

LAMPIRAN

Lampiran 1. Pedoman Wawancara

Narasumber:

1. I Made Sutiawan, S.P. (Penyuluh Pertanian Lapangan (PPL) Desa Pancasari & Desa Pegayaman)
2. I Nyoman Mara (Petani Tanaman Paprika Desa Pancasari & Pemilik Smart Green House Sayram Garden)

Tanggal: 26 Desember 2024

Lokasi: Banjar Karma, Desa Pancasari, Kecamatan Sukasada, Kabupaten Buleleng.

Tabel Wawancara dengan I Made Sutiawan, S.P.

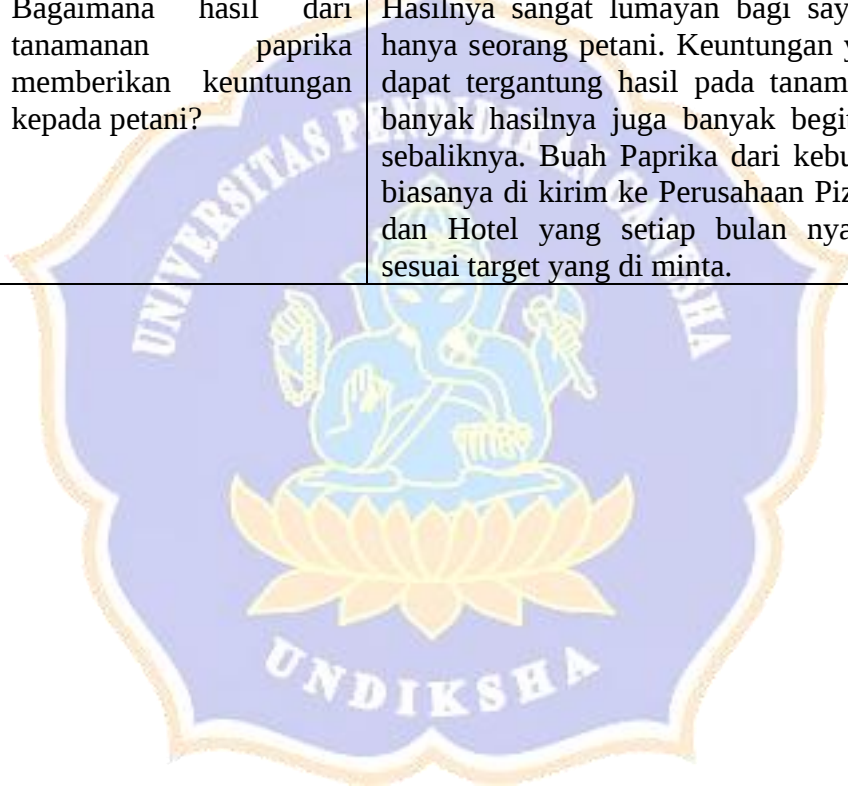
No	Pertanyaan	Hasil Wawancara
1	Apakah anda tau di mana terdapat pertanian tanaman paprika di kabupaten buleleng?	Ya, pertanian tanaman paprika banyak ada di Desa Pancasari
2	Apakah benar yang diberitakan oleh Dinas Pertanian terkait serangan hama yang ada di pertanian di Desa Pancasari?	Ya benar, Seperti yang sudah di beritakan di website resmi kami bahwa berita tersebut benar adanya
3	Apakah anda tau petani yang mengetahui terkait permasalahan tersebut di Desa Pancasari?	Ya saya tau, Namanya adalah Nyoman Mara merupakan petani paprika yang bersal dari Desa Pancasari dan yang memberitau saya terkait permasalahan tersebut yang ada pada website resmi kami
4	Apakah bapak berkenan selaku PPL di Dinas Pertanian Buleleng saya masukan sebagai narasumber?	Ya, boleh untuk narasumbernya gunakan saya dan petani nya (Nyoman Mara)

Tabel Wawancara dengan I Nyoman Mara

No	Pertanyaan	Hasil Wawancara
1	Apakah Benar di Desa Pancasari ini menjadi desa yang memiliki banyak	Ya benar, Di Desa Pancasari sangat cocok di tanami tanaman paprika karena memiliki berberapa faktor yang tidak di miliki daerah

No	Pertanyaan	Hasil Wawancara
	pertanian tanaman paprika?	lain seperti kualitas tanah,kelembaban udara,pencahayaannya membuat tanaman paprika bisa tumbuh subur di daerah ini yang membuat para petani disini banyak melakukan budidaya tanaman paprika.
2	Apa yang menjadi alasan anda berbudidaya tanaman paprika?	Alasan saya berbudidaya tanaman paprika karena harga jual nya yang lumayan stabil dan faktor lokasi yang memadai untuk berbudidaya tanaman paprika.
3	Bagaimana Sejarah yang anda ketahui terkait pertanian tanaman paprika di Desa Pancasari?	Sejarah yang saya tau,saya menjadi pelopor petani untuk berbudidaya tanaman paprika di desa ini sejak tahun 2001 dengan mendirikan kelompok petani muda keren dan membangun perkebunan paprika dengan nama sayram garden dan masih beroperasi sampai sekarang.
4	Bagaimana cara anda untuk berbudidaya tanaman paprika?	Saya sebagai petani berbudidaya tanaman paprika dengan metode SGH dimulai dari fase pembibitan sampai fase panen di lakukan di SGH ini Di mana setiap hari di kontrol perkembangannya di mulai dari suhu,pencayaan,kelembaban dan lainnya menggunakan teknologi IoT (Internet Of Thing) untuk membantu tanaman paprika berada pada fase pertumbuhan yang ideal.
5	Apakah ada pihak terkait yang turut membantu anda dalam berbudidaya tanaman paprika?	Ya jelas ada, SGH yang digunakan sekarang diinisiasi Direktorat Jenderal Prasarana dan Sarana Pertanian (Ditjen PSP) Kementerian Pertanian (Kementan), SGH ini merupakan program terobosan untuk membangun pertanian modern yang sudah di realisasikan di Desa Pancasari dan diketahui oleh pihak Dinas terkait seperti Dinas Pertanian Kabupaten Buleleng.
6	Apakah benar ada nya permasalahan serangan penyakit hama pada daun tanaman paprika di Desa Pancasari?	Ya benar pernah terjadi serangan hama terutama hama thrips pada musim hujan. Sering membuat daun pada tanaman paprika menjadi keriput dan hasil dari buah paprika menjadi sedikit dari pada biasanya.
7	Berapa hasil setiap panen untuk tanaman paprika di kebun bapak?	Hasil setiap panen tentunya bermacam macam hasilnya tergantung banyaknya buah yang sehat jika musim panen biasanya bisa sampai 300 kiloan per minggu namun paling apes nya seperti kemarin banyak tanaman yang terkena penyakit bisa turun sampai 100 kiloan saja perminggu.

No	Pertanyaan	Hasil Wawancara
8	Apakah sudah ada sebelumnya inovasi terkait deteksi penyakit daun paprika dalam SGH yang anda miliki dalam berbudidaya tanaman paprika?	Sejauh ini belum ada terutama di Lokasi ini. Saya hanya melihat tumbuhannya biasanya kalau sudah terkena penyakit baru saya sadar dan baru saya berikan pengobatan pada tanaman yang terkena penyakit.
9	Apa saja penanggulangan yang anda lakukan untuk mencegah permasalahan penyakit tersebut?	Saya biasanya melakukan penanganan menggunakan obat dan pupuk untuk mencegah penyakit yang ada pada tanaman tidak menyebar ke tanaman yang lain. Kalau sudah tidak bisa di obati tanamannya saya potong.
10	Bagaimana hasil dari tanaman paprika memberikan keuntungan kepada petani?	Hasilnya sangat lumayan bagi saya yang hanya seorang petani. Keuntungan yang di dapat tergantung hasil pada tanaman jika banyak hasilnya juga banyak begitu juga sebaliknya. Buah Paprika dari kebun saya biasanya di kirim ke Perusahaan Pizza Hut dan Hotel yang setiap bulan nya harus sesuai target yang di minta.



Lampiran 2. Dokumentasi



Gambar 4
Dokumentasi hasil pertanian
Paprika di kebun Nyoman
Mara



Gambar 2
Dokumentasi Wawancara
bersama Nyoman Mara



Gambar 3
Dokumentasi Foto bersama
Nyoman Mara



Gambar 1
Dokumentasi Pengambilan
Data di kebun Nyoman Mara

Lampiran 3. Kode Augmentasi

```
import os
import cv2
import numpy as np
from glob import glob

# Path input & output
input_dir = r'/content/drive/MyDrive/Skripsi Final'/Data Asli'
output_dir = r'/content/drive/MyDrive/Skripsi Final/Data Augmentasi'
classes = ['Sehat', 'Hama', 'Jamur', 'Virus']

# Buat folder output
for cls in classes:
    os.makedirs(os.path.join(output_dir, cls), exist_ok=True)

# Fungsi rotasi dengan sudut tertentu
def rotate_image(image, angle):
    h, w = image.shape[:2]
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, angle, 1.0)
    rotated = cv2.warpAffine(image, M, (w, h),
borderMode=cv2.BORDER_REFLECT)
    return rotated

# Fungsi utama augmentasi
def augment_image(image):
    augmented_images = []

    # Resize ke 256x256
    if image.shape[:2] != (256, 256):
        image = cv2.resize(image, (256, 256))

    # 1. Flip horizontal
    flip_h = cv2.flip(image, 1)
    augmented_images.append(flip_h)

    # 2. Flip vertical
    flip_v = cv2.flip(image, 0)
    augmented_images.append(flip_v)

    # 3-10. Rotate dengan sudut tetap
    angles = [15, 30, 45, 60, 90, 120, 180, 270]
    for angle in angles:
        rotated = rotate_image(image, angle)
        augmented_images.append(rotated)
```

```

    return augmented_images

# Proses semua gambar
for cls in classes:
    input_cls_path = os.path.join(input_dir, cls)
    output_cls_path = os.path.join(output_dir, cls)

    for img_path in glob(os.path.join(input_cls_path, '*.jpg')): #
        ubah ke *.* jika banyak ekstensi
        filename = os.path.splitext(os.path.basename(img_path))[0]
        image = cv2.imread(img_path)
        if image is None:
            print(f"Gagal membaca: {img_path}")
            continue

        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        aug_images = augment_image(image)

        # Simpan hasil
        for idx, aug in enumerate(aug_images):
            out_name = f'{filename}_aug{idx+1}.jpg'
            out_path = os.path.join(output_cls_path, out_name)
            cv2.imwrite(out_path, cv2.cvtColor(aug,
cv2.COLOR_RGB2BGR))

print("Augmentasi selesai. Setiap gambar ditambah 10 variasi.")

```



Lampiran 4. Kode CNN

```
# =====
# 1. Import Library
# =====
import os
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Dropout
from tensorflow.keras.callbacks import ModelCheckpoint

# =====
# 2. Path Dataset dan Konfigurasi
# =====
base_dir = r'/content/drive/MyDrive/Skripsi Final'
train_dir = os.path.join(base_dir, 'train')
test_dir = os.path.join(base_dir, 'test')
img_size = (256, 256) # Disesuaikan dengan ukuran gambar asli
batch_size = 32
num_classes = 4 # Hama, Jamur, Sehat, Virus

# =====
# 3. Data Generator untuk Train dan Test
# =====
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_data = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

test_data = test_datagen.flow_from_directory(
    test_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
```

```

        shuffle=False
    )

# =====
# 4. Arsitektur CNN
# =====
def build_model():
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256,
3)),
        MaxPooling2D(2, 2),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D(2, 2),
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
    return model

# =====
# 5. Fungsi Evaluasi dan Visualisasi
# =====
def evaluate_model(model, test_data, title, save_path=None):
    test_data.reset()
    y_pred = model.predict(test_data)
    y_pred_classes = np.argmax(y_pred, axis=1)
    y_true = test_data.classes
    labels = list(test_data.class_indices.keys())

    # Confusion matrix
    cm = confusion_matrix(y_true, y_pred_classes)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=labels)
    plt.figure(figsize=(5, 5))
    disp.plot(cmap=plt.cm.Blues, values_format='d')
    plt.title(f'Confusion matrix - {title}')
    plt.show()

    # Classification Report
    print(f"=== Classification Report: {title} ===")
    print(classification_report(y_true, y_pred_classes,
target_names=labels))

    # Simpan fitur prediksi dan label ke CSV (opsional)

```

```

        if save_path:
            df = pd.DataFrame(y_pred, columns=[f'Prob_{label}' for label
in labels])
            df['y_pred'] = y_pred_classes
            df['y_true'] = y_true
            df.to_csv(save_path, index=False)
            print(f"Prediksi dan label disimpan di: {save_path}")

def plot_history(history, title):
    plt.figure(figsize=(10, 4))

    # Akurasi
    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'], label='Train')
    plt.plot(history.history['val_accuracy'], label='Val')
    plt.title(f'{title} - Akurasi')
    plt.xlabel('Epoch')
    plt.ylabel('Akurasi')
    plt.legend()

    # Loss
    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'], label='Train')
    plt.plot(history.history['val_loss'], label='Val')
    plt.title(f'{title} - Loss')
    plt.xlabel('Epoch')
    plt.ylabel('Loss')
    plt.legend()

    plt.tight_layout()
    plt.show()

# =====
# 8. Training dan Evaluasi Model (Early Stopping)
# =====
from tensorflow.keras.callbacks import EarlyStopping

# Membuat ulang model
model_early = build_model()

# Callback EarlyStopping: menghentikan pelatihan jika val_loss tidak
membaik selama 5 epoch
early_stop = EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)

# ModelCheckpoint opsional jika ingin menyimpan model terbaik

```

```
checkpoint_early = ModelCheckpoint('cnn_earlystop_best_model.h5',  
monitor='val_loss', save_best_only=True)  
  
# Training model  
history_early = model_early.fit(  
    train_data,  
    validation_data=test_data,  
    epochs=100, # Batas maksimal epoch, bisa dihentikan lebih awal  
    callbacks=[early_stop, checkpoint_early],  
    verbose=1  
)  
  
# Evaluasi hasil model dengan test data  
evaluate_model(model_early, test_data, "Early Stopping",  
save_path="cnn_result_earlystop.csv")  
  
# Visualisasi hasil training  
plot_history(history_early, 'Early Stopping')
```



Lampiran 5. Kode SVM

```
import os
import cv2
import numpy as np
from sklearn import svm
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import joblib

# =====
# 1. Ekstraksi Fitur
# =====
def extract_features(image_path):
    img = cv2.imread(image_path)
    img = cv2.resize(img, (256, 256)) # Resize ke 256x256

    # RGB Mean & Std
    rgb = img.reshape(-1, 3)
    rgb_mean = np.mean(rgb, axis=0)
    rgb_std = np.std(rgb, axis=0)

    # HSL Mean & Std
    hsl = cv2.cvtColor(img, cv2.COLOR_BGR2HLS).reshape(-1, 3)
    hsl_mean = np.mean(hsl, axis=0)
    hsl_std = np.std(hsl, axis=0)

    # Sobel Mean
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
    sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
    sobel_mag = np.sqrt(sobelx**2 + sobely**2)
    sobel_mean = np.mean(sobel_mag)

    # Gabung semua fitur
    features = np.concatenate([
        rgb_mean, rgb_std,
        hsl_mean, hsl_std,
        [sobel_mean]
    ])
    return features

# =====
# 2. Load Dataset
# =====
def load_dataset(folder_path):
```

```

features = []
labels = []
label_map = {'Hama': 0, 'Jamur': 1, 'Sehat': 2, 'Virus': 3}

for class_name in os.listdir(folder_path):
    class_folder = os.path.join(folder_path, class_name)
    if not os.path.isdir(class_folder):
        continue
    for img_name in os.listdir(class_folder):
        img_path = os.path.join(class_folder, img_name)
        try:
            feat = extract_features(img_path)
            features.append(feat)
            labels.append(label_map[class_name])
        except:
            print(f"Gagal memproses gambar: {img_path}")

return np.array(features), np.array(labels)

# =====
# 3. Path Dataset
# =====
train_path = '/content/drive/MyDrive/Skripsi Final/train'
test_path = '/content/drive/MyDrive/Skripsi Final/test'

# =====
# 4. Load dan Normalisasi Data
# =====
print("Memuat dan mengekstraksi data...")
X_train, y_train = load_dataset(train_path)
X_test, y_test = load_dataset(test_path)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# =====
# 5. Training SVM (Kernel Polynomial)
# =====
print("Melatih model SVM kernel polynomial...")
svm_model = svm.SVC(kernel='poly', degree=3, C=1.0, gamma='scale')
svm_model.fit(X_train_scaled, y_train)

# =====
# 6. Evaluasi Model
# =====
y_pred = svm_model.predict(X_test_scaled)

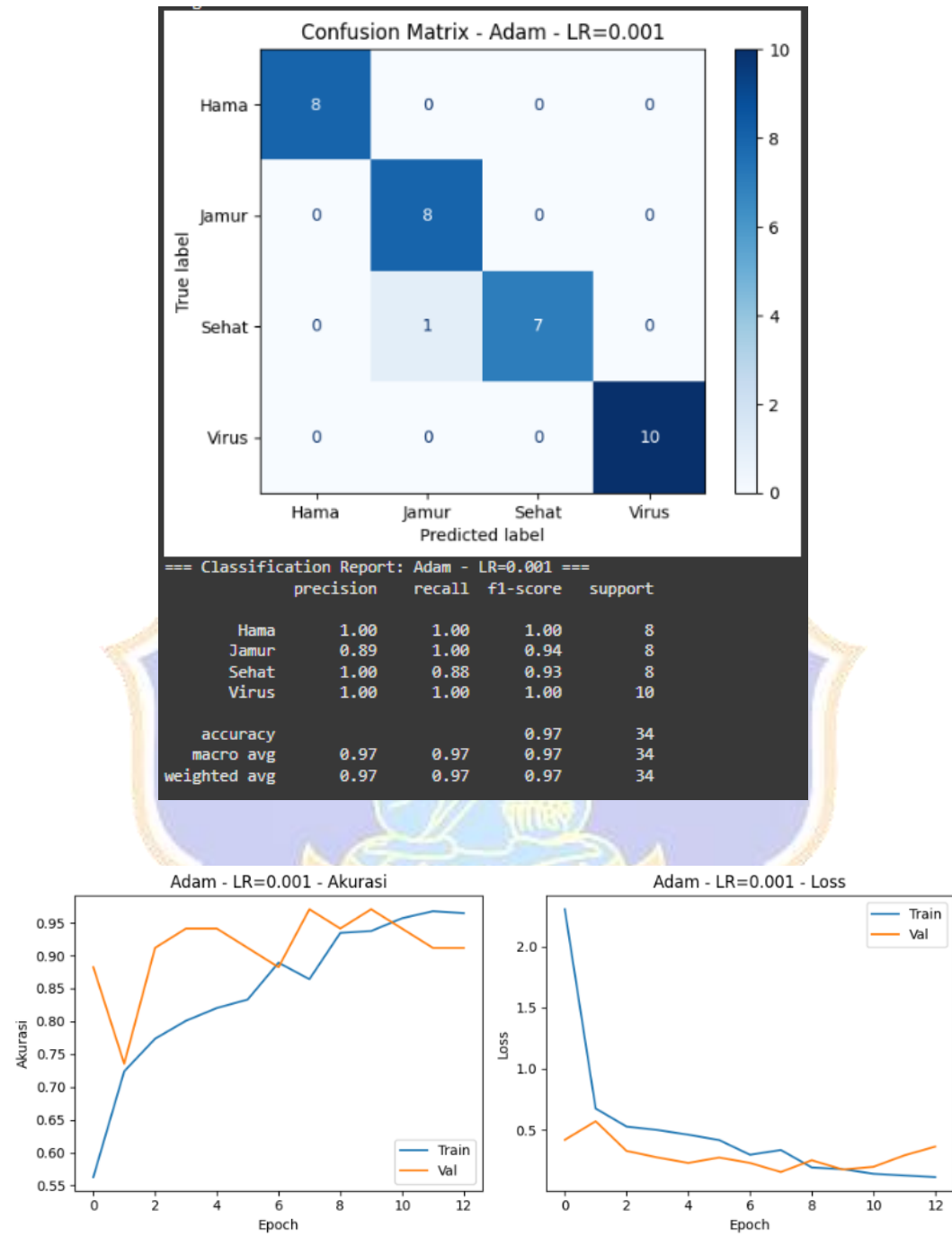
```

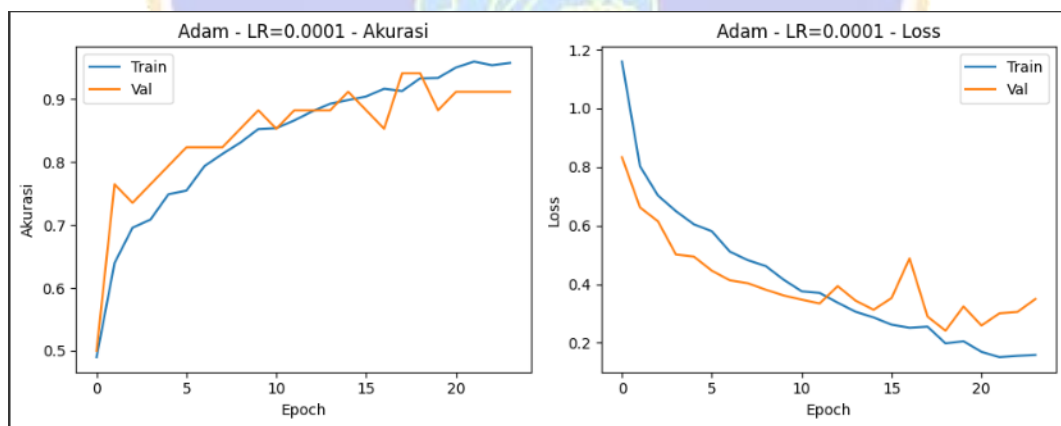
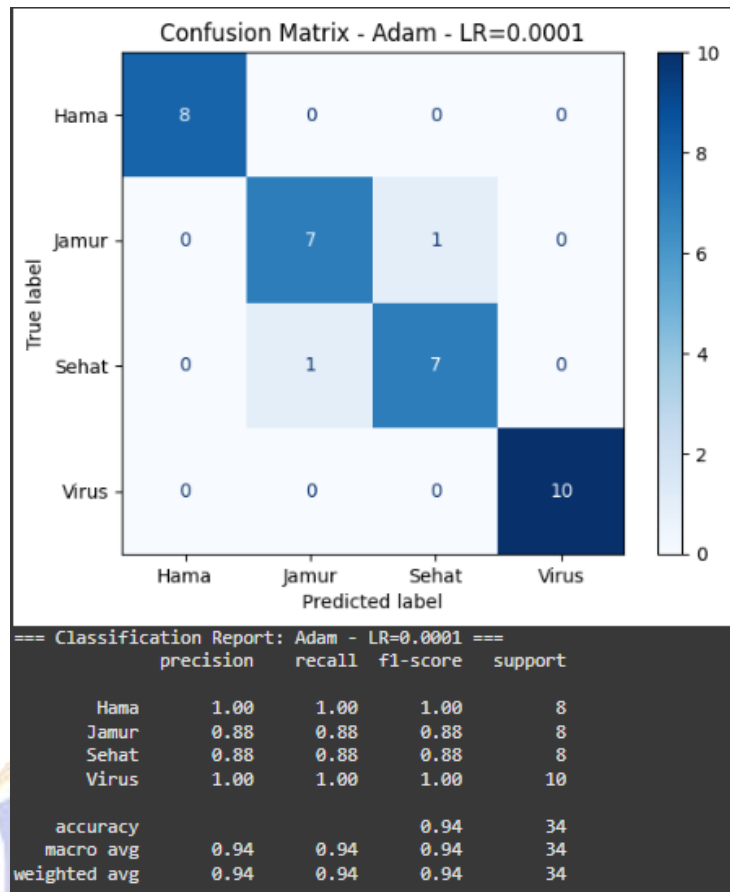
```
print("\n=== Hasil Evaluasi ===")
print("Akurasi:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred, target_names=['Hama', 'Jamur', 'Sehat', 'Virus']))
print("Confusion matrix:\n", confusion_matrix(y_test, y_pred))

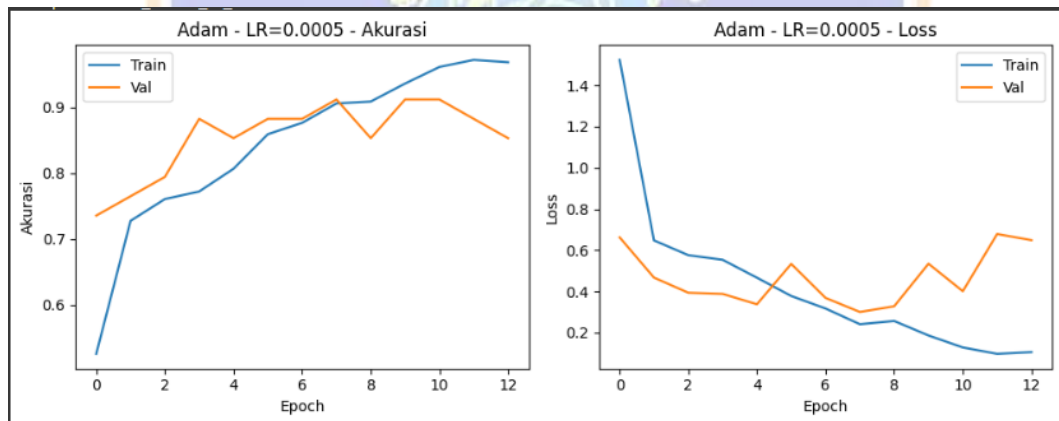
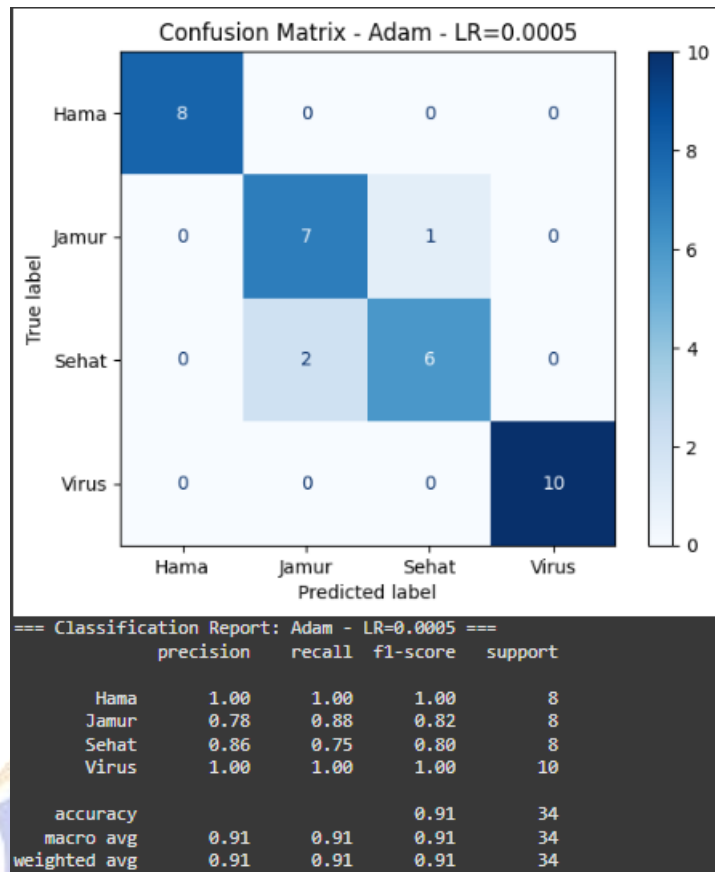
# =====
# 7. Simpan Model dan Scaler
# =====
joblib.dump(svm_model, "svm_model_poly.pkl")
joblib.dump(scaler, "scaler_poly.pkl")
print("\nModel dan scaler berhasil disimpan.")
```

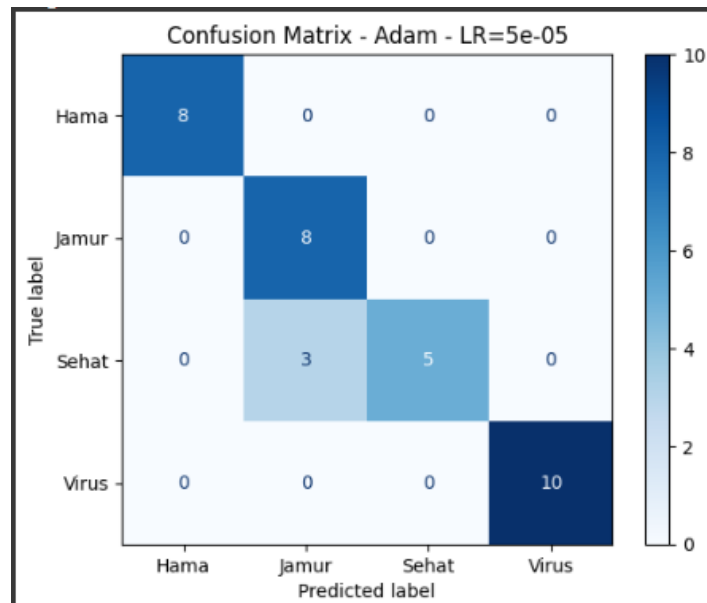


Lampiran 6. Hasil uji coba pengembangan model CNN pada Learning Rate



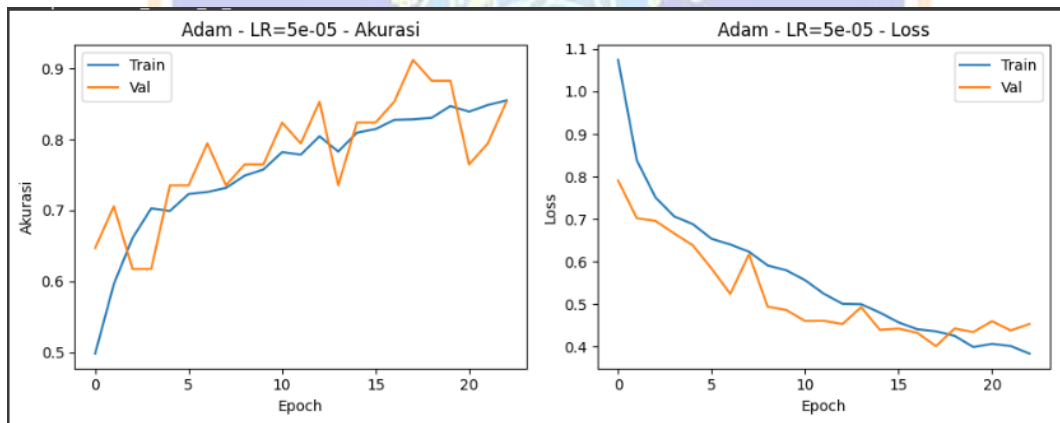






=== Classification Report: Adam - LR=5e-05 ===

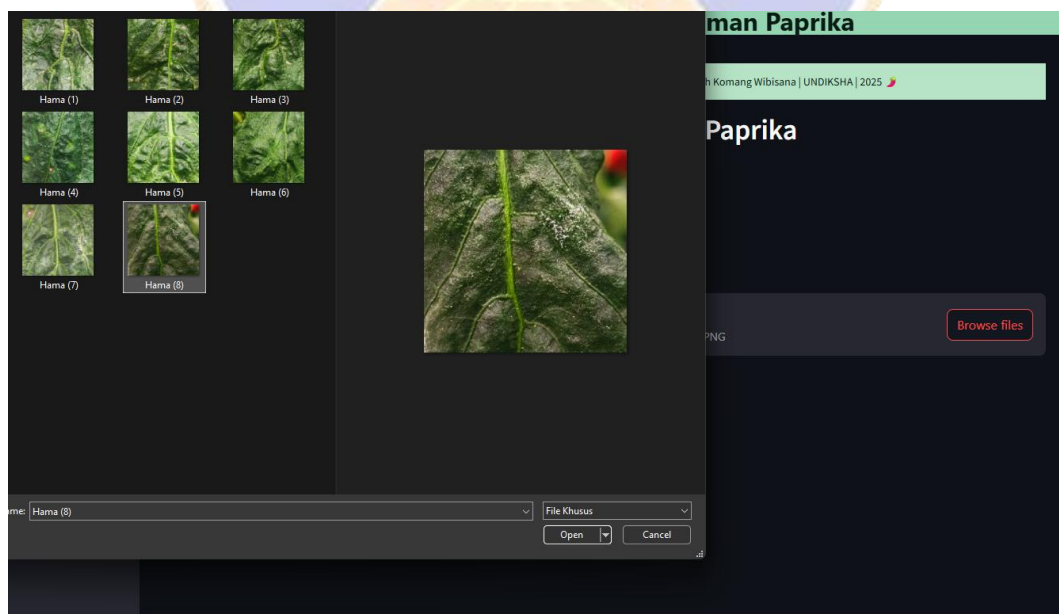
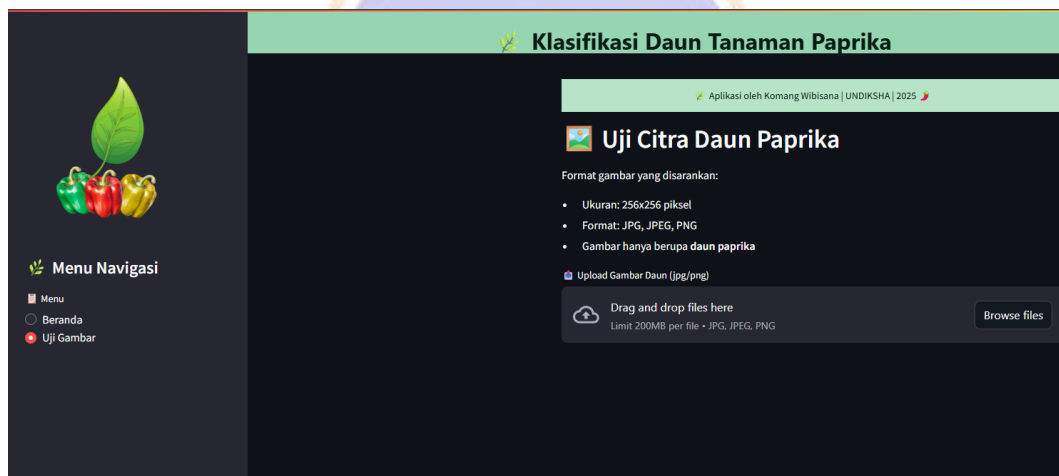
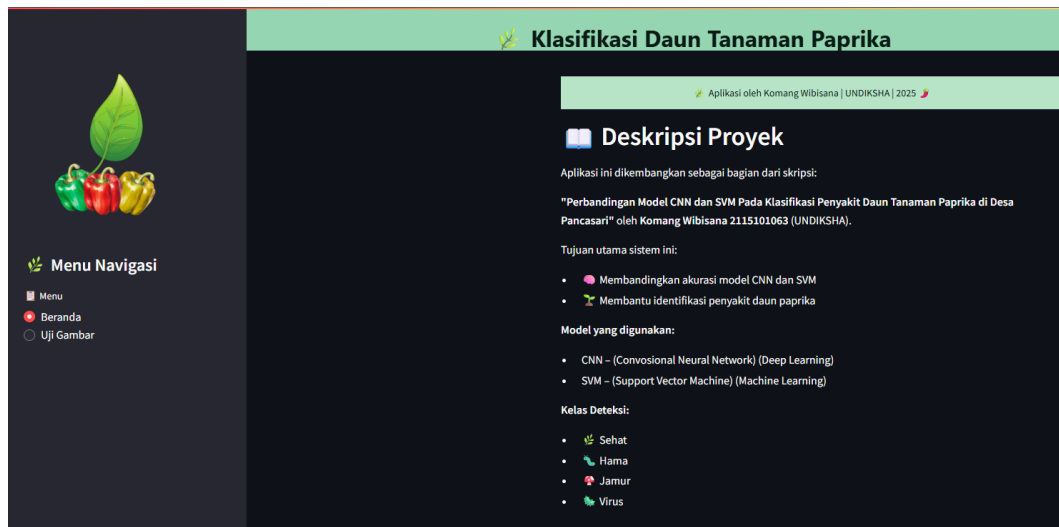
	precision	recall	f1-score	support
Hama	1.00	1.00	1.00	8
Jamur	0.73	1.00	0.84	8
Sehat	1.00	0.62	0.77	8
Virus	1.00	1.00	1.00	10
accuracy			0.91	34
macro avg	0.93	0.91	0.90	34
weighted avg	0.94	0.91	0.91	34



Lampiran 7. Hasil uji coba pengembangan model SVM pada parameter kernel.

No	C	Degree	Gamma	Kernel	Akurasi Training (%)
1	100	2	Scale	Polinomial	94.00
2	100	2	0.1	Polinomial	93.70
3	10	2	0.1	Polinomial	93.20
4	100	2	0.01	Polinomial	92.90
5	10	2	Scale	Polinomial	92.60
6	1	2	0.1	Polinomial	92.00
7	100	2	Auto	Polinomial	91.80
8	0.1	2	0.1	Polinomial	90.50
9	10	2	Auto	Polinomial	89.80
10	100	–	Scale	RBF	88.00
11	100	–	0.1	RBF	86.90
12	100	–	0.01	RBF	86.50
13	10	–	Auto	RBF	85.20
14	10	–	0.01	RBF	84.70
15	10	–	0.001	RBF	83.00
16	1	–	0.01	RBF	82.30
17	100	–	Scale	Linear	88.00
18	100	–	0.1	Linear	87.50
19	10	–	–	Linear	87.20
20	10	–	Auto	Linear	86.10
21	1	–	–	Linear	85.40
22	10	–	0.01	Linear	85.00
23	0.1	–	–	Linear	82.80

Lampiran 8. Pengujian Model CNN dan SVM pada website *Streamlit*





Menu Navigasi

- Menu
- Beranda
- Uji Gambar

Klasifikasi Daun Tanaman Paprika

- Format: JPG, JPEG, PNG
- Gambar hanya berupa daun paprika

Upload Gambar Daun (jpg/png)

Drag and drop files here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Hama (8).jpg 37.1KB



Gambar: Hama (8).jpg

CNN: Hama - Terserang serangga pengganggu.

SVM: Hama - Terserang serangga pengganggu.

Ringkasan Prediksi

	Nama File	Prediksi CNN	Prediksi SVM
0	Hama (8).jpg	Hama	Hama



Menu Navigasi

- Menu
- Beranda
- Uji Gambar

Klasifikasi Daun Tanaman Paprika

Upload Gambar Daun (jpg/png)

Drag and drop files here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Sehat (8).jpg 31.7KB



Gambar: Sehat (8).jpg

CNN: Sehat - Daun sehat tanpa gejala.

SVM: Hama - Terserang serangga pengganggu.

Prediksi CNN dan SVM tidak sama. Pertimbangkan evaluasi lanjutan.

Ringkasan Prediksi

	Nama File	Prediksi CNN	Prediksi SVM
0	Sehat (8).jpg	Sehat	Hama

RIWAYAT HIDUP



Komang Wibisana lahir di Desa Bengkel pada tanggal 29 November 2003. Penulis lahir dari pasangan suami istri Bapak I Nyoman Sudarsana dan Ibu Nengah Buntrini. Penulis berkebangsaan indonesia dan beragama Hindu. Kini peneliti tinggal di Taman Wira Segara Gang V, No. 2, Singaraja, Bali.

Peneliti menyelesaikan pendidikan dasar di SD Negeri 3 Bengkel dan lulus pada tahun 2015. Kemudian, peneliti melanjutkan pendidikan di SMP Negeri 1 Busungbiu dan lulus pada tahun 2018. Pada tahun 2021, peneliti lulus dari SMA Negeri 1 Banjar dengan mengambil jurusan Matematika dan Ilmu Pengetahuan Alam (MIPA). Setelah itu, peneliti terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer di Universitas Pendidikan Ganesha sejak tahun 2021 sampai dengan penulisan skripsi ini.

