

**PRAKTIKUM
SISTEM CERDAS DAN PENDUKUNG KEPUTUSAN
SEMESTER GENAP T.A 2024/2025
LAPORAN PROYEK AKHIR**



DISUSUN OLEH :

**NIM : 123230005
123230006**
**NAMA : MELANIA INTAN SAGITA
FADILAH NUR SABIYYAH**
PLUG : IF-G
**NAMA ASISTEN : ARVIDION HAVAS OKTAVIAN
PANCA AULIA RAHMAN**

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
YOGYAKARTA
2025**

HALAMAN PENGESAHAN LAPORAN PROYEK AKHIR

Disusun oleh :

Melania Intan Sagita
Fadilah Nur Sabiyyah

123230005
123230006

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Sistem Cerdas dan
Pendukung Keputusan

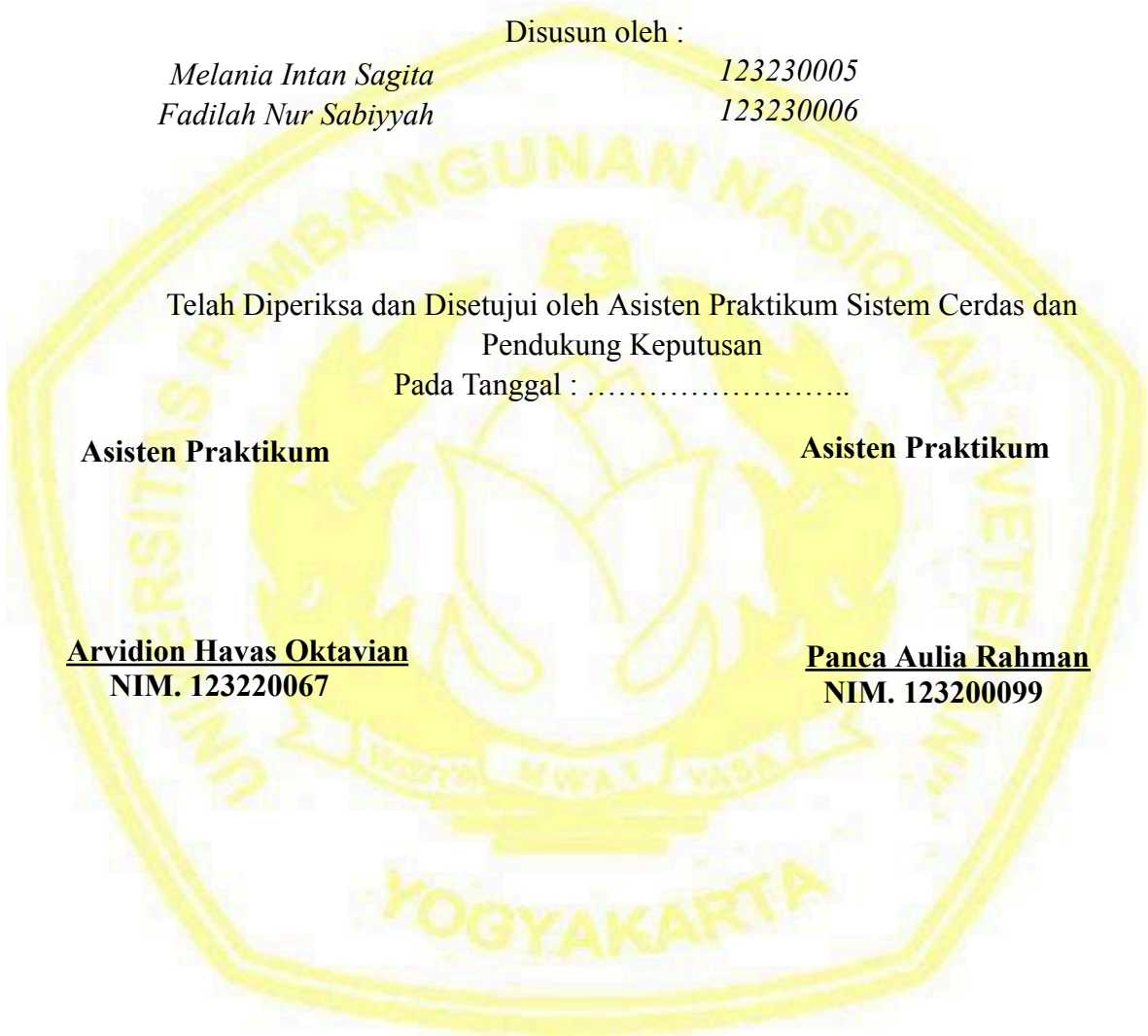
Pada Tanggal :

Asisten Praktikum

Asisten Praktikum

Arvidion Havas Oktavian
NIM. 123220067

Panca Aulia Rahman
NIM. 123200099



KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan praktikum Sistem Cerdas dan Pendukung Keputusan serta laporan proyek akhir praktikum yang berjudul Pemilihan Hasil Panen. Adapun laporan ini berisi tentang proyek akhir yang saya pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten dosen yang selalu membimbing dan mengajari saya dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun saya harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, saya ucapkan terimakasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 02 Juni 2025

Penyusun I

Penyusun II

Melania Intan Sagita

NIM. 123230006

Fadilah Nur Sabiyyah

NIM. 123230006

DAFTAR ISI

HALAMAN PENGESAHAN.....	2
LAPORAN PROYEK AKHIR.....	2
KATA PENGANTAR.....	3
DAFTAR ISI.....	4
BAB I.....	6
PENDAHULUAN.....	6
1.1 Latar Belakang Masalah.....	6
1.2 Tujuan Proyek Akhir.....	6
1.3 Manfaat Proyek Akhir.....	6
BAB II.....	7
PEMBAHASAN.....	7
2.1 Dasar Teori.....	7
2.2 Deskripsi Umum Proyek Akhir.....	7
2.3 Inti Pembahasan.....	7
BAB III.....	22
JADWAL Pengerjaan dan Pembagian Tugas.....	22
3.1 Jadwal Pengerjaan.....	22
3.2 Pembagian Tugas.....	22
BAB IV.....	23
KESIMPULAN DAN SARAN.....	23
4.1 Kesimpulan.....	23
4.2 Saran.....	23
DAFTAR PUSTAKA.....	24

Pemilihan Hasil Panen

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pemilihan hasil panen terbaik merupakan langkah penting dalam menunjang keberhasilan sektor pertanian, terutama untuk memastikan kualitas, kuantitas, dan keberlanjutan produksi pertanian. Banyaknya jenis tanaman, variasi lokasi lahan, perbedaan teknik budidaya, serta faktor-faktor lain seperti curah hujan, kelembaban, dan jenis pupuk, membuat proses penentuan hasil panen terbaik menjadi kompleks. Tanpa bantuan analisis yang tepat, petani atau pihak terkait dapat kesulitan dalam menentukan pilihan terbaik dari berbagai alternatif yang ada. Oleh karena itu, diperlukan suatu sistem cerdas yang mampu membantu dalam proses pengambilan keputusan berdasarkan berbagai kriteria yang mempengaruhi hasil panen. Sistem ini akan memanfaatkan metode pengambilan keputusan seperti SAW (*Simple Additive Weighting*) untuk menilai dan merekomendasikan hasil panen terbaik berdasarkan data yang tersedia.

1.2 Tujuan Proyek Akhir

Tujuan dari proyek ini adalah untuk mengembangkan suatu sistem cerdas yang mampu merekomendasikan hasil panen terbaik berdasarkan data hasil pertanian. Sistem ini akan memanfaatkan metode *Simple Additive Weighting* (SAW) untuk melakukan penilaian terhadap berbagai alternatif hasil panen berdasarkan sejumlah kriteria, seperti jenis tanaman, jumlah produksi, luas lahan, curah hujan, kelembaban, dan penggunaan pupuk. Dengan metode ini, setiap alternatif akan diberi skor berdasarkan pembobotan kriteria yang telah ditentukan, sehingga sistem dapat memberikan rekomendasi hasil panen yang paling optimal secara objektif dan terukur.

1.3 Manfaat Proyek Akhir

1. Memudahkan Pengambilan Keputusan: Sistem ini akan membantu petani atau pihak terkait dalam memilih hasil panen terbaik tanpa harus melakukan perhitungan manual atau pendekatan coba-coba, sehingga dapat menghemat waktu dan tenaga dalam proses evaluasi hasil pertanian.
2. Meningkatkan Efisiensi Pertanian: Dengan pemilihan hasil panen berdasarkan data dan analisis yang objektif, sistem ini dapat meningkatkan kesadaran akan pentingnya pengelolaan pertanian berbasis data, sehingga mendorong praktik pertanian yang lebih efisien dan tepat sasaran.
3. Penggunaan Sumber Daya Secara Optimal: Dengan rekomendasi hasil panen yang tepat, pengguna dapat memastikan bahwa mereka memanfaatkan lahan, pupuk, dan air secara optimal untuk memperoleh hasil panen yang maksimal, sehingga mendukung produktivitas dan keberlanjutan sektor pertanian.

BAB II PEMBAHASAN

2.1 Dasar Teori

Sistem Pendukung Keputusan adalah sebuah sistem informasi yang menyediakan data, proses manipulasi, serta pemodelan untuk mendukung pengambilan keputusan yang lebih fleksibel. Dengan memanfaatkan sistem ini, proses pembuatan keputusan dapat menjadi lebih efektif dan terarah [3].

Metode *Simple Additive Weighting* (SAW) merupakan teknik dalam sistem pendukung keputusan yang mengutamakan penjumlahan terbobot atau pembobotan sederhana. Metode ini bertujuan untuk menentukan peringkat kinerja atau skala prioritas dari setiap alternatif berdasarkan seluruh atribut yang ada [2].

2.2 Deskripsi Umum Proyek Akhir

Proyek akhir ini mengembangkan aplikasi berbasis web menggunakan Streamlit yang bertujuan untuk membantu pengguna dalam menentukan hasil panen terbaik berdasarkan berbagai faktor dari dataset yang mencakup informasi seperti Jenis tanaman (*Crop*), Wilayah (*Region*), Jenis tanah (*Soil Type*), Kondisi cuaca (*Weather_Condition*), Curah hujan (*Rainfall_mm*), Suhu (*Temperature_Celsius*), Penggunaan pupuk (*Fertilizer_Used*), Penggunaan irigasi (*Irrigation_Used*), Lama panen (*Days_to_Harvest*), dan Hasil panen (*Yield_tons_per_hectare*).

Projek ini menyediakan fitur Visualisasi data, Filter data berdasarkan kriteria, Penyesuaian bobot kriteria dan tipe, dan Hasil akhir berupa skor dan peringkat SAW. Dengan kombinasi data aktual dan metode SAW, pengguna dapat memperoleh rekomendasi tanaman terbaik untuk ditanam berdasarkan kondisi yang ada.

2.3 Inti Pembahasan

2.3.1 Pengumpulan Data

Data yang digunakan dalam proyek ini berasal dari file *crop_yield.csv* yang diperoleh dari situs web *Kaggle*. Dataset ini berisi informasi mengenai hasil panen berbagai jenis tanaman berdasarkan sejumlah faktor agrikultur dan cuaca. Data mencakup atribut seperti jenis tanaman (*Crop*), wilayah (*Region*), jenis tanah (*Soil Type*), kondisi cuaca (*Weather_Condition*), curah hujan (*Rainfall_mm*), suhu (*Temperature_Celsius*), penggunaan pupuk (*Fertilizer_Used*), penggunaan irigasi (*Irrigation_Used*), lama waktu panen (*Days_to_Harvest*), dan hasil panen per hektare (*Yield_tons_per_hectare*). Dataset disimpan dalam format CSV dan diolah menggunakan pustaka Python seperti *Pandas* dan *scikit-learn*. Untuk keperluan analisis menggunakan metode SAW, penulis memilih 6 kriteria utama, yaitu *Rainfall_mm*, *Temperature_Celsius*, *Fertilizer_Used*, *Irrigation_Used*, *Days_to_Harvest*, dan *Yield_tons_per_hectare*.

2.3.2 Preprocessing Data

Preprocessing dilakukan untuk membersihkan dan menyiapkan data sebelum

diterapkan metode SAW. Langkah-langkah yang dilakukan meliputi penghapusan nilai kosong, konversi data kategorikal (seperti *Crop*, *Region*, dan *Soil_Type*) menjadi numerik dengan *LabelEncoder*, serta mengubah nilai *boolean* (*Fertilizer_Used* dan *Irrigation_Used*) ke format angka (0 dan 1). Proses ini dilakukan menggunakan Python dengan pustaka *pandas* dan *scikit-learn* di VSCode.

2.3.3 Implementasi Algoritma SAW

Pada proyek ini, metode *Simple Additive Weighting* (SAW) diimplementasikan menggunakan bahasa Python dan *framework* Streamlit. SAW merupakan salah satu metode yang umum digunakan dalam Sistem Pendukung Keputusan karena kemudahannya dalam mengolah data numerik berdasarkan bobot dan kriteria yang telah ditentukan.

Berikut adalah langkah-langkah implementasi SAW:

a. Sample (Alternatif):

Setiap baris data dalam dataset *crop_yield.csv* mewakili satu alternatif hasil panen (misalnya, panen tanaman tertentu di wilayah tertentu), yang akan dievaluasi berdasarkan beberapa kriteria.

b. Criteria & Data Matrix:

Matriks data diambil dari atribut-atribut numerik pada dataset, seperti:

- Rainfall_mm
- Temperature_Celsius
- Fertilizer_Used
- Irrigation_Used
- Days_to_Harvest
- Yield_tons_per_hectare

Setiap kolom adalah kriteria, dan setiap baris adalah alternatif.

c. Normalisasi:

Untuk menyamakan skala antar kriteria, dilakukan normalisasi sebagai berikut:

- Untuk kriteria benefit:

$$r_{ij} = \frac{x_{ij}}{x_j^{max}}$$

- Untuk kriteria cost:

$$r_{ij} = \frac{x_j^{min}}{x_{ij}}$$

d. Pembobotan dan Skoring:

Setiap nilai normalisasi dikalikan dengan bobot kriteria (yang ditentukan oleh pengguna). Skor total untuk masing-masing alternatif dihitung dengan rumus:

$$V_i = \sum_{j=1}^n w_j \cdot r_{ij}$$

Di mana:

- V_i adalah skor total alternatif ke-i
- w_j adalah bobot dari kriteria ke-j

- r_{ij} adalah nilai normalisasi alternatif ke-i terhadap kriteria ke-j
- e. Peringkat:
Setelah skor total dihitung untuk seluruh alternatif, dilakukan pengurutan berdasarkan skor tertinggi untuk menentukan hasil panen terbaik.

2.3.4 Source Code Program

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, MinMaxScaler

# 1. Load data
df = pd.read_csv("Projek/crop_yield.csv")

# 2. Cek dan tangani nilai kosong (jika ada)
if df.isnull().sum().sum() > 0:
    df = df.dropna() # atau bisa juga df.fillna(...)

# 3. Encoding fitur kategorikal
label_cols = ["Region", "Soil_Type", "Crop",
               "Weather_Condition"]
le_dict = {} # Untuk menyimpan encoder jika ingin decode
nanti

for col in label_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    le_dict[col] = le

# 4. Normalisasi fitur numerik
scaler = MinMaxScaler()
numerical_cols = ["Rainfall_mm", "Temperature_Celsius",
                  "Days_to_Harvest", "Yield_tons_per_hectare"]
df[numerical_cols] =
scaler.fit_transform(df[numerical_cols])

# 5. Pastikan fitur boolean dikonversi jadi angka
df["Fertilizer_Used"] = df["Fertilizer_Used"].astype(int)
df["Irrigation_Used"] = df["Irrigation_Used"].astype(int)

# 6. Simpan data siap pakai
df.to_csv("Projek/crop_yield_preprocessed.csv", index=False)

# Preview
print("✅ Data selesai dipreprocessing. Dimensi:", df.shape)
print(df.head())
```

processing.py

```
import streamlit as st
```

```

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder

# Konfigurasi halaman
st.set_page_config(page_title="Hasil Panen Tanaman",
page_icon="🌾")
st.title('🌱 Aplikasi SAW Hasil Panen Tanaman 🌱')

# Informasi anggota
col1, col2 = st.columns(2)
with col1:
    st.write("Melania Intan Sagita")
    st.write("Fadiyah Nur Sabiyah")
with col2:
    st.write("123230005")
    st.write("123230006")
st.header("Deskripsi")
st.write("Aplikasi ini menggunakan metode Simple Additive Weighting (SAW) "
"untuk membantu menentukan tanaman terbaik berdasarkan hasil panen dan beberapa "
"kriteria penting seperti curah hujan, suhu, pupuk, irigasi, waktu panen, dan hasil produksi.")

# Load Dataset
df_raw = pd.read_csv("Projek/crop_yield.csv")
df = pd.read_csv("Projek/crop_yield_preprocessed.csv")

# Recreate LabelEncoders
label_cols = ["Region", "Soil_Type", "Crop", "Weather_Condition"]
le_dict = {}
for col in label_cols:
    le = LabelEncoder()
    le.fit(df_raw[col])
    le_dict[col] = le

def decode_labels(df, label_cols, le_dict):
    for col in label_cols:
        if col in df.columns:
            df[col] = le_dict[col].inverse_transform(df[col])
    return df

# Kriteria dan Bobot
default_criteria = {
    'Rainfall_mm': {'is_benefit': True, 'weight': 0.15},
    'Temperature_Celsius': {'is_benefit': True, 'weight': 0.15},
    'Fertilizer_Used': {'is_benefit': True, 'weight': 0.10},
    'Irrigation_Used': {'is_benefit': True, 'weight': 0.10},

```

```

        'Days_to_Harvest': {'is_benefit': False, 'weight':
0.20},
        'Yield_tons_per_hectare': {'is_benefit': True, 'weight':
0.30},
    }

if 'criteria_config' not in st.session_state:
    st.session_state.criteria_config =
default_criteria.copy()

# Tabs
tab1, tab2, tab3, tab4, tab5 = st.tabs(["Dataset", "Grafik",
"Cari Data", "Input Data", "Hasil Akhir"])

# Tab 1: Dataset
with tab1:
    st.subheader("📁 Informasi Dataset")
    jmlh_raw = st.number_input('Jumlah data mentah
ditampilkan', 0, len(df_raw), 5, key='jmlh_raw')
    st.dataframe(df_raw.astype(str).head(jmlh_raw))
    st.write('📏 Dimensi:', df_raw.shape)
    st.write('❓ Nilai Kosong:', df_raw.isnull().sum())
    st.write("Statistik Deskriptif:")
    numeric_raw = df_raw.select_dtypes(include='number')
    if not numeric_raw.empty:
        stat_raw = numeric_raw.describe().T[['mean', '50%',
'std', 'min', 'max']]
        stat_raw.columns = ['Rata-rata', 'Median', 'Standar
Deviiasi', 'Minimum', 'Maksimum']
        st.dataframe(stat_raw)
    else:
        st.info("Tidak ada kolom numerik dalam data
mentah.")

    st.markdown("---")
    st.markdown("### ✅ Dataset Sudah Preprocessing")
    jmlh = st.number_input('Jumlah data preprocessed
ditampilkan', 0, len(df), 5, key='jmlh_prep')
    st.dataframe(df.head(jmlh))
    st.write('📏 Dimensi:', df.shape)
    st.write('❓ Nilai Kosong:', df.isnull().sum())
    st.write("Statistik Deskriptif:")
    numeric_prep = df.select_dtypes(include='number')
    if not numeric_prep.empty:
        stat_prep = numeric_prep.describe().T[['mean',
'50%', 'std', 'min', 'max']]
        stat_prep.columns = ['Rata-rata', 'Median', 'Standar
Deviiasi', 'Minimum', 'Maksimum']
        st.dataframe(stat_prep)
    else:
        st.info("Tidak ada kolom numerik dalam data yang
sudah diproses.")

```

```

with st.expander("📄 Lihat Mapping LabelEncoder"):
    for col in label_cols:
        st.write(f"Mapping kolom **{col}**:")
        mapping_df = pd.DataFrame({
            'Label': le_dict[col].classes_,
            'Encoded': range(len(le_dict[col].classes_))
        })
        st.dataframe(mapping_df)

# Tab 2: Grafik
with tab2:
    st.subheader("📊 Grafik Data")
    numeric_cols =
df.select_dtypes(include='number').columns.tolist()
    selected_col = st.selectbox("Pilih kolom numerik",
numeric_cols)
    if 'Crop' in df.columns:
        avg_by_crop =
df.groupby('Crop')[selected_col].mean().sort_values(ascending=False)
        fig, ax = plt.subplots(figsize=(10, 5))
        avg_by_crop.plot(kind='bar', ax=ax,
color='lightgreen', edgecolor='black')
        ax.set_title(f"Rata-rata {selected_col} per
Tanaman")
        st.pyplot(fig)
    else:
        st.warning("Kolom 'Crop' tidak ditemukan di data
preprocessed.")

# Tab 3: Cari Data
with tab3:
    st.subheader("🔍 Cari Data dari Dataset Mentah")
    crop_options = ['All'] +
sorted(df_raw['Crop'].unique().tolist())
    region_options = ['All'] +
sorted(df_raw['Region'].unique().tolist())
    soil_options = ['All'] +
sorted(df_raw['Soil_Type'].unique().tolist())

    selected_crop = st.selectbox('Pilih Tanaman',
crop_options)
    selected_region = st.selectbox('Pilih Wilayah',
region_options)
    selected_soil = st.selectbox('Pilih Jenis Tanah',
soil_options)

    filtered = df_raw.copy()
    if selected_crop != 'All':
        filtered = filtered[filtered['Crop'] ==
selected_crop]

```

```

        if selected_region != 'All':
            filtered = filtered[filtered['Region'] ==
selected_region]
        if selected_soil != 'All':
            filtered = filtered[filtered['Soil_Type'] ==
selected_soil]

        st.write(f"{len(filtered)} data ditemukan")
        st.dataframe(filtered.astype(str))

# Tab 4: Konfigurasi Kriteria
with tab4:
    st.subheader("⚙️ Konfigurasi SAW")
    st.write("Yuk bantu Lucas cari tanaman yang nggak drama,
cuma butuh air dan cinta 💧💕")
    st.markdown("Atur bobot dan jenis kriteria (Benefit /
Cost). Total bobot harus = 1.0 yaa!")
    st.image("Projek/Lucas.jpg", width=200)
    updated_criteria = {}
    total_weight = 0

    for k in default_criteria:
        st.markdown(f"#### {k}")
        col1, col2 = st.columns(2)
        with col1:
            is_benefit = st.radio(f"Tipe {k}", ["Benefit",
"Cost"],
                                index=0 if
default_criteria[k]['is_benefit'] else 1,
                                key=f"type_{k}")
            is_benefit_bool = is_benefit == "Benefit"
        with col2:
            weight = st.number_input(f"Bobot {k}", 0.0, 1.0,
default_criteria[k]['weight'], 0.01, key=f"weight_{k}")
            total_weight += weight
            updated_criteria[k] = {'is_benefit':
is_benefit_bool, 'weight': weight}

        if abs(total_weight - 1.0) > 0.001:
            st.warning("⚠️ Total bobot harus = 1.0 (sekarang
{:.2f})".format(total_weight))
        else:
            st.success("✅ Total bobot valid!")
            st.session_state.criteria_config = updated_criteria

# Tab 5: Hasil SAW
with tab5:
    st.subheader("📋 Hasil SAW")
    criteria = [(k, v['is_benefit']) for k, v in
st.session_state.criteria_config.items()]
    weights = {k:
st.session_state.criteria_config[k]['weight'] for k, _ in

```

```

criteria}

    data_saw = df[[col for col, _ in criteria]].copy()
    normalized = data_saw.copy()

    for col, is_benefit in criteria:
        if is_benefit:
            normalized[col] = data_saw[col] /
data_saw[col].max()
        else:
            normalized[col] = data_saw[col].min() /
data_saw[col]

    normalized['SAW_Score'] = sum(normalized[col] *
weights[col] for col, _ in criteria)
    df_result = df.copy()
    df_result['SAW_Score'] = normalized['SAW_Score']
    df_result['Ranking'] =
df_result['SAW_Score'].rank(ascending=False, method='min')
    df_sorted = df_result.sort_values(by='SAW_Score',
ascending=False)

    df_sorted = decode_labels(df_sorted, label_cols,
le_dict)

    st.markdown("### Konfigurasi")
    st.dataframe(pd.DataFrame({
        'Kriteria': [k for k, _ in criteria],
        'Tipe': ['Benefit' if b else 'Cost' for _, b in
criteria],
        'Bobot': [weights[k] for k, _ in criteria]
    })))

    st.markdown("### Top 10 Rekomendasi")
    top_cols = ['Crop', 'Region', 'Soil_Type',
'Yield_tons_per_hectare', 'SAW_Score', 'Ranking']
    st.dataframe(df_sorted[top_cols].head(10).astype(str))

    st.subheader("📌 Kesimpulan")
    top_crop = df_sorted.iloc[0]['Crop']
    kriteria_terpenting = max(weights, key=weights.get)
    st.markdown(f"""
- 🌾 Tanaman terbaik: **{top_crop}**
- 🍎 Kriteria paling penting: **{kriteria_terpenting}**
(bobot {weights[kriteria_terpenting]:.2f})
""")

```

projek.py

2.3.5 Screenshot Program

- Tampilan Awal



Aplikasi SAW Hasil Panen Tanaman

Melania Intan Sagita

123230005


Fadiyah Nur Sabiyyah

123230006

Deskripsi

Aplikasi ini menggunakan metode Simple Additive Weighting (SAW) untuk membantu menentukan tanaman terbaik berdasarkan hasil panen dan beberapa kriteria penting seperti curah hujan, suhu, pupuk, irigasi, waktu panen, dan hasil produksi.

- Informasi Dataset Mentah



Informasi Dataset


Jumlah data mentah ditampilkan

5

	Region	Soil_Type	Crop	Rainfall_mm	Temperature_Celsius	Fertilizer_Used	Irrigation_U
0	West	Sandy	Cotton	897.0772391101236	27.676966373377603	False	True
1	South	Clay	Rice	992.6732816189208	18.02614225436302	True	True
2	North	Loam	Barley	147.9980252926104	29.79404241557257	False	False
3	North	Sandy	Soybean	986.8663313367324	16.64419019137728	False	True
4	South	Silt	Wheat	730.379174445627	31.6206873708058	True	True

- Dimensi dan Nilai Kosong

 Dimensi: (1000000, 10)

 Nilai Kosong:


	0
Region	0
Soil_Type	0
Crop	0
Rainfall_mm	0
Temperature_Celsius	0
Fertilizer_Used	0
Irrigation_Used	0
Weather_Condition	0
Days_to_Harvest	0
Yield_tons_per_hectare	0

- Statistik Deskriptif



Statistik Deskriptif:

	Rata-rata	Median	Standar Deviasi	Minimum	Maksimum
Rainfall_mm	549.9819	550.1241	259.8513	100.0009	999.9981
Temperature_Celsius	27.505	27.5074	7.2206	15	40
Days_to_Harvest	104.495	104	25.9534	60	149
Yield_tons_per_hectare	4.6495	4.6518	1.6966	-1.1476	9.9634

- Dataset yang sudah Preprocessing

 **Dataset Sudah Preprocessing**

Jumlah data preprocessed ditampilkan

5  

	Region	Soil_Type	Crop	Rainfall_mm	Temperature_Celsius	Fertilizer_Used	Irrigation_Used	Weather_Condition
0	3	4	1	0.8856	0.5071	0	1	
1	2	1	3	0.9919	0.121	1	1	
2	1	2	0	0.0533	0.5918	0	0	
3	1	4	4	0.9854	0.0658	0	1	
4	2	5	5	0.7004	0.6648	1	1	

- Dimensi dan Nilai Kosong

🌐 Dimensi: (1000000, 10)

🔍 Nilai Kosong:

	0
Region	0
Soil_Type	0
Crop	0
Rainfall_mm	0
Temperature_Celsius	0
Fertilizer_Used	0
Irrigation_Used	0
Weather_Condition	0
Days_to_Harvest	0
Yield_tons_per_hectare	0

- Statistik Deskriptif

Statistik Deskriptif:

	Rata-rata	Median	Standar Deviasi	Minimum	Maksimum
Region	1.5005	2	1.1178	0	3
Soil_Type	2.5006	3	1.7081	0	5
Crop	2.4994	2	1.7078	0	5
Rainfall_mm	0.5	0.5001	0.2887	0	1
Temperature_Celsius	0.5002	0.5003	0.2888	0	1
Fertilizer_Used	0.4999	0	0.5	0	1
Irrigation_Used	0.4995	0	0.5	0	1
Weather_Condition	1.0011	1	0.8164	0	2
Days_to_Harvest	0.4999	0.4944	0.2916	0	1
Yield_tons_per_hectare	0.5217	0.522	0.1527	0	1

- Mapping LabelEncoder

Lihat Mapping LabelEncoder

Mapping kolom **Region**:

	Label	Encoded
0	East	0
1	North	1
2	South	2
3	West	3

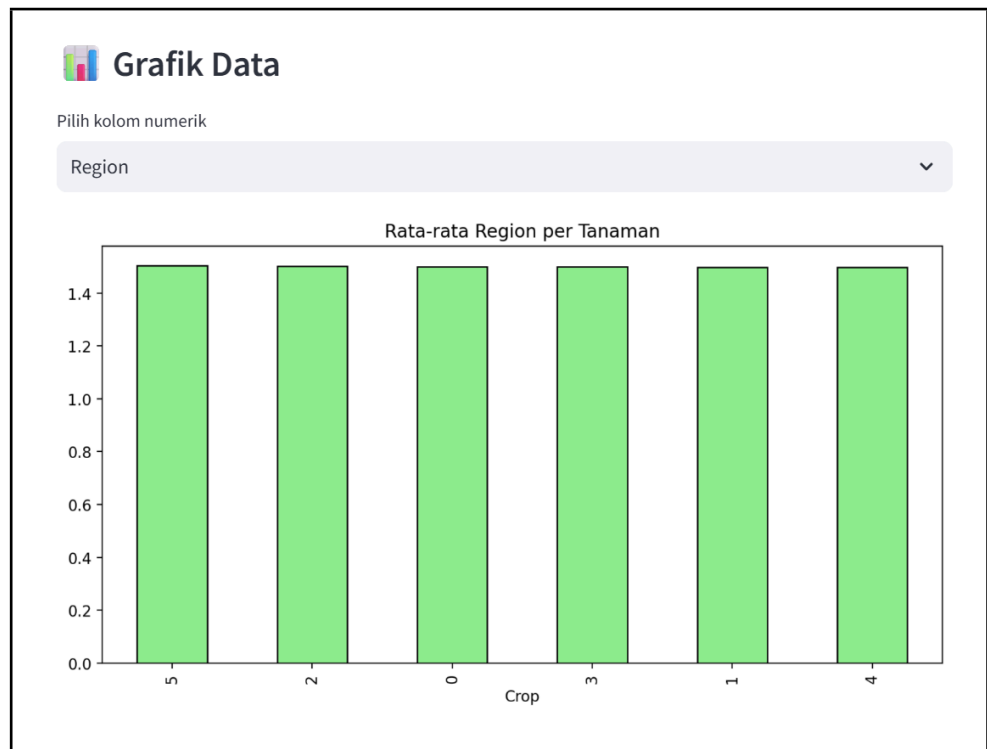
Mapping kolom **Soil_Type**:

	Label	Encoded
0	Chalky	0
1	Clay	1
2	Loam	2
3	Peaty	3
4	Sandy	4
5	Silt	5


Mapping kolom **Crop**:

	Label	Encoded
0	Barley	0
1	Cotton	1
2	Maize	2
3	Rice	3
4	Soybean	4
5	Wheat	5


- Grafik




- Cari Data (Pemilihan)

 **Cari Data dari Dataset Mentah**


Pilih Tanaman

All 

Pilih Wilayah

All 

Pilih Jenis Tanah

All 

- Cari Data (Hasil)

1000000 data ditemukan


	Region	Soil_Type	Crop	Rainfall_mm	Temperature_Celsius	Fertilizer_Used	Irrigation_Used
0	West	Sandy	Cotton	897.0772391101236	27.676966373377603	False	True
1	South	Clay	Rice	992.6732816189208	18.02614225436302	True	True
2	North	Loam	Barley	147.9980252926104	29.79404241557257	False	False
3	North	Sandy	Soybean	986.8663313367324	16.64419019137728	False	True
4	South	Silt	Wheat	730.379174445627	31.6206873708058	True	True
5	South	Silt	Soybean	797.4711823962564	37.70497446941277	False	True
6	West	Clay	Wheat	357.90235724297685	31.59343138976995	False	False
7	South	Sandy	Rice	441.13115357285005	30.88710699523619	True	True
8	North	Silt	Wheat	181.5878606243205	26.752728580811905	True	False
9	West	Sandy	Wheat	395.0489682684721	17.646198956255336	False	True

- Konfigurasi SAW

⚙️ Konfigurasi SAW

Yuk bantu Lucas cari tanaman yang nggak drama, cuma butuh air dan cinta 💧❤️

Atur bobot dan jenis kriteria (Benefit / Cost). Total bobot harus = 1.0 yaa!



Rainfall_mm

Tipe Rainfall_mm

☒ Benefit

☐ Cost

Bobot Rainfall_mm

0.15 - +

Temperature_Celsius

Tipe Temperature_Celsius

☒ Benefit

☐ Cost

Bobot Temperature_Celsius

0.15 - +

Fertilizer_Used

Tipe Fertilizer_Used

☒ Benefit

☐ Cost

Bobot Fertilizer_Used


0.10 - +

- Hasil SAW

<div>  Hasil SAW </div>			
Konfigurasi			
	Kriteria	Tipe	Bobot
0	Rainfall_mm	Benefit	0.15
1	Temperature_Celsius	Benefit	0.15
2	Fertilizer_Used	Benefit	0.1
3	Irrigation_Used	Benefit	0.1
4	Days_to_Harvest	Cost	0.2
5	Yield_tons_per_hectare	Benefit	0.3

- Rekomendasi Top 10 dan Kesimpulan

Top 10 Rekomendasi						
	Crop	Region	Soil_Type	Yield_tons_per_hectare	SAW_Score	Ranking
937201	Maize	South	Clay	0.9688996589283432	0.7862179192468861	1.0
732933	Soybean	North	Peaty	0.9616658884651132	0.785562348281604	2.0
564030	Cotton	North	Sandy	0.9677049342338968	0.7846197419789576	3.0
230072	Cotton	North	Loam	0.9715866153758096	0.7832285958786835	4.0
901176	Barley	North	Loam	0.960491623927022	0.7823285286721942	5.0
881804	Barley	West	Silt	0.9646355468692056	0.7817432837008305	6.0
477139	Rice	East	Sandy	0.9637567482276403	0.7816867031877685	7.0
252308	Soybean	South	Sandy	0.961117836626868	0.7813696392998857	8.0
780321	Rice	South	Chalky	0.9603662320538549	0.7812149704769911	9.0
868145	Rice	South	Chalky	1.0	0.7803374390432614	10.0

<div>  Kesimpulan </div>	
<ul style="list-style-type: none"> 🌱 Tanaman terbaik: Maize ❤️ Kriteria paling penting: Yield_tons_per_hectare (bobot 0.30) 	

BAB III

JADWAL Pengerjaan dan Pembagian Tugas

3.1 Jadwal Pengerjaan

Tabel 3.1 Jadwal Pengerjaan

No	Kegiatan	2025			
		Mei			
		1	2	3	4
1	Menentukan tema dan ide				
2	Pembuatan Program				
3	Pembuatan Laporan				

3.2 Pembagian Tugas

Tabel 3.2 Pembagian Tugas

No	Kegiatan	Penanggung Jawab
1	Menentukan tema dan ide	Melania dan Fadilah
2	Pembuatan Program	Melania dan Fadilah
3	Pembuatan Laporan	Melania dan Fadilah

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Sistem ini dirancang untuk memfasilitasi pengambilan keputusan dalam memilih hasil panen terbaik berdasarkan data agrikultur dan cuaca. Melalui penggunaan metode *Simple Additive Weighting* (SAW) yang diimplementasikan pada aplikasi berbasis Streamlit, sistem ini mampu menghitung skor dari berbagai alternatif hasil panen dengan mempertimbangkan beberapa kriteria seperti curah hujan, suhu, penggunaan pupuk dan irigasi, lama panen, dan hasil produksi per hektar.

Eksperimen menunjukkan bahwa pendekatan SAW mampu memberikan rekomendasi hasil panen secara objektif dan terukur. Aplikasi ini memberikan kemudahan dalam pengaturan bobot kriteria serta visualisasi data, sehingga mempermudah pengguna dalam mengambil keputusan tanpa harus melakukan analisis manual yang kompleks. Dengan demikian, sistem ini tidak hanya meningkatkan efisiensi proses pengambilan keputusan, tetapi juga menunjukkan potensi penerapan metode pendukung keputusan pada sektor pertanian digital berbasis data.

4.2 Saran

Untuk pengembangan lebih lanjut, sistem ini dapat ditingkatkan dengan menambahkan jumlah dan variasi data, baik dari segi waktu, lokasi, maupun jenis tanaman, agar hasil rekomendasi menjadi lebih akurat dan representatif. Selain itu, penggunaan metode pembobotan lain seperti AHP dapat dipertimbangkan untuk memberikan hasil yang lebih sistematis dan berbobot. Pengembangan fitur prediktif seperti estimasi hasil panen berdasarkan data cuaca dan kondisi tanah juga akan sangat berguna untuk perencanaan jangka panjang. Antarmuka pengguna sebaiknya dibuat lebih responsif dan mudah digunakan, terutama bagi pengguna dari kalangan non-teknis seperti petani atau pengambil kebijakan lapangan. Terakhir, akan lebih baik jika aplikasi ini dapat diakses secara daring (*online*) sehingga bisa digunakan lebih luas tanpa terbatas pada perangkat lokal.

DAFTAR PUSTAKA

Internet

- [1] Dataset *Crop Recommendation Dataset*,
(<https://www.kaggle.com/datasets/samuelotiattakorah/agriculture-crop-yield>)
- [2] Nurhasanah dkk. 2020. Sistem Pendukung Keputusan Pemilihan Mahasiswa Berprestasi dengan Metode Simple Addictive Weighting. *TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika*, Vol 7, No.2, (<https://jurnal.stmcileungsi.ac.id/index.php/tekno/article/view/18>, diakses pada 29 Mei 2025).
- [3] Ristiana Rizka dan Jumardi Yuwan. 2021. Sistem Pendukung Keputusan Pemilihan Paket Wedding Organizer Menggunakan Metode SAW (Simple Additive Weighting). *Jurnal SISFOKOM (Sistem Informasi dan Komputer)* , Vol 10, No.01, (<https://jurnal.atmaluhur.ac.id/index.php/sisfokom/article/view/946>, diakses pada 29 Mei 2025).