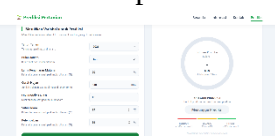
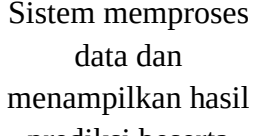


Lampiran 1 *BlackBox Testing*

A. Pengujian User

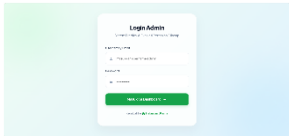
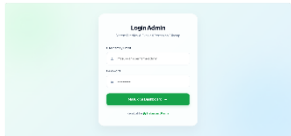
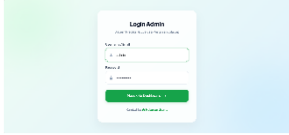
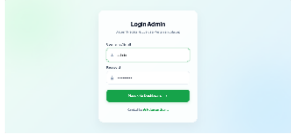
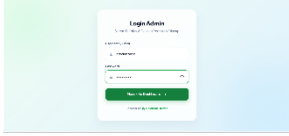
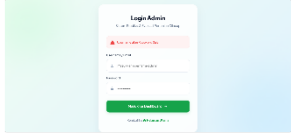
Tabel 13 Hasil Pengujian User

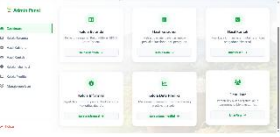
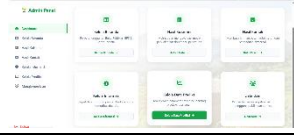
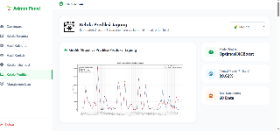
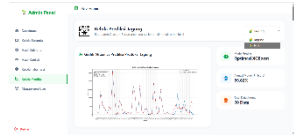
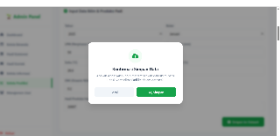
Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
(1)	(2)	(3)	(4)
Akses beranda	Mengakses halaman website 	Menampilkan halaman beranda 	Berhasil
Akses menu prediksi	Klik menu “Prediksi” 	Menampilkan pilihan komoditas prediksi 	Berhasil
Akses prediksi padi	Klik komoditas “Padi” 	Menampilkan halaman prediksi padi 	Berhasil
Input variabel prediksi	Menginputkan variabel prediksi 	Data variabel prediksi dapat diinput 	Berhasil
Validasi input tidak lengkap	Klik tombol “Hitung Prediksi” tanpa melengkapi data 	Sistem menampilkan pesan error 	Berhasil
Proses prediksi	Klik tombol “Hitung Prediksi” dengan data lengkap 	Sistem memproses data dan menampilkan hasil prediksi beserta kategorinya 	Berhasil

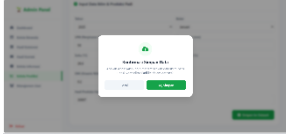
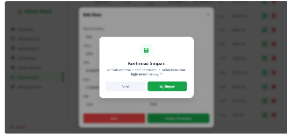
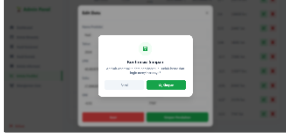
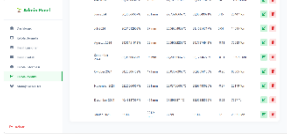
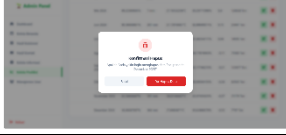
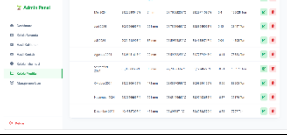
Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
(1)	(2)	(3)	(4)
			
Tampilan rekomendasi	Setelah prediksi berhasil	Sistem menampilkan rekomendasi sesuai kategorinya 	Berhasil

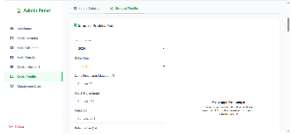
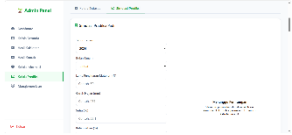
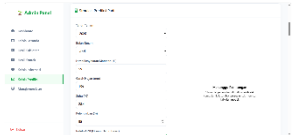
B. Pengujian Admin

Tabel 14 Hasil Pengujian Admin

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
(1)	(2)	(3)	(4)
Login Admin	Masuk ke halaman login admin 	Menampilkan halaman login admin 	Berhasil
Input <i>username</i> dan <i>password</i>	Mengisi kolom <i>username</i> dan <i>password</i> 	<i>Username</i> dan <i>password</i> dapat diinput 	Berhasil
Validasi salah inputan	Klik tombol “Masuk ke <i>Dashbord</i> ” dengan data yang salah 	Sistem menampilkan pesan eror 	Berhasil
Akses <i>dashboard</i> admin	Setelah login berhasil	Menampilkan <i>dashboard</i> admin	Berhasil

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
(1)	(2)	(3)	(4)
			
Akses kelola prediksi	Klik menu “Kelola Prediksi” 	Menampilkan halaman kelola prediksi 	Berhasil
Akses kelola prediksi padi	Klik pilihan kelola prediksi padi 	Menampilkan halaman kelola prediksi padi 	Berhasil
Akses kelola dataset	Klik pilihan “Kelola Dataset” 	Menampilkan halaman kelola dataset 	Berhasil
Input data baru	Mengisi kolom tahun, bulan, prediksi, variabel iklim, dan produksi padi 	Data tahun, bulan, prediksi, variabel iklim, dan produksi padi dapat diinputkan 	Berhasil
Konfirmasi simpan data baru	Klik button “Simpan ke Dataset” 	Sistem menampilkan konfirmasi 	Berhasil
Simpan dataset baru	Klik button “Ya Simpan”	Data tersimpan di database dan riwayat dataset terdaftar	Berhasil

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
(1)	(2)	(3)	(4)
			
Akses Edit data	Pilih salah satu data lalu klik tombol edit 	Menampilkan data yang akan diedit 	Berhasil
Melakukan edit data	Mengganti data yang ada dengan data baru 	Data dapat diubah ke data baru 	Berhasil
Konfirmasi edit data	Klik button “Simpan Perubahan” 	Sistem menampilkan konfirmasi 	Berhasil
Simpan edit data	Klik button “Ya Simpan” 	Data tersimpan di database dan riwayat dataset terdaftar 	Berhasil
Akses hapus data	Pilih salah satu data lalu klik tombol hapus 	Sistem menampilkan konfirmasi 	Berhasil
Konfirmasi edit data	Klik “Ya Hapus Data” 	Data terhapus dari database dan riwayat dataset terdaftar 	Berhasil

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
(1)	(2)	(3)	(4)
Akses simulasi prediksi	Klik menu “Simulasi Prediksi” 	Menampilkan halaman simulasi prediksi 	Berhasil
Input variabel prediksi	Mengisi variabel prediksi 	Data variabel prediksi dapat diinput 	Berhasil
Validasi input tidak lengkap	Klik tombol “Hitung Prediksi” tanpa melengkapi data 	Sistem menampilkan pesan eror 	Berhasil
Proses prediksi	Klik tombol “Hitung Prediksi” dengan data lengkap 	Sistem memproses data dan menampilkan hasil prediksi 	Berhasil
Tampilan hasil, kategori, dan rekomendasi	Setelah prediksi berhasil	Sistem menampilkan estimasi bulan panen, prediksi panen dalam ton, kategori prediksi, dan rekomendasi 	Berhasil

Lampiran 2 Source Code Komparasi Algoritma

A. Algoritma XGBoost

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os

from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_absolute_error,
mean_squared_error

# =====
# 1. LOAD DATA
# =====
df = pd.read_excel("data_padi.xlsx")

# =====
# 2. PROSES LAGGING (SEMUA VARIABEL IKLIM)
# =====
kolom_fitur = ['LPM (%)', 'CH (mm)', 'ONI', 'Suhu (°C)', 'Kelembaban (%)']

for col in kolom_fitur:
    df[f'{col}_Lag1'] = df[col].shift(1)
    df[f'{col}_Lag2'] = df[col].shift(2)
    df[f'{col}_Lag3'] = df[col].shift(3)

df_clean = df.dropna().reset_index(drop=True)

# =====
# 3. FITUR (X) & TARGET (y)
# =====
features = [col for col in df_clean.columns if 'Lag' in col]
X = df_clean[features]
y = df_clean['P (ton)']

# =====
# 4. SPLIT DATA (RANDOM APPROACH)
# =====
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42
)

# =====
# 5. MODEL XGBOOST
```

```

# =====
xgb = XGBRegressor(
    n_estimators=100, #Jumlah pohon keputusan
    learning_rate=0.5, #Kecepatan belajar
    max_depth=5, #kedalaman maksimum pohon
    subsample=1, #Data yg digunakan untuk membangun setiap pohon
    colsample_bytree=1, #Fitur untuk membangun setiap pohon
    random_state=42 #Agar hasil prediksi konsisten setiap dijalankan
)

xgb.fit(X_train, y_train)

# =====
# 6. PREDIKSI SELURUH DATA (57 BARIS)
# =====
y_all_pred = xgb.predict(X)

# =====
# 7. EVALUASI (SELURUH DATA)
# =====
r2_all = r2_score(y, y_all_pred)
mae_all = mean_absolute_error(y, y_all_pred)
rmse_all = np.sqrt(mean_squared_error(y, y_all_pred))

# =====
# 8. VISUALISASI GRAFIK 57 BULAN
# =====
date_range = pd.date_range(start='2020-04-01', periods=len(y),
                             freq='MS')
date_labels = [d.strftime('%b-%y') for d in date_range]

plt.figure(figsize=(16, 8))

plt.plot(date_labels, y.values,
         label='Data Aktual (Asli)',
         color='#1f77b4',
         marker='o',
         linewidth=2)

plt.plot(date_labels, y_all_pred,
         label='Prediksi XGBoost',
         color='#d62728',
         linestyle='--',
         marker='s',
         alpha=0.8)

info_box = (f"Statistik Model (57 Bulan):\n"
            f"R2 Score: {round(r2_all, 3)}\n"
            f"MAE: {round(mae_all, 2)} Ton\n"
            f"RMSE: {round(rmse_all, 2)} Ton")

```

```

plt.gca().text(0.02, 0.95,
               info_box,
               transform=plt.gca().transAxes,
               fontsize=12,
               verticalalignment='top',
               bbox=dict(boxstyle='round', facecolor='white',
alpha=0.9))

plt.title('Grafik Prediksi Produksi Padi (Random Split 80:20 -
XGBoost)',
          fontsize=16, pad=20)

plt.xlabel('Periode Waktu (Bulan-Tahun)', fontsize=12)
plt.ylabel('Produksi Padi (Ton)', fontsize=12)
plt.xticks(rotation=45, fontsize=9)
plt.legend(loc='upper right', fontsize=11)
plt.grid(True, linestyle=':', alpha=0.6)

plt.tight_layout()

# Pastikan folder grafik tersedia sebelum menyimpan file gambar
if not os.path.exists("grafik"):
    os.makedirs("grafik")

# Menyimpan grafik aktual vs prediksi XGBoost
plt.savefig("grafik/xg_random80.png", dpi=300, bbox_inches='tight')
print("✓ Grafik dengan pendekatan Random Split 80:20 berhasil
disimpan di folder /grafik!")

plt.show()
plt.close()

# =====
# 9. OUTPUT HASIL DI TERMINAL
# =====
print("\n" + "="*40)
print(f"HASIL AKHIR R²: {r2_all:.3f}")
print(f"HASIL AKHIR MAE: {mae_all:.2f} Ton")
print(f"HASIL AKHIR RMSE: {rmse_all:.2f} Ton")
print("="*40)

# =====
# 10. FEATURE IMPORTANCE (HANYA DITAMPILKAN DI TERMINAL)
# =====
importances = pd.Series(xgb.feature_importances_, index=features)
importances = importances.sort_values(ascending=False)

# Mengubah ke skala persentase murni (dikali 100)

```

```

top5_xgb = importances.head(5) * 100

print("\nTOP 5 VARIABEL PALING BERPENGARUH (XGBOOST):")
for rank, (nama_fitur, nilai) in enumerate(top5_xgb.items(), 1):
    print(f"{rank}. {nama_fitur}: {nilai:.2f}%")
print("="*40)

```

B. Algoritma ANN

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os

from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_absolute_error,
mean_squared_error

# =====
# 1. LOAD DATA
# =====
df = pd.read_excel("data_padi.xlsx")

# =====
# 2. PROSES LAGGING (SEMUA VARIABEL IKLIM)
# =====
kolom_fitur = ['LPM (%)', 'CH (mm)', 'ONI', 'Suhu (°C)', 'Kelembaban (%)']

for col in kolom_fitur:
    df[f'{col}_Lag1'] = df[col].shift(1) # 1 bulan sebelum panen
    df[f'{col}_Lag2'] = df[col].shift(2) # 2 bulan sebelum panen
    df[f'{col}_Lag3'] = df[col].shift(3) # 3 bulan sebelum panen

# Hapus data NaN akibat lagging (Menghasilkan 57 baris bersih)
df_clean = df.dropna().reset_index(drop=True)

# =====
# 3. FITUR (X) & TARGET (y)
# =====
features = [col for col in df_clean.columns if 'Lag' in col]

X = df_clean[features]
y = df_clean['P (ton)']

# =====
# 4. SPLIT DATA (RANDOM APPROACH 80:20)

```

```

# =====
X_train_raw, X_test_raw, y_train_raw, y_test_raw = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42
)

# =====
# 5. SCALING / NORMALISASI
# =====
scaler_X = StandardScaler()
scaler_y = StandardScaler()

# fit_transform HANYA diaplikasikan pada data latih (Train)
X_train = scaler_X.fit_transform(X_train_raw)
y_train = scaler_y.fit_transform(y_train_raw.values.reshape(-1,1)).ravel()

# Data uji (Test) hanya di-transform menggunakan parameter skala
dari data latih
X_test = scaler_X.transform(X_test_raw)
y_test = scaler_y.transform(y_test_raw.values.reshape(-1,1)).ravel()

# Mempersiapkan skala seluruh data untuk prediksi total di grafik
X_scaled = scaler_X.transform(X)

# =====
# 6. MODEL ANN (Baseline Tanpa Optimasi)
# =====
ann = MLPRegressor(
    hidden_layer_sizes=(100),
    alpha=0.0005,
    learning_rate_init=0.01,
    max_iter=100, #Jumlah maksimum iterasi training
    random_state=42
)

# Proses pelatihan 80% data latih yang sudah discaling bersih
ann.fit(X_train, y_train)

# =====
# 7. PREDIKSI SELURUH DATA (57 BARIS)
# =====
y_all_pred_scaled = ann.predict(X_scaled)

# Denormalisasi: Mengembalikan angka skala ke satuan Ton asli
y_all_pred = scaler_y.inverse_transform(
    y_all_pred_scaled.reshape(-1,1)
).ravel()

```

```

# =====
# 8. EVALUASI AKURASI TOTAL
# =====
r2_all = r2_score(y, y_all_pred)
mae_all = mean_absolute_error(y, y_all_pred)
rmse_all = np.sqrt(mean_squared_error(y, y_all_pred))

# =====
# 9. VISUALISASI GRAFIK 57 BULAN
# =====
date_range = pd.date_range(start='2020-04-01', periods=len(y),
freq='MS')
date_labels = [d.strftime('%b-%y') for d in date_range]

plt.figure(figsize=(16, 8))

# Plot Garis Aktual Asli
plt.plot(date_labels, y.values, label='Data Aktual (Asli)',
color='#1f77b4', marker='o', linewidth=2)

# Plot Garis Hasil Prediksi ANN
plt.plot(date_labels, y_all_pred, label='Prediksi ANN',
color='#d62728', linestyle='--', marker='s', alpha=0.8)

# Informasi Statistik Performa Model
info_box = (f"Statistik Model (57 Bulan):\n"
            f"R2 Score: {round(r2_all, 3)}\n"
            f"MAE: {round(mae_all, 2)} Ton\n"
            f"RMSE: {round(rmse_all, 2)} Ton")

plt.gca().text(0.02, 0.95, info_box, transform=plt.gca().transAxes,
fontSize=12,
               verticalalignment='top', bbox=dict(boxstyle='round',
facecolor='white', alpha=0.9))

plt.title('Grafik Prediksi Produksi Padi (Random Split 80:20 -
ANN)', fontsize=16, pad=20)
plt.xlabel('Periode Waktu (Bulan-Tahun)', fontsize=12)
plt.ylabel('Produksi Padi (Ton)', fontsize=12)
plt.xticks(rotation=45, fontsize=9)
plt.legend(loc='upper right', fontsize=11)
plt.grid(True, linestyle=':', alpha=0.6)
plt.tight_layout()

# --- SIMPAN GRAFIK PREDIKSI ---
os.makedirs("grafik", exist_ok=True)
plt.savefig("grafik/ann_random80.png", dpi=300, bbox_inches='tight')

```

```

print("✓ Grafik ANN dengan pendekatan Random Split (80:20) berhasil
disimpan!")

# Menampilkan grafik prediksi aktual ke layar
plt.show()
plt.close('all')

# =====
# 10. OUTPUT HASIL AKHIR DI TERMINAL
# =====
print("\n" + "="*40)
print(f"HASIL AKHIR R²: {r2_all:.3f}")
print(f"HASIL AKHIR MAE: {mae_all:.2f} Ton")
print(f"HASIL AKHIR RMSE: {rmse_all:.2f} Ton")
print("="*40)

# =====
# 11. WEIGHT ANALYSIS IMPORTANCE (MURNI TERMINAL)
# =====
bobot_input_hidden = np.abs(ann.coefs_[0])
kontribusi_fitur = np.mean(bobot_input_hidden, axis=1)
skor_importance = (kontribusi_fitur / np.sum(kontribusi_fitur)) *
100

importances = pd.Series(skor_importance,
index=features).sort_values(ascending=False)

print("\nWEIGHT ANALYSIS IMPORTANCE (TOP 5 VARIABEL PALING
BERPENGARUH):")
for rank, (nama_fitur, nilai) in
enumerate(importances.head(5).items(), 1):
    print(f"{rank}. {nama_fitur}: {nilai:.2f}%")
print("="*40)

```