Dense, Dropout

files.upload()  # Upload the Kaggle API token JSON file

!mkdir ~/.kaggle
! cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
! kaggle datasets download -d kaustubhb999/tomatoleaf
! unzip /content/tomatoleaf.zip
```

Lampiran 3 Kode Program Preprocessing Data

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range = 40,
    width_shift_range = 0.2,
    height_shift_range = 0.2,
    fill_mode='nearest',
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True,
    validation_split=0.2 # Trainingnya 80% Validasinya 20%
(validasi = test),
)

test_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2)

training_set = train_datagen.flow_from_directory(
    train_folder,
    class_mode='categorical',
    batch_size = 16,
    target_size=(224, 224),
    shuffle=True,
    subset='training')
test_set = test_datagen.flow_from_directory(
    train_folder,
    batch_size = 16,
    class_mode='categorical',
    target_size=(224, 224),
    subset='validation')
```

Lampiran 4 Kode Program Pembuatan Model CNN dan Training Data

```
pre_trained_model = InceptionV3(input_shape = (224, 224, 3),
                                include_top = False, # Leave out the last
fully connected layer
                                weights = 'imagenet')

for layer in pre_trained_model.layers[:249]:
    layer.trainable = False
for layer in pre_trained_model.layers[249:]:
    layer.trainable = True

class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs={}):
    if logs.get('accuracy')>=0.999:
      print("\nReached 99.9% accuracy so cancelling training!")
      self.model.stop_training = True

# Flatten the output layer to 1 dimension
x = layers.Flatten()(pre_trained_model.output)
# Add a fully connected layer with 1,024 hidden units and ReLU activation
x = layers.Dense(1024, activation='relu')(x)
# Add a dropout rate of 0.2
x = layers.Dropout(0.2)(x)
# Add a final sigmoid layer for classification
x = layers.Dense  (10, activation='sigmoid')(x)

model = Model( pre_trained_model.input, x)
model.compile(optimizer = RMSprop(learning_rate=0.0001),
              loss = 'binary_crossentropy',
              metrics = ['accuracy'])
model.summary()

callbacks = myCallback()
history = model.fit(
    training_set,
    validation_data=test_set,
    epochs=20,
    steps_per_epoch=100,
    validation_steps=50,
    verbose = 2,
    callbacks=[callbacks]
)
saved_model_path = "./model_epoch100.h5"

# YOUR CODE HERE
model.save(saved_model_path)
```

Lampiran 5 Kode Program Tabel Confusion Matrix

```
def labels_confusion_matrix(validation_folder):

    folder_path=validation_folder

    mapping={}
    for i,j in enumerate(sorted(os.listdir(folder_path))):
        mapping[j]=i

    files=[]
    real=[]
    predicted=[]

    for i in os.listdir(folder_path):

        true=os.path.join(folder_path,i)
        true=true.split('/')[-1]
        true=mapping[true]

        for j in os.listdir(os.path.join(folder_path,i)):

            img_ = image.load_img(os.path.join(folder_path,i,j),
target_size=(224,224))
            img_array = image.img_to_array(img_)
            img_processed = np.expand_dims(img_array, axis=0)
            img_processed /= 255.
            prediction = model.predict(img_processed)
            index = np.argmax(prediction)

            predicted.append(index)
            real.append(true)

    return (real,predicted)

class_names = np.array(['Tomato___Bacterial_spot',
'Tomato___Early_blight', 'Tomato___Late_blight', 'Tomato___Leaf_Mold',
"Tomato___Septoria_leaf_spot",
    'Tomato___Spider_mites Two-spotted_spider_mite',
'Tomato___Target_Spot', 'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
    'Tomato___Tomato_mosaic_virus', 'Tomato___healthy'])

def print_confusion_matrix(real,predicted):
    total_output_labels = 10
    cmap="Blues"
    cm_plot_labels = class_names

    cm = confusion_matrix(y_true=real, y_pred=predicted)
    df_cm = pd.DataFrame(cm,cm_plot_labels,cm_plot_labels)
    sns.set(font_scale=1.2) # for label size
    plt.figure(figsize = (15,10))
    s=sns.heatmap(df_cm,fmt="d", annot=True,cmap=cmap) # font size

    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.savefig('confusion_matrix.png')
    plt.show()


y_true,y_pred=labels_confusion_matrix(validation_folder)
print_confusion_matrix(y_true,y_pred)

# Calculate the metrics
accuracy = accuracy_score(y_true, y_pred)
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

metrics_dict = {
    'Metric': ['Accuracy', 'Precision', 'Recall', 'F1-Score'],
    'Value': [accuracy, precision, recall, f1]
}
metrics_df = pd.DataFrame(metrics_dict)
display(metrics_df)
```

Lampiran 6 Kode Program Webssite Menggunakan Framework Flask

```
from flask import Flask, request, jsonify, render_template
from keras.models import load_model
from keras.preprocessing.image import img_to_array, load_img
import numpy as np
import os
import uuid
import logging
from PIL import Image
import cv2

app = Flask(__name__)

# Load the pre-trained model in the global scope
model = load_model('TomatoInceptionV3.h5')

UPLOAD_FOLDER = 'static/uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

CLASS_NAMES = [
    'Tomato___Bacterial_spot', 'Tomato___Early_blight',
'Tomato___Late_blight',
    'Tomato___Leaf_Mold', 'Tomato___Septoria_leaf_spot',
'Tomato___Spider_mites Two-spotted_spider_mite',
    'Tomato___Target_Spot', 'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
'Tomato___Tomato_mosaic_virus',
    'Tomato___healthy'
]

CLASS_DESCRIPTIONS = {
    'Tomato___Bacterial_spot': 'Bercak Daun Bakteri/Bacterial Spot',
    'Tomato___Early_blight': 'Bercak Kering Altemaria/Early Blight',
    'Tomato___Late_blight': 'Hawar Daun/Late Blight',
    'Tomato___Leaf_Mold': 'Kapang Daun/Leaf Mold',
    'Tomato___Septoria_leaf_spot': 'Bercak Daun Septoria/Septoria Spot',
    'Tomato___Spider_mites Two-spotted_spider_mite': 'Bercak Daun akibat
Gigitan Serangga/Spider Mites',
    'Tomato___Target_Spot': 'Bercak Daun Jamur/Target Spot',
    'Tomato___Tomato_Yellow_Leaf_Curl_Virus': 'Daun Kuning dan
Keriting/Yellow Leaf Curl Virus',
    'Tomato___Tomato_mosaic_virus': 'Virus Mosaic',
    'Tomato___healthy': 'Sehat'
}

# Configure logging
logging.basicConfig(level=logging.DEBUG)

# Preprocess image function
def preprocess_image(image_path):
    image = load_img(image_path, target_size=(224, 224))
    image = img_to_array(image)
    image = np.expand_dims(image, axis=0)
    image = image / 255.0
    return image

# Check if image is grayscale
def is_grayscale(image_path):
    image = Image.open(image_path).convert('RGB')
    np_image = np.array(image)
    if np_image.ndim == 3 and np_image.shape[2] == 3:
        if np.all(np_image[:,:,0] == np_image[:,:,1]) and
np.all(np_image[:,:,1] == np_image[:,:,2]):
            return True
    return False

# Check if image is a tomato leaf
def is_tomato_leaf(image_path):
    image = Image.open(image_path).convert('RGB')
    np_image = np.array(image)

    hsv_image = Image.fromarray(np_image).convert('HSV')
    np_hsv_image = np.array(hsv_image)
```

```
    lower_green = np.array([35, 40, 40])
    upper_green = np.array([85, 255, 255])

    mask = cv2.inRange(np_hsv_image, lower_green, upper_green)
    green_percentage = (np.sum(mask) / 255) / (mask.shape[0] *
mask.shape[1])
    return green_percentage > 0.15

#routes
@app.route('/')
def index():
    return render_template("index.html")

@app.route('/check')
def check():
    return render_template('check.html')

@app.route('/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        app.logger.debug('No file part in request.files')
        return jsonify({'error': 'No file part'})

    file = request.files['file']

    if file.filename == '':
        app.logger.debug('No selected file')
        return jsonify({'error': 'No selected file'})

    if file:
        filename = str(uuid.uuid4()) + os.path.splitext(file.filename)[1]
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(filepath)

        if is_grayscale(filepath):
            app.logger.debug('Jangan upload gambar yang berwarna hitam
putih')
            return jsonify({'error': 'Jangan upload gambar yang berwarna
hitam putih'})

        if not is_tomato_leaf(filepath):
            app.logger.debug('Gambar yang diupload bukan daun tomat')
            return jsonify({'error': 'Gambar yang diupload bukan daun
tomat'})

        image = preprocess_image(filepath)
        app.logger.debug(f'Preprocessed image shape: {image.shape}')

        prediction = model.predict(image)
        app.logger.debug(f'Raw prediction: {prediction}')

        predicted_class = CLASS_NAMES[np.argmax(prediction)]
        confidence = np.max(prediction) * 100
        app.logger.debug(f'Predicted class: {predicted_class} with
confidence: {confidence:.2f}%')

        predicted_description = CLASS_DESCRIPTIONS.get(predicted_class,
'Unknown')

        # Check if the prediction is valid
        if predicted_class not in CLASS_NAMES:
            app.logger.debug('Uploaded image is not recognized by the
model')
            return jsonify({'error': 'Uploaded image is not recognized by
the model'})

        return jsonify({'image_url': '/' + filepath, 'prediction':
predicted_description, 'confidence': confidence})

if __name__ == '__main__':
    if not os.path.exists(UPLOAD_FOLDER):
        os.makedirs(UPLOAD_FOLDER)
    app.run(debug=True)
```

Lampiran 7 Kode Program Website Halaman Beranda

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>Sistem Deteksi Penyakit Daun Tomat</title>

  <!-- CSS FILES -->
  <link href="../static/css/bootstrap.min.css" rel="stylesheet">
  <link href="../static/css/bootstrap-icons.css" rel="stylesheet">
  <link href="../static/css/style.css" rel="stylesheet">
</head>

<body id="section_1">

  <header class="site-header">
    <div class="container"></div>
  </header>

  <nav class="navbar navbar-expand-lg bg-light shadow-lg">
    <div class="container">
      <a class="navbar-brand" href="{{ url_for('index') }}">
        <span>
          Tomato Leaf Check
          <small>Deteksi Penyakit Tomat Melalui Citra Daun</small>
        </span>
      </a>

      <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav"
        aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a class="nav-link click-scroll" href="#top">Beranda</a>
          </li>

          <li class="nav-item">
            <a class="nav-link click-scroll" href="#section_2">Jenis
Penyakit Daun Tomat</a>
          </li>

          <li class="nav-item ms-3">
            <a class="nav-link custom-btn custom-border-btn btn" href="{{
url_for('check') }}">Cek Penyakit
              Daun Tomat</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

  <main>

    <section class="hero-section hero-section-full-height">
      <div class="container-fluid">
        <div class="row">

          <div class="col-lg-12 col-12 p-0">
            <div id="hero-slide" class="carousel carousel-fade slide"
data-bs-ride="carousel">
              <div class="carousel-inner">
                <div class="carousel-item active">
                  <img src="../static/images/slide/foto-tomat-1.jpg"
```

```
class="carousel-image img-fluid" alt="...">
                  <div class="carousel-caption d-flex flex-column
justify-content-end">
                    <h1>Selamat Datang</h1>
                    <p>Website Deteksi Penyakit Daun Tomat</p>
                  </div>
                </div>

                <div class="carousel-item">
                  <img src="../static/images/slide/foto-tomat-2.jpg"
class="carousel-image img-fluid" alt="...">
                  <div class="carousel-caption d-flex flex-column
justify-content-end">
                    <h1>Selamat Datang</h1>
                    <p>Website Deteksi Penyakit Daun Tomat</p>
                  </div>
                </div>

                <div class="carousel-item">
                  <img src="../static/images/slide/foto-tomat-3.jpg"
class="carousel-image img-fluid" alt="...">
                  <div class="carousel-caption d-flex flex-column
justify-content-end">
                    <h1>Selamat Datang</h1>
                    <p>Website Deteksi Penyakit Daun Tomat</p>
                  </div>
                </div>
              </div>

              <button class="carousel-control-prev" type="button" data-
bs-target="#hero-slide" data-bs-slide="prev">
                <span class="carousel-control-prev-icon" aria-
hidden="true"></span>
                <span class="visually-hidden">Previous</span>
              </button>

              <button class="carousel-control-next" type="button" data-
bs-target="#hero-slide" data-bs-slide="next">
                <span class="carousel-control-next-icon" aria-
hidden="true"></span>
                <span class="visually-hidden">Next</span>
              </button>
            </div>
          </div>

        </div>
      </div>
    </section>

    <section class="section-padding" id="section_2">
      <div class="container">
        <div class="row mb-4">

          <div class="col-lg-12 col-12 text-center mb-4">
            <h2>Jenis-Jenis Penyakit Daun Tomat</h2>
          </div>

          <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/mozaik-virus.png"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Mosaic Virus</h5>
                  <p>Penyakit Mosaic Virus pada daun tomat adalah
penyakit yang disebabkan oleh Tobacco Mosaic Virus
                    (TMV) dan Cucumber Mosaic Virus (CMV). Gejala utama
menyebabkan bercak hijau dan kuning, distorsi
                    daun, penggulungan daun, dan pertumbuhan kerdil pada
tanaman</p>
                </div>
              </div>
            </div>
          </div>
```

```
          <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/target-spot.jpg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Target Spot/Bercak Daun Karena Jamur
</h5>
                  <p>Penyakit Target Spot pada daun tomat ditandai dengan
bercak konsentris coklat tua pada daun,
                    batang, dan buah. Infeksi berat menyebabkan daun
rontok, mengurangi fotosintesis dan hasil panen</p>
                </div>
              </div>
            </div>
          </div>

          <div class="col-lg-4 col-md-6 col-12">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/bacterial-spot.jpg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Bacterial Spot/Bercak Daun Karena
Bakteri</h5>
                  <p>Penyakit Bacterial Spot pada daun tomat disebabkan
oleh bakteri yang menyerang daun, batang, dan
                    buah. Gejalanya berupa luka kecil berwarna kuning
kehijauan pada daun muda, yang biasanya tampak
                    cacat dan berubah bentuk
                  </p>
                </div>
              </div>
            </div>
          </div>
        </div>

        <div class="row mb-4">
          <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/yellow-leaf-curl.jpeg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Yellow Leaf Curl/Daun Kuning dan
Keriting</h5>
                  <p>Yellow Leaf Curl Virus (TYLCV) pada tomat
menyebabkan daun menguning, mengeriting, dan menghambat
                    pertumbuhan. Penyakit ini menyebar melalui kutu kebul
(Bemisia tabaci)</p>
                </div>
              </div>
            </div>
          </div>

          <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/late-blight.jpg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Late Blight/Busuk Daun</h5>
                  <p>Late Blight pada tomat menyebabkan bercak daun
cokelat dimulai dari pinggiran, bagian bawah daun
                    tertutup lapisan putih, dan buah memiliki noda
keriput abu-abu atau coklat</p>
                </div>
              </div>
            </div>
          </div>

          <div class="col-lg-4 col-md-6 col-12">
            <div class="custom-block-wrap">
```

```
              <img src="../static/images/causes/leaf-mold.jpg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Leaf Mold/Kapang Daun</h5>
                  <p>Leaf Mold pada tomat disebabkan oleh jamur
Cladosporium fulvum, menginfeksi daun tua terlebih
                    dahulu, muncul bercak-bercak kuning atau cokelat di
permukaan daun</p>
                </div>
              </div>
            </div>
          </div>
        </div>

        <div class="row mb-4">
          <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/early-blight.jpg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Early Blight/Bercak Kering Karena
Altemaria</h5>
                  <p>Early Blight pada daun tomat disebabkan oleh jamur
Alternaria solani, ditandai dengan bercak coklat
                    melingkar dengan tepi kuning, menyebabkan defoliasi
dan menurunkan hasil panen</p>
                </div>
              </div>
            </div>
          </div>

          <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/spider-mites.jpg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Spider Mites/Bercak Daun Karena
Serangga</h5>
                  <p>Spider mites pada daun tomat menyebabkan bintik-
bintik kecil, daun menguning, dan jaring halus.
                    Mengatasinya dengan menyemprotkan air bertekanan
tinggi atau menggunakan insektisida khusus</p>
                </div>
              </div>
            </div>
          </div>

          <div class="col-lg-4 col-md-6 col-12">
            <div class="custom-block-wrap">
              <img src="../static/images/causes/septoria-leaf-spot.jpg"
class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Septoria Leaf Spot/Bercak Daun Karena
Septoria</h5>
                  <p>Septoria Leaf Spot adalah penyakit jamur, gejalanya
berupa bercak bulat berwarna coklat dengan tepi
                    kuning pada daun bawah. Pengendaliannya meliputi
sanitasi, rotasi tanaman, dan aplikasi fungisida jika diperlukan
                  </p>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>

    <footer class="site-footer" id="section_3">
      <div class="container">
        <div class="row">
```

```
          <div class="col-lg-12 col-md-12 col-12 mx-auto">
            <h5 class="site-footer-title mb-3">Kontak Informasi &
Saran</h5>
            <p class="text-white d-flex mb-4">
              <i class="bi-telephone me-2"></i>
              <a href="tel: 120-240-9600" class="site-footer-link">
                0853-3454-4167
              </a>
            </p>
          </div>
        </div>
      </div>
    </footer>

    <!-- JAVASCRIPT FILES -->
    <script src="../static/js/jquery.min.js"></script>
    <script src="../static/js/bootstrap.min.js"></script>
    <script src="../static/js/jquery.sticky.js"></script>
    <script src="../static/js/click-scroll.js"></script>
    <script src="../static/js/counter.js"></script>
    <script src="../static/js/custom.js"></script>

</body>

</html>
```

Lampiran 8 Kode Program Website Halaman Prediksi

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>Sistem Deteksi Penyakit Daun Tomat</title>

  <!-- CSS FILES -->
  <link href="../static/css/bootstrap.min.css" rel="stylesheet">
  <link href="../static/css/bootstrap-icons.css" rel="stylesheet">
  <link href="../static/css/style.css" rel="stylesheet">
</head>

<body>

  <header class="site-header">
    <div class="container"></div>
  </header>

  <nav class="navbar navbar-expand-lg bg-light shadow-lg">
    <div class="container">
      <a class="navbar-brand" href="{{ url_for('index') }}">
        <span>
          Tomato Leaf Check
          <small>Deteksi Penyakit Tomat Melalui Citra Daun</small>
        </span>
      </a>

      <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav"
        aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a class="nav-link click-scroll" href="{{ url_for('index')
}}#section_1">Beranda</a>
          </li>

          <li class="nav-item">
            <a class="nav-link click-scroll" href="{{ url_for('index')
}}#section_2">Jenis Penyakit Daun
              Tomat</a>
          </li>

          <li class="nav-item ms-3">
            <a class="nav-link custom-btn custom-border-btn btn" href="{{
url_for('check') }}">Cek Penyakit
              Daun Tomat</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

  <main>
    <section class="upload-section">
      <div class="container">
        <div class="row">
          <div class="col-lg-8 col-12 mx-auto">
            <form id="uploadForm" class="custom-form upload-form"
action="/predict" method="post" role="form"
              enctype="multipart/form-data">
              <h3 class="mb-4">Upload Foto Daun Tomat</h3>
```

```
              <div class="row" style="justify-content: center;">
                <div class="col-lg-8 col-10">
                  <input type="file" id="fileInput" accept=".png, .jpg,
.jpeg" class="form-control" required
                    name="file">
                  <label for="fileInput" class="form-control"
style="text-align: center;">Pilih Foto (PNG, JPG,
                    JPEG)</label>
                  <img id="previewImage" src="#" alt="Preview Gambar"
style="display:none; max-width:100%;">
                </div>
                <div class="col-lg-8 col-10">
                  <button type="submit" class="form-control mt-4"
id="predictButton">
                    Prediksi
                    <div id="spinner" class="spinner-border spinner-
border-sm text-light ms-2" role="status"
                      style="display: none;">
                      <span class="visually-hidden">Loading...</span>
                    </div>
                  </button>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </section>

    <section class="result-section">
      <div class="container">
        <div class="row">
          <div class="col-lg-8 col-12 mx-auto">
            <div class="result-form">
              <h3 class="mb-4">Hasil Prediksi</h3>

              <div class="row">
                <div class="col-lg-12 col-12">
                  <div id="alertContainer" class="alert alert-danger
alert-dismissible fade show" role="alert"
                    style="display: none;">
                    <span id="alertMessage"></span>
                  </div>
                  <h5 class="mb-3">
                    <span id="diseaseMessage" style="display:
none;">Tomat Anda Terkena :</span>
                    <span id="healthyMessage" style="display:
none;">Tomat Anda :</span>
                    <span id="predictionResult" style="font-weight:
bold;"></span>
                    <span id="confidenceResult"></span>
                  </h5>
                </div>
                <div class="col-lg-12 col-12">
                  <img id="resultImage" src="#" alt="Hasil Gambar"
style="max-width:100%; display: none;">
                </div>
              </div>
            </div>

          </div>
        </div>
      </div>
    </section>
  </main>

  <footer class="site-footer">
    <div class="container">
      <div class="row">
        <div class="col-lg-12 col-md-12 col-12 mx-auto">
        </div>
      </div>
    </div>
  </footer>
```

```
  <!-- JAVASCRIPT FILES -->
  <script src="../static/js/jquery.min.js"></script>
  <script src="../static/js/bootstrap.min.js"></script>
  <script src="../static/js/jquery.sticky.js"></script>
  <script src="../static/js/counter.js"></script>
  <script src="../static/js/custom.js"></script>
  <script>
    document.getElementById('uploadForm').addEventListener('submit',
async function (event) {
      event.preventDefault();

      const form = event.target;
      const formData = new FormData(form);

      // Reset and hide alert
      const alertContainer = document.getElementById('alertContainer');
      alertContainer.style.display = 'none';
      alertContainer.classList.remove('show');
      document.getElementById('alertMessage').innerText = '';

      // Reset prediction result
      document.getElementById('predictionResult').innerText = '';
      document.getElementById('confidenceResult').innerText = '';
      document.getElementById('resultImage').style.display = 'none';
      document.getElementById('diseaseMessage').style.display = 'none';
      document.getElementById('healthyMessage').style.display = 'none';

      // Show spinner
      document.getElementById('spinner').style.display = 'inline-block';
      document.getElementById('predictButton').disabled = true;

      const startTime = Date.now();

      try {
        const response = await fetch('/predict', {
          method: 'POST',
          body: formData
        });

        const result = await response.json();

        const elapsedTime = Date.now() - startTime;
        const remainingTime = 2000 - elapsedTime;

        setTimeout(() => {
          // Hide spinner
          document.getElementById('spinner').style.display = 'none';
          document.getElementById('predictButton').disabled = false;

          if (result.error) {
            document.getElementById('alertMessage').innerText =
result.error;
            alertContainer.style.display = 'block';
            alertContainer.classList.add('show');
          } else {
            alertContainer.style.display = 'none';
            if (result.prediction === 'Sehat/Tidak Terkena Penyakit') {
              document.getElementById('healthyMessage').style.display =
'inline';
              document.getElementById('diseaseMessage').style.display =
'none';
            } else {
              document.getElementById('diseaseMessage').style.display =
'inline';
              document.getElementById('healthyMessage').style.display =
'none';
            }
            document.getElementById('predictionResult').innerText =
result.prediction;
            document.getElementById('confidenceResult').innerText =
              `(Persentase Kecocokan:
${result.confidence.toFixed(2)}%)`;
            document.getElementById('resultImage').src =
```

```
result.image_url;
            document.getElementById('resultImage').style.display =
'inline-block';
          }
        }, Math.max(remainingTime, 0));
      } catch (error) {
        console.error('Error:', error);
        // Hide spinner in case of error
        document.getElementById('spinner').style.display = 'none';
        document.getElementById('predictButton').disabled = false;
        document.getElementById('alertMessage').innerText = 'Terjadi
kesalahan saat memproses gambar.';
        alertContainer.style.display = 'block';
        alertContainer.classList.add('show');

        // Clear prediction result in case of error
        document.getElementById('predictionResult').innerText = '';
        document.getElementById('confidenceResult').innerText = '';
        document.getElementById('resultImage').style.display = 'none';
        document.getElementById('diseaseMessage').style.display = 'none';
        document.getElementById('healthyMessage').style.display = 'none';
      }
    });

    document.getElementById('fileInput').addEventListener('change',
function (event) {
      const file = event.target.files[0];
      if (file) {
        const reader = new FileReader();
        reader.onload = function (e) {
          document.getElementById('previewImage').src = e.target.result;
          document.getElementById('previewImage').style.display =
'inline-block';
        };
        reader.readAsDataURL(file);
      }
    });

    // Ensure alert can be shown again after being closed
    document.querySelector('.btn-close').addEventListener('click',
function () {
      const alertContainer = document.getElementById('alertContainer');
      alertContainer.style.display = 'none';
      alertContainer.classList.remove('show');
    });
  </script>
</body>

</html>
```