
Sistem Pendukung Keputusan Pemilihan Prodi Fakultas Sains dan Teknologi di Universitas Sari Mulia

Decision Support System for Faculty Of Science and Technology Program Selection at Sari Mulia University

Zaid¹, Gregorius Amori², Arpan Muallief Saprizal³, Syarifuddin⁴, Nor Anisa⁵

¹²³⁴⁵ Sistem Informasi, Fakultas Sains & Teknologi, Universitas Sari Mulia, Pal 6 Jl. Pramuka, Banjarmasin Timur, 70238, Indonesia.

Email : zaid@student.unism.ac.id¹, gregorius.amori@student.unism.ac.id², arpan.muallief@student.unism.ac.id³, syarifuddin@student.unism.ac.id⁴, noranisa@unism.ac.id⁵

ABSTRAK

Sistem pendukung keputusan merupakan salah satu alat terbaik dalam membantu manusia dalam memutuskan sebuah keputusan yang bijak. Universitas Sari Mulia dewasa ini memiliki tiga fakultas dan salah satunya adalah Fakultas Sains dan Teknologi. Fakultas ini untuk sekarang sudah memiliki tiga program studi yang hampir mirip. Untuk itu diperlukan sebuah sistem pendukung keputusan untuk membantu calon mahasiswa baru dalam memilih prodi di fakultas ini yang sesuai dengan diri mereka. Sistem ini menggunakan bahasa pemrograman python yang dibantu dengan bantuan library numpy dan pandas untuk memproses data dan melakukan penghitungan serta matplotlib dan seaborn untuk memvisualisasikan data dari hasil penghitungan metode SAW.

Kata Kunci: Python, Simple Additive Weighting, Kriteria, Bobot, Alternatif

ABSTRACT

Decision support systems are one of the best tools to help humans make wise decisions. Sari Mulia University currently has three faculties and one of them is the Faculty of Science and Technology. This faculty currently has three study programs that are almost similar. For this reason, a decision support system is needed to help prospective new students choose a study program at this faculty that suits them. This system uses the Python programming language which is assisted by the numpy and pandas libraries to process data and perform calculations as well as matplotlib and seaborn to visualize data from the results of SAW calculations

Keyword: Python, Simple Additive Weighting, Criteria, Weights, Alternatives



Creative Commons License

Artikel ini berlisensi Creative Common Attribution-ShareAlike 4.0 International

Pendahuluan

Dewasa ini, Universitas Sari Mulia sudah memiliki tiga fakultas, di antaranya adalah Fakultas Kesehatan, Fakultas Humaniora, dan Fakultas Sains dan Teknologi. Fakultas Sains dan Teknologi merupakan fakultas yang menyediakan tiga program studi di

lingkup sains dan komputer, di antaranya adalah Program Studi Sarjana Sistem Informasi, Program Studi Sarjana Teknologi Informasi, dan Program Studi Sarjana Teknik Industri. Namun, kurangnya informasi membuat calon mahasiswa baru menjadi kesulitan untuk menentukan pilihan program

studinya.

Maka dari itu diperlukan suatu sistem yang berbasis komputer untuk menentukan mana pilihan program studi yang terbaik berdasarkan penilaian dari potensi yang dimiliki oleh calon mahasiswa baru Fakultas Sains dan Teknologi Universitas Sari Mulia.

Guna menyelesaikan masalah ini dibuatlah sebuah sistem pendukung keputusan untuk memberikan rekomendasi-rekomendasi program studi di Fakultas Sains dan Teknologi. Sistem Pendukung Keputusan (SPK) adalah sistem berbasis komputer interaktif yang dapat membantu para pengambil keputusan dengan cara memanfaatkan data dan model untuk menyelesaikan masalah-masalah baik yang tak terstruktur maupun semi terstruktur (Made et al., 2023) dengan Python sebagai bahasa pemrogramannya.

Python merupakan sebuah bahasa pemrograman populer yang multifungsi dan mudah untuk dipelajari, bahasa pemrograman ini bekerja dengan cara menerjemahkan *bytecode* sebelum diterjemahkan ke dalam bahasa manusia (Nyoman Tri Anindia Putra et al., 2023). Python adalah sebuah bahasa pemrograman yang dikembangkan oleh Guido van Rossum pada tahun 80an di abad ke-20 (Kalyani Jeslyn Lim et al., 2023).

Penulis menggunakan Jupyter Notebook untuk menjalankan kode python. Jupyter Notebook adalah sebuah program perangkat lunak yang ditemukan pada tahun 2015 yang ditujukan untuk *data scientist* dalam berbagi dokumen berisikan kode, hasil perhitungan dan sebagainya. (Asyrofi & Asyrofi, 2023)

Metode

Metode yang digunakan dalam penelitian ini adalah metode SAW. Metode *Simple Additive Weighting* atau bisa disingkat dengan SAW adalah suatu metode penjumlahan yang memiliki nilai bobot. Metode ini mencari bobot nilai terbesar dari rating kinerja pada setiap alternatif di semua atribut yang tersedia. Metode SAW memerlukan proses normalisasi matriks keputusan terlebih dahulu untuk nantinya dibandingkan dengan semua rating alternatif yang ada (Gemawaty & Yuliani, 2023)(Sembiring et al., 2020)(Setiadi et al., 2018).

Dalam metode SAW terdapat langkah-langkah (Aprianti, 2017)(Sonata et al., 2016) dalam perhitungannya sebagai berikut:

1. Menentukan nama dan jumlah alternatif, yaitu dapat ditulis A_i
2. Menentukan nama dan jumlah kriteria agar nantinya dapat dijadikan acuan dalam pengambilan keputusan C_i
3. Menentukan bobot untuk masing-masing kriteria yang telah dibuat. Dapat ditulis dengan W .
4. Membuat matriks keputusan yang telah dibentuk dari tabel rating kecocokan dari setiap alternatif yang ada.
5. Melakukan normalisasi matriks keputusan.
6. Menampilkan skor SAW.



Gambar 1. Tahapan Sistem Pendukung Keputusan

Hasil

Untuk melakukan penelitian dengan metode SAW, penulis membuat sebuah dataset terlebih dahulu. Dataset tersebut berisikan alternatif, kriteria, dan bobot kriteria.

Sebelum membuat dataset, penulis meng-*import* terlebih dahulu *library* yang dibutuhkan untuk membangun sebuah sistem pendukung keputusan ini, yakni Numpy, Pandas, Matplotlib.pyplot, dan Seaborn.

```

# Mengimport library yang dibutuhkan
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
    
```

Gambar 2. Kode *Import Library*

Setelah itu, penulis membuat jumlah alternatif yang ditentukan adalah sesuai dengan program studi yang ada di Universitas Sari Mulia.

Tabel 1. Daftar Alternatif

No.	Nama Alternatif
1.	S1 Sistem Informasi
2.	S1 Teknologi Informasi
3.	S1 Teknik Industri

Selanjutnya, penulis menentukan tiga kriteria beserta bobotnya masing-masing.

Tabel 2. Daftar Kriteria dan Bobot

No.	Kriteria	Bobot
1.	Minat	0,4
2.	Pengetahuan terhadap prodi	0,3
3.	Kemampuan	0,3

Alternatif beserta kriteria tersebut kemudian diimplementasikan ke dalam kode python.

```

# Dataset
alternatif = ['Sistem Informasi', 'Teknologi Informasi', 'Teknik Industri']
jumlah_alternatif = len(alternatif)
jumlah_kriteria = 3
nama_kriteria = ['Minat', 'Pengetahuan Terhadap Prodi', 'Kemampuan']
bobot_kriteria = np.array([0.4, 0.3, 0.3])
    
```

Gambar 3. Kode Dataset

Agar sebuah sistem pendukung keputusan ini interaktif dengan pengguna maka ditambahkan sebuah kode *input*. Pengguna sistem ini diharuskan untuk mengisi nilai masing-masing kriteria untuk setiap alternatif melalui proses *input*.

```

# Input nilai kriteria dari user
kriteria = []
for i in range(jumlah_kriteria):
    kriteria.append([])
    for j in range(jumlah_alternatif):
        nilai = float(input(f'Masukkan nilai {nama_kriteria[i]} untuk alternatif {alternatif[j]}: '))
        kriteria[i].append(nilai)
    
```

Gambar 4. Kode *Input*

Kode dilanjutkan dengan membuat sebuah *data frame* dan kode untuk menormalisasi matriks keputusannya.

```
# Membuat DataFrame
data = {'Prodi': alternatif}
for i in range(jumlah_kriteria):
    data[nama_kriteria[i]] = kriteria[i]
df = pd.DataFrame(data)
```

Gambar 5. Kode Data Frame

```
# Normalisasi matriks keputusan
def normalisasi(df):
    df_normalized = df.copy()
    for column in df.columns[1:]:
        if column in [nama_kriteria[0], nama_kriteria[1]]:
            # Benefit
            df_normalized[column] = df[column] / df[column].max()
        else:
            # Cost
            df_normalized[column] = df[column].min() / df[column]
    return df_normalized

df_normalized = normalisasi(df)
```

Gambar 6. Kode Normalisasi Matriks Keputusan

Nilai yang telah diproses melalui normalisasi matriks keputusan tersebut dikalikan dengan bobot kriteria kemudian ditampilkan sesuai urutan alternatif. Nilai juga ditampilkan secara berurutan mulai dari skor tertinggi. Agar memudahkan pengguna dalam membaca hasil rekomendasi program studi maka ditambahkan kode untuk visualisasi data.

```
# Menghitung skor SAW
df_normalized['Skor_SAW'] = df_normalized.iloc[:, 1:].dot(bobot_kriteria)

# Menampilkan hasil akhir
print("Hasil akhir perhitungan SAW:")
print(df_normalized[['Prodi', 'Skor_SAW']])

# Mengurutkan berdasarkan skor tertinggi
df_sorted = df_normalized.sort_values(by='Skor_SAW', ascending=False)
print("Hasil matriks berdasarkan peringkat:")
print(df_sorted[['Prodi', 'Skor_SAW']])

# Visualisasi
sns.barplot(x='Skor_SAW', y='Prodi', data=df_sorted)
plt.title("Hasil Perhitungan SAW untuk Penentuan Prodi di Fakultas Sains dan Teknologi Universitas Sari Mulia")
plt.show()
```

Gambar 7. Kode Menampilkan Hasil

Dengan demikian, jika kode ini dijalankan oleh pengguna maka pengguna diminta memasukkan nilai kriteria di masing-masing alternatif berdasarkan skala 1 sampai 10. Penulis mencoba kode sistem pendukung keputusan ini dengan memasukkan nilai kriteria untuk masing-masing alternatif sesuai tabel di bawah ini:

Tabel 3. Alternatif dan Nilai Kriteria

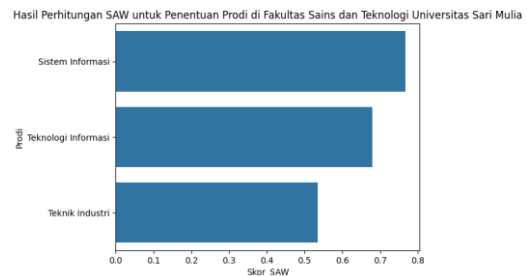
Nama Alternatif (C1, C2, C3)
S1 Sistem Informasi (10, 8, 9)
S1 Teknologi Informasi (7, 8, 6)
S1 Teknik Industri (4, 2, 2)

Sehingga ketika kode telah dijalankan maka akan mendapatkan hasil berupa skor SAW dalam bentuk tulisan maupun grafik.

```
Hasil akhir perhitungan SAW:
      Prodi Skor_SAW
0  Sistem Informasi 0.766667
1  Teknologi Informasi 0.680000
2  Teknik Industri 0.535000

Alternatif berdasarkan peringkat:
      Prodi Skor_SAW
0  Sistem Informasi 0.766667
1  Teknologi Informasi 0.680000
2  Teknik Industri 0.535000
```

Gambar 8. Skor SAW



Gambar 9. Grafik Skor SAW

Dengan demikian sesuai dengan gambar 9, program studi yang paling direkomendasikan berdasarkan nilai-nilai kriteria pada tabel 3 adalah S1 Sistem Informasi dan disusul oleh S1 Teknologi Informasi dan S1 Teknik Industri secara berurutan.

Pembahasan

Dalam sistem pendukung keputusan yang dibuat, kode didahului dengan meng-*import library* ke dalam python. Hal ini dilakukan guna membuat

kode python dalam sistem pendukung keputusan ini dapat berjalan dengan baik. *Library* adalah sebuah sumber program kode tambahan yang digunakan untuk berbagai kebutuhan dalam pemrograman. Terdapat 140 ribu lebih *library* yang tersedia dalam bahasa pemrograman ini untuk mengembangkan berbagai macam proyek pemrograman (Sadli, 2022). Dalam penelitian ini digunakan beberapa *library* seperti Numpy yang berfungsi untuk penghitungan angka dan data sains, selain itu Pandas berfungsi untuk memproses suatu data.

Kode-kode python yang dibuat akan memproses semua *input* yang dimasukkan oleh pengguna untuk menghasilkan sebuah rekomendasi program studi yang ada di Fakultas Sains dan Teknologi di Universitas Sari mulia berdasarkan kriteria yang telah disediakan.

Kesimpulan

Dengan demikian, terdapat beberapa kesimpulan yang dapat ditarik dalam penelitian ini yaitu:

1. Sistem pendukung keputusan ini ditujukan untuk calon mahasiswa baru di Universitas Sari Mulia dalam membantu mereka untuk memilih program studi Fakultas Sains dan Teknologi mana yang sesuai dengan diri mereka.
2. Metode yang digunakan dalam penelitian ini adalah metode SAW yang memungkinkan untuk menghitung setiap nilai kriteria pada masing-masing alternatif dengan mengalikan bobot kriteria.
3. Dalam pengembangan sistem pendukung keputusan ini hanya

digunakan satu bahasa pemrograman yakni python.

Adapun kelebihan yang didapatkan dari sistem pendukung keputusan ini, yaitu:

1. Penggunaan program yang sangat mudah. Pengguna hanya perlu memasukkan *input* berupa nilai kriteria pada setiap alternatif.
2. Program menghasilkan rekomendasi sesuai dengan minat, pengetahuan pengguna terhadap prodi, dan kemampuan.

Selain itu, kekurangan dari program ini adalah sebagai berikut:

1. Program yang dibuat masih memiliki kriteria yang sangat sedikit sehingga rekomendasi yang diperoleh mungkin saja masih tidak sesuai dengan kebutuhan pengguna.
2. Program yang dibuat masih sebatas kode dan tidak memiliki UI yang menarik.

Saran untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Menambahkan lebih banyak kriteria agar keputusan yang dihasilkan lebih sesuai dengan diri pengguna.
2. Menambahkan metode lain agar sistem pendukung keputusan menghasilkan rekomendasi yang lebih baik kepada pengguna.
3. Mengintegrasikan kode python dengan web atau aplikasi mobile agar lebih memudahkan pengguna dalam menjalankan program ini.

Referensi

- Aprianti, H. (2017). Sistem Pendukung Keputusan Pemilihan Smartphone dengan Menerapkan Metode Simple Additive Weighting (SAW). *Jurnal Sistem Informasi*.
- Asyrofi, R. R., & Asyrofi, R. (2023). IMPLEMENTASI APLIKASI JUPYTER NOTEBOOK SEBAGAI ANALISIS KRETERIA PLAGIASI DENGAN TEKNIK SIMANTIK. *JIPi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 8(2), 627–637. <https://doi.org/10.29100/jipi.v8i2.3699>
- Gemawaty, C. A., & Yuliani, Y. (2023). Ciptaan disebarluaskan di bawah Lisensi Creative Commons Atribusi 4.0 Internasional. PEMILIHAN DOSEN TERBAIK DENGAN METODE SAW (SIMPLE ADDITIVE WEIGHTING). *Journal of Information System, Applied, Management, Accounting and Research*, 7(3), 711–717. <https://doi.org/10.52362/jisamar.v7i3.1159>
- Kalyani Jeslyn Lim, Clement Nathanael, Felicia Angel Wijaya, Jeson Adhi Dharma, Thaddeus Kendrick Andrian, Wilsen Soetresno, & Rahmi Yulia Ningsih. (2023). Penggunaan Bahasa Pemrograman Python Untuk Memvisualisasikan Data Peluang Selamat Dari Kecelakaan Titanic. *Jurnal Publikasi Teknik Informatika*, 2(2), 66–79. <https://doi.org/10.55606/jupti.v2i2.1735>
- Made, I., A1, S., Ayu, I., Dewi, U., Kadek, I., & Asmarajaya, A. (2023). Penerapan SPK dengan Metode SAW Terhadap Pemilihan Calon Karyawan LPD Desa Adat Sidakarya dengan Bahasa Pemrograman Python.
- Nyoman Tri Anindia Putra, I., Sepdyana Kartini, K., Kristian Suyitno, Y., Made Sugiarta, I., & Kadek Era Puspita, N. (2023). Penerapan Library Tensorflow, Cvzone, dan Numpy pada Sistem Deteksi Bahasa Isyarat Secara Real Time (Vol. 2, Issue 3). <https://ejournal.sidyanusa.org/index.php/jkdn>
- Sadli, A. (2022). Using The Python Library to Create Simple Game Animations. *International Journal of Management Science and Information Technology*, 2(2), 21–31. <https://doi.org/10.35870/ijmsit.v2i2.699>
- Sembiring, F., Fauzi, M. T., Khalifah, S., Khotimah, A. K., & Rubiati, Y. (2020). Sistem Pendukung Keputusan Penerima Bantuan Covid 19 menggunakan Metode Simple Additive Weighting (SAW) (Studi Kasus : Desa Sundawenang). 11.
- Setiadi, A., Ratna Ningsih, A., Studi Manajemen Informatika, P., BSI Karawang, A., Studi Teknik Informatika, P., Nusa Mandiri Jakarta, S., & Studi Sistem Informasi, P. (2018). Penerapan Metode Simple Additive Weighting(SAW) Untuk Pemilihan Siswa Terbaik. In *Jurnal SISFOKOM* (Vol. 07).
- Sonata, F., Tinggi, S., & Komputer, I. (2016). IMPLEMENTASI METODE SIMPLE ADDITIVE WEIGHTING (SAW) DENGAN PROSES FUZZIFIKASI DALAM PENILAIAN KINERJA DOSEN IMPLEMENTATION SIMPLE ADDITIVE

*WEIGHTING (SAW) METHOD WITH
FUZZIFICATION PROCESS IN LECTURER
PERFORMANCE ASSESSMENT (Vol. 5,
Issue 2).*