

LAPORAN PROYEK SISTEM PENDUKUNG KEPUTUSAN

PENGAMBILAN MATA KULIAH PILIHAN SEMESTER LIMA DI

JURUSAN TEKNIK INFORMATIKA FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA



Disusun oleh:

M. Rabyndra Janitra Binello	09021282328074
Monica Amrina Rosyada	09021382328144
Nabila Ayu Talita	09021382328158
Nabilla Kesuma	09021382328132
Zsa Zsa Aulia Az Zahrah	09021382328136

Dosen Pengampu:

Yunita, S.Si., M.Cs.

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2025

1. Latar Belakang

Dalam pendidikan tinggi, kurikulum dirancang untuk memberikan keseimbangan antara mata kuliah wajib dan mata kuliah pilihan agar mahasiswa dapat mengembangkan kompetensi sesuai minat serta tujuan karir mereka. Kurikulum 2021 pada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya juga menerapkan prinsip tersebut, dimana mahasiswa bebas memilih mata kuliah pilihan yang mendukung bidang keahlian tertentu. Pemilihan ini memiliki peran penting karena mempengaruhi kesiapan akademik, beban studi, serta relevansinya terhadap kebutuhan industri teknologi informasi di masa depan.

Pada semester lima, mahasiswa mulai berhadapan dengan sejumlah alternatif mata kuliah pilihan yang memiliki karakteristik berbeda, seperti beban SKS, prasyarat, kesesuaian dengan peminatan, dan kontribusi terhadap prospek karier. Namun, banyak mahasiswa masih mengalami kesulitan dalam menentukan mata kuliah yang paling tepat. Pemilihan sering dipengaruhi faktor subjektif, seperti saran teman atau kecenderungan mengikuti kelompok, tanpa mempertimbangkan aspek akademik yang lebih penting. Hal ini sejalan dengan temuan Nurzaman dan Putri (2023) yang menunjukkan bahwa mahasiswa cenderung memilih mata kuliah pilihan tanpa melakukan analisis berbasis kriteria yang objektif, sehingga keputusan yang diambil kurang optimal bagi perkembangan akademik mereka.

Untuk mengatasi permasalahan tersebut, dibutuhkan sistem pendukung keputusan berbasis multi-kriteria. Pendekatan ini telah luas dimanfaatkan dalam konteks pendidikan untuk mendukung proses pemilihan alternatif secara objektif dan sistematis. Dalam hal ini, metode Stepwise Weight Assessment Ratio Analysis (SWARA) dapat diterapkan untuk menentukan bobot setiap kriteria berdasarkan penilaian pakar secara terstruktur. Penelitian menunjukkan bahwa SWARA mampu menghasilkan bobot kriteria yang seimbang dan terukur melalui proses penilaian berurutan, sehingga bobot yang dihasilkan lebih representatif terhadap tingkat kepentingannya (Ramadhani & Satria, 2024).

Setelah bobot kriteria berhasil ditentukan menggunakan SWARA, tahap selanjutnya adalah proses penilaian dan penentuan peringkat alternatif. Untuk itu, metode Simple Additive Weighting (SAW) dapat diterapkan untuk menilai dan merangking alternatif. SAW adalah metode pengambilan keputusan yang bekerja dengan menjumlahkan semua nilai kriteria yang

telah dikalikan dengan bobotnya, sehingga menghasilkan nilai akhir yang digunakan untuk peringkat (*ranking*) setiap alternatif (Taherdoost, 2023).

Dengan menggabungkan kedua metode ini, mahasiswa dapat memperoleh rekomendasi pemilihan mata kuliah pilihan yang lebih objektif, terukur, dan sesuai dengan kebutuhan akademik maupun prospek karier. Pendekatan ini diharapkan dapat membantu mahasiswa membuat keputusan yang lebih tepat pada semester lima, sehingga proses pembelajaran selanjutnya menjadi lebih terarah dan mendukung pencapaian kompetensi yang diperlukan.

2. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini, yaitu:

1. Bagaimana sistem mengatasi subjektivitas pembobotan kriteria melalui metode SWARA?
2. Bagaimana sistem menghasilkan peringkat rekomendasi mata kuliah melalui metode SAW?
3. Seberapa efektif integrasi SWARA dan SAW dalam memberikan rekomendasi yang objektif dan optimal?

3. Tujuan Proyek

Adapun tujuan dari penelitian ini adalah:

1. Menganalisis dan menerapkan metode SWARA untuk mengatasi subjektivitas dalam proses pembobotan kriteria pada sistem rekomendasi mata kuliah.
2. Mengembangkan sistem yang mampu menghasilkan peringkat rekomendasi mata kuliah menggunakan metode *Simple Additive Weighting* (SAW) secara terstruktur dan konsisten.
3. Mengevaluasi efektivitas integrasi metode SWARA dan SAW dalam menghasilkan rekomendasi mata kuliah yang lebih objektif, akurat, dan optimal bagi pengguna.

4. Metode SPK Yang Digunakan

4.1 Penjelasan Metode

Metode SWARA (*Step - wise Weight Assessment Ratio Analysis*) merupakan sebuah pendekatan analisis keputusan yang digunakan untuk mengevaluasi dan memilih alternatif

yang paling sesuai dalam konteks suatu masalah multi-kriteria (Sriyasa, 2023). Metode ini bekerja melalui tahapan penilaian yang terstruktur untuk membantu pengambil keputusan menentukan tingkat kepentingan dari setiap kriteria. Proses yang dilakukan mencakup penyusunan kriteria, pengurutan prioritas, pemberian bobot sesuai tingkat signifikansi, serta analisis perbandingan antar kriteria. Keunggulan metode ini terletak pada kemampuannya mengakomodasi ketidakpastian dalam preferensi pengguna dengan melibatkan penilaian dari ahli maupun pihak berkepentingan. Selain itu, fleksibilitas SWARA memungkinkan pembaruan bobot apabila terjadi perubahan kebutuhan atau kondisi yang mempengaruhi keputusan. Pendekatan bertahap yang disediakan SWARA juga membantu memberikan gambaran yang lebih komprehensif mengenai kontribusi setiap kriteria terhadap tujuan keseluruhan keputusan.

Metode SAW (*Simple Additive Weighting*) adalah metode pengambilan keputusan yang digunakan untuk menentukan alternatif terbaik dari beberapa pilihan berdasarkan sejumlah kriteria yang sudah diberi bobot (Taherdoost, 2023). Metode ini bekerja dengan menjumlahkan semua nilai kriteria yang telah dikalikan dengan bobotnya, sehingga menghasilkan nilai akhir yang digunakan untuk peringkat (*ranking*) setiap alternatif. Fleksibilitas dalam penyesuaian bobot juga membuatnya banyak digunakan dalam berbagai kasus pengambilan keputusan. Namun, kekurangan seperti sensitivitas terhadap normalisasi atau skala serta ketidakmampuan menangani hubungan antar kriteria tetap menjadi perhatian dalam penggunaannya.

Integrasi metode SWARA dan SAW memberikan pendekatan yang lebih kuat untuk permasalahan pengambilan keputusan multi-kriteria. SWARA berperan dalam menentukan bobot kriteria secara sistematis berdasarkan penilaian dan prioritas dari pemangku kepentingan, sehingga menghasilkan bobot yang lebih objektif. Setelah bobot diperoleh, SAW digunakan untuk menghitung skor akhir setiap alternatif agar dapat dilakukan perankingan. Kolaborasi kedua metode tersebut menggabungkan kekuatan SWARA dalam menyusun bobot secara rinci dengan kepraktisan SAW dalam penghitungan nilai akhir, sehingga menghasilkan proses rekomendasi yang lebih menyeluruh, terarah, dan optimal.

4.2 Tahapan Metode

Metode yang digunakan adalah kombinasi SWARA dan SAW, terdiri dari tiga fase utama, menentukan kriteria dan alternatif, pembobotan kriteria menggunakan SWARA dan perhitungan nilai perankingan alternatif menggunakan SAW.

1. Fase 1 - Menentukan Kriteria dan Alternatif

Pada tahap awal, ditentukan kriteria yang digunakan sebagai dasar penilaian terhadap setiap alternatif mata kuliah. Penetapan kriteria ini disesuaikan dengan kebutuhan mahasiswa dalam memilih mata kuliah yang relevan dan mampu mendukung pengembangan kompetensi, baik dari sisi minat, kemampuan akademik, maupun relevansinya terhadap dunia kerja.

2. Fase 2 - Menentukan Bobot Kriteria

Metode SWARA (*Stepwise Weight Assessment Ratio Analysis*) merupakan pendekatan penilaian bobot yang dilakukan secara bertahap berdasarkan urutan prioritas kriteria. Proses SWARA efektif digunakan ketika penilaian kriteria berasal dari pendapat ahli, sehingga bobot yang diperoleh lebih realistis dan sesuai konteks keputusan (Situmorang et al., 2025). Langkah-langkahnya meliputi:

- a. Mengurutkan kriteria berdasarkan tingkat kepentingannya menurut penilaian para ahli. Namun, untuk nilai rata-rata awal t_j dalam pengurutan kriteria ditentukan berdasarkan diskusi dari anggota kelompok mengenai tingkat kepentingan relatif setiap kriteria. Kriteria kemudian diurutkan dari nilai tertinggi (paling penting) ke nilai terendah.
- b. Menentukan nilai komparatif (s_j) dimana nilai subjektif yang ditetapkan melalui diskusi kelompok untuk mengukur seberapa jauh tingkat kepentingan kriteria saat ini berkurang dibandingkan dengan kriteria yang berada tepat di urutan sebelumnya. Nilai ini dipilih dari skala perbandingan tertentu (misalnya 0,1 hingga 0,9).
- c. Menghitung koefisien perbandingan (K_j) berdasarkan nilai komparatif (s_j) untuk menyesuaikan bobot awal. Berikut rumus yang digunakan dalam menghitung koefisien perbandingan:

$$k_j = \begin{cases} 1, & \text{\& } j=1 \\ s_j + 1, & \text{\& } j > 1 \end{cases}$$

- d. Menghitung koefisien berurutan dimana nilai bobot relatif dihitung secara berurutan dan bobot kriteria berikutnya dikurangi berdasarkan koefisien dan kriteria sebelumnya. Berikut rumus yang digunakan dalam menghitung koefisien berurutan:

$$k_j = \begin{cases} 1, & \text{\& } j=1 \\ \frac{k_j - 1}{k_j}, & \text{\& } j > 1 \end{cases}$$

- e. Menentukan bobot akhir (W_j) dimana koefisien berurutan (W_j) dinormalisasi untuk mendapatkan bobot akhir setiap kriteria, dengan semua bobotnya adalah 1. Berikut rumus yang digunakan dalam menghitung koefisien berurutan:

$$W_j = \frac{q_j}{\sum_{j=1}^n q_j}$$

3. Fase 3 - Menentukan Alternatif Terbaik

Setelah bobot diperoleh, selanjutnya dilakukan proses penentuan alternatif terbaik menggunakan metode SAW. Metode SAW (*Simple Additive Weighting*) digunakan untuk menentukan perangkian alternatif melalui proses perhitungan nilai preferensi yang sederhana dan transparan (Taherdoost, 2023). SAW bekerja dengan melakukan normalisasi data dan mengalikan setiap nilai dengan bobot kriteria. Langkah-langkahnya meliputi:

- Membuat matriks keputusan (X) yang berisi nilai rating kecocokan awal (X_{ij}) setiap alternatif (pilihan mata kuliah) terhadap setiap kriteria.
- Menentukan pembobotan menggunakan bobot SWARA yang dimana bobot W_j yang diperoleh dari SWARA digunakan untuk menghitung nilai preferensi di SAW. Setiap nilai pada matriks normalisasi akan dikalikan dengan bobot kriteria yang sesuai.
- Setelah menentukan pembobotan, selanjutnya dilakukan proses normalisasi terhadap nilai alternatif. Nilai X_{ij} akan diubah menjadi nilai ternormalisasi (r_{ij}) menggunakan rumus *Benefit* atau *Cost*. Tujuannya agar semua nilai berada pada skala yang sama, yaitu 0-1. Berikut rumus yang digunakan dalam mengubah nilai menjadi ternormalisasi dengan menggunakan *Benefit* atau *Cost*:

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max x_{ij}}, & \text{jika benefit} \\ \frac{\min x_{ij}}{x_{ij}}, & \text{jika cost} \end{cases}$$

- Menghitung nilai preferensi (V_i) dimana setiap alternatif dihitung dengan menjumlahkan hasil perkalian antara bobot akhir (W_j) dari SWARA dengan nilai ternormalisasi (r_{ij}). Berikut rumus yang digunakan dalam menghitung nilai preferensi:

$$V_i = \sum (W_j \times r_{ij})$$

- e. Menentukan perangkian dimana alternatif dengan nilai V_i tertinggi adalah alternatif terbaik.

4.3 Kelebihan dan Kekurangan

Setiap metode pengambilan keputusan memiliki keunggulan dan keterbatasannya masing-masing. Begitu juga dengan metode SWARA dan SAW yang digunakan dalam penelitian ini. Untuk memahami alasan pemilihannya serta efektivitasnya dalam proses rekomendasi, berikut penjelasan kelebihan dan kekurangan dari kedua metode tersebut:

No	Kelebihan SWARA	Kekurangan SWARA
1	Proses metode sederhana karena hanya memerlukan penilaian berurutan terhadap kriteria.	Akurasi bergantung pada kompetensi dan konsistensi penilaian para ahli.
2	Bobot kriteria ditentukan berdasarkan evaluasi ahli sehingga hasil lebih realistis dan sesuai kondisi nyata.	Memerlukan keahlian khusus dari pakar yang memahami konteks masalah.
3	Melibatkan pendapat ahli secara langsung dalam penentuan bobot.	Tinggi tingkat subjektivitas dimana hasil bisa bervariasi tergantung ahli yang terlibat.
4	Tidak memerlukan data besar karena fokus pada analisis kualitatif.	Kurang sesuai digunakan untuk analisis data kuantitatif kompleks atau berskala besar.
5	Menjaga konsistensi prioritas antar kriteria melalui penilaian berurutan.	Jika banyak ahli terlibat, proses diskusi untuk mencapai konsensus dapat memakan waktu.

Tabel 4.1 Kelebihan dan Kekurangan Metode SWARA

No	Kelebihan SAW	Kekurangan SAW
1	Hasil perhitungan transparan dan mudah diverifikasi.	Sangat sensitif terhadap metode normalisasi yang digunakan.

2	Mudah diimplementasikan baik secara manual maupun menggunakan <i>software</i> .	Perbedaan teknik normalisasi dapat mengubah hasil ranking.
3	Proses perhitungan sederhana, efisien, dan mudah dipahami.	Kurang efektif menangani data yang mengandung ketidakpastian atau bersifat fuzzy.
4	Setiap tahap perhitungan dapat ditelusuri sehingga memudahkan validasi hasil.	Mengasumsikan setiap kriteria bersifat independen, padahal tidak selalu demikian.

Tabel 4.2 Kelebihan dan Kekurangan Metode SAW

5. Dataset

Berisi sumber dataset, jumlah data, atribut atau kriteria, dan contoh tabel.

Berisi:

- a. Sumber dataset (manual, survei, internet, dll)

Dataset diperoleh secara manual dari dokumen Kurikulum Program Studi Informatika Fakultas Ilmu Komputer Universitas Sriwijaya Tahun 2021, yang memuat daftar mata kuliah pilihan beserta informasi SKS dan prasyarat.

- b. Jumlah data

1. Jumlah Total Baris (Rekor/Mata Kuliah): 117 baris
2. Jumlah Total Kolom (Atribut/Kriteria): 11 kolom

- c. Penjelasan setiap atribut atau kriteria

Berikut adalah deskripsi dari 11 kolom (atribut) yang terdapat dalam dataset Kurikulum 2021:

No	Nama Atribut (Kolom)	Penjelasan/Deskripsi	Tipe Data
1	NO	Nomor urut mata kuliah dalam daftar.	Numerik (Integer)
2	KELOMPOK	Kategori/kelompok mata kuliah.	Kategorikal/String

3	KODE M.K	Kode unik identifikasi mata kuliah.	Kategorikal/String
4	NAMA MATA KULIAH	Nama lengkap mata kuliah.	Kategorikal/String
5	SKS	Jumlah Satuan Kredit Semester (SKS) mata kuliah tersebut.	Numerik (Integer)
6	SEMESTER	Semester wajib mata kuliah tersebut.	Numerik (Integer)
7	PRASYARAT KODE	Kode mata kuliah yang harus diambil sebelumnya (prasyarat).	Kategorikal/String
8	PRASYARAT M.K	Nama mata kuliah prasyarat.	Kategorikal/String
9	CPL M.K	Capaian Pembelajaran Lulusan yang dibebankan pada mata kuliah ini.	Kategorikal/String
10	ASAL PRODI	Program Studi Asal (umumnya 'TI' atau 'IF').	Kategorikal/String
11	KETERANGAN	Informasi tambahan atau detail khusus mata kuliah.	Kategorikal/String

Tabel 5.1 Atribut atau Kriteria Dataset

d. Format dataset

Tabel Mata Kuliah Wajib Semester 1 (Tabel Contoh)

No	Kode MK	Nama Mata kuliah	SKS	Prasyarat
1	FIK1101	Pengantar Algoritma dan Struktur Data	3	-
2	FIK1102	Pemrograman Komputer	3	-
3	FIK1103	Praktikum Pemrograman Komputer	1	-
4	FTI1101	Pengantar Teknologi Informasi	2	-
5	FTI1102	Matriks dan Vektor	3	-
6	FTI1103	Kalkulus	3	-
7	UNI1001	Agama	2	-
8	UNI1002	Pancasila	2	-
Total			19	

Tabel 5.2 Mata Kuliah Wajib Semester 1

e. Konversi data

Keterangan:

1. Prasyarat Binary

- Ada prasyarat: Skor 1
- Tidak ada prasyarat: Skor 0

No	Mata Kuliah Pilihan	SKS	SKS Score	Prasyarat	Prasyarat Binary
1	Sistem Pendukung Keputusan	3	3	Kecerdasan Buatan	1
2	Metode Numerik	3	3	Praktikum Pemrograman Komputer	1
3	Pengenalan Pola	3	3	Probabilitas dan Statistika	1
4	Animasi dan Multimedia	3	3	Pengolahan Citra	1
5	Pemrograman Visual	3	3	Pemrograman Komputer Lanjut	1
6	Manajemen Basis Data	3	3	Basis Data	1
7	Pemrograman Komputasi Bergerak	3	3	Pengembangan Perangkat Lunak Berorientasi Obyek	1
8	Grafika Komputer	3	3	Matriks dan Vektor	1
9	Sistem Informasi	3	3	Pengantar Teknologi Informasi	1
10	Keamanan Jaringan Komputer	3	3	Jaringan Komputer	1
11	Pembelajaran Mesin	3	3	Probabilitas dan Statistika	1
12	Data Science	3	3	Probabilitas dan Statistika	1
13	Komputasi Berkinerja Tinggi	3	3	-	0
14	Logika Samar	3	3	Kecerdasan Buatan	1

Tabel 5.3 Prasyarat Mata Kuliah Pilihan

6. Diagram Tahapan Metode

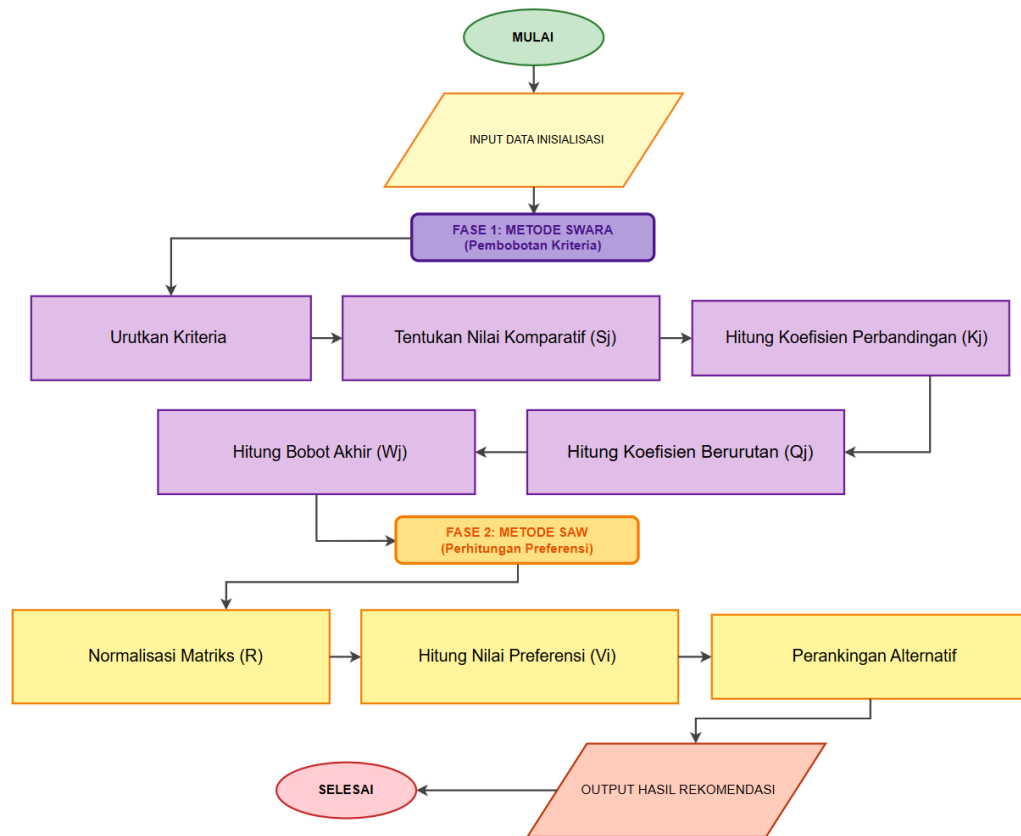


Diagram 6.1 flowchart Metode SWARA - SAW

Proses penentuan rekomendasi mata kuliah pilihan dimulai dengan inisialisasi data yaitu memasukkan alternatif (A1 – A6), kriteria penilaian (C1 – C5) beserta tipenya, matriks keputusan awal, dan nilai rata-rata penilaian pakar untuk setiap kriteria. Data ini menjadi dasar seluruh perhitungan selanjutnya.

Setelah data siap, sistem masuk ke Fase 1: *SWARA*, yaitu proses pembobotan kriteria. Tahap ini diawali dengan mengurutkan kriteria berdasarkan nilai rata - rata tertinggi ke terendah. Urutan ini penting karena *SWARA* menghitung bobot secara berurutan. Selanjutnya ditentukan *nilai komparatif* (S_j), yaitu seberapa besar penurunan kepentingan setiap kriteria dibandingkan kriteria sebelumnya. Kriteria pertama memiliki $S_j = 0$.

Nilai S_j kemudian digunakan untuk menghitung *koefisien perbandingan* (K_j) dengan rumus $K_j = S_j + 1$. Setelah itu dihitung *koefisien berurutan* (Q_j) yang menggambarkan bobot sementara sebelum normalisasi. Nilai $Q_1 = 1$, sedangkan Q_j lainnya diperoleh dari $Q_j = \frac{Q_j - 1}{K_j}$. Seluruh nilai Q_j kemudian dinormalisasi sehingga menghasilkan *bobot*

akhir (W_j) yang totalnya selalu bernilai 1. Bobot ini menjadi input utama pada proses SAW.

Tahapan berikutnya adalah Fase 2: *SAW*, yang digunakan untuk menentukan nilai preferensi setiap alternatif. Pertama, dilakukan *normalisasi matriks keputusan (R)* agar seluruh nilai berada pada skala yang sama. Normalisasi disesuaikan dengan tipe kriteria *benefit* dihitung menggunakan $r_{ij} = \frac{x_{ij}}{\max_{x_{ij}}}$, sedangkan *cost* menggunakan $r_{ij} = \frac{\min_{x_{ij}}}{x_{ij}}$.

Setelah data ternormalisasi, sistem menghitung *nilai preferensi* (V_i) dengan mengalikan setiap nilai normalisasi dengan bobot W_j , lalu menjumlahkan seluruh hasilnya. Nilai V_i menunjukkan tingkat kelayakan suatu alternatif berdasarkan seluruh kriteria.

Tahap terakhir adalah perangkingan alternatif, yaitu mengurutkan alternatif dari nilai V_i tertinggi hingga terendah. Alternatif dengan V_i terbesar menjadi rekomendasi terbaik, dan hasil ini ditampilkan sebagai *output* akhir sistem.

7. Penjelasan Perhitungan Menggunakan Excel

7.1 Sheet Dataset (Matriks X & Jenis Kriteria)

Sheet pertama berisi seluruh data dasar yang diperlukan untuk perhitungan, meliputi definisi kriteria, alternatif, nilai rating, dan tipe kriteria.

Kriteria	
Prasyarat Akademik	C1
Bidang Minat / Spesialisasi	C2
Kesinambungan Kurikulum	C3
Relevansi terhadap Dunia Kerja dan Industri	C4
Tingkat Kesulitan Materi	C5

Tabel 7.1 Daftar Kriteria

Alternatif

Pembelajaran Mesin	A1
Data Science	A2
Pengenalan Pola	A3
Manajemen Basis Data	A4
Keamanan Jaringan Komputer	A5
Animasi dan Multimedia	A6

Tabel 7.2 Daftar Alternatif

Pengurutan kriteria dilakukan berdasarkan nilai t_j (tingkat kepentingan) dari tertinggi ke terendah. Nilai t_j diperoleh dari diskusi kelompok mengenai prioritas setiap kriteria.

Kriteria	t_j	Urutan (j)
Relevansi terhadap Dunia Kerja dan Industri (C4)	5	1
Bidang Minat / Spesialisasi (C2)	4,5	2
Prasyarat Akademik (C1)	4	3
Kesinambungan Kurikulum (C3)	3,5	4
Tingkat Kesulitan Materi (C5)	3	5

Tabel 7.3 Nilai Rata-Rata dan Pengurutan Kriteria

Setiap kriteria diklasifikasikan sebagai Benefit atau Cost sesuai dengan karakteristiknya.

Kriteria (Cj)	Tipe
Prasyarat Akademik (C1)	Cost (Semakin tinggi prasyarat, semakin 'mahal' usaha)
Bidang Minat / Spesialisasi (C2)	Benefit (Semakin tinggi rating, semakin sesuai minat)
Kesinambungan Kurikulum (C3)	Benefit (Semakin tinggi rating, semakin bagus kelanjutannya)
Relevansi terhadap Dunia Kerja dan Industri (C4)	Benefit (Semakin tinggi rating, semakin relevan)
Tingkat Kesulitan Materi (C5)	Cost (Semakin tinggi rating, semakin sulit)

Tabel 7.4 Tipe Kriteria

Matriks ini berisi nilai rating awal setiap alternatif terhadap masing-masing kriteria, dengan skala 1-5.

Alternatif (Ai)	C1 (Cost)	C2 (Benefit)	C3 (Benefit)	C4 (Benefit)	C5 (Cost)
Pembelajaran Mesin (A1)	4	5	5	5	5
Data Science (A2)	3	4	4	5	4
Pengenalan Pola (A3)	4	4	3	4	3
Manajemen Basis Data (A4)	2	3	5	4	2
Keamanan Jaringan Komputer (A5)	3	5	4	4	3
Animasi dan Multimedia (A6)	1	3	3	3	2
Nilai Maks (Max)	4	5	5	5	5
Nilai Min (Min)	1	3	3	3	2

Tabel 7.5 Matriks Keputusan Awal (X)

7.2 Sheet SWARA (Sj, Kj, Qj, Wj)

Sheet kedua menampilkan proses perhitungan bobot kriteria menggunakan metode SWARA secara bertahap.

Urutan (j)	Kriteria	Nilai Komparatif (Sj)	Koefisien Perbandingan (Kj)	Koefisien Berurutan (Qj)	Bobot Akhir (Wj)
1	C4 (Relevansi)	0	1	1	0,2879
2	C2 (Minat)	0,2	1,2	0,8333	0,2399
3	C1 (Prasyarat)	0,3	1,3	0,6410	0,1845
4	C3 (Kesesuaian)	0,1	1,1	0,5827	0,1677
5	C5 (Kesulitan)	0,4	1,4	0,4162	0,1198
Total				0,4733	1

Tabel 7.6 Hasil Perhitungan SWARA

Penjelasan perhitungan ini dimulai dari penentuan *Nilai Komparatif* (Sj), yang menunjukkan besarnya penurunan tingkat kepentingan suatu kriteria dibandingkan kriteria

sebelumnya. Pada kriteria pertama ($j = 1$), nilai S_j selalu ditetapkan sebesar 0 karena tidak ada perbandingan awal. Nilai S_j kemudian digunakan untuk menghitung dengan *Koefisien Perbandingan* (K_j) rumus $K_j = S_j + 1$, sehingga untuk kriteria pertama nilai K_j menjadi 1. Selanjutnya, *Koefisien Berurutan* (Q_j) dihitung menggunakan rumus $Q_j = \frac{Q_j - 1}{K_j}$, dan untuk $j = 1$ nilai Q_j ditetapkan sebesar 1 sebagai dasar perhitungan. Tahap terakhir adalah memperoleh *Bobot Akhir* (W_j) melalui proses normalisasi menggunakan rumus $W_j = \frac{Q_j}{\sum Q_j}$, sehingga seluruh bobot kriteria bernilai total 1 dan dapat digunakan sebagai bobot keputusan yang proporsional.

7.3 Sheet Normalisasi SAW

Sheet ketiga menampilkan hasil normalisasi matriks keputusan menggunakan metode SAW.

Alternatif (Ai)	C1 (Cost)	C2 (Benefit)	C3 (Benefit)	C4 (Benefit)	C5 (Cost)
Pembelajaran Mesin (A1)	0,25	1	1	1	0,4
Data Science (A2)	0,33	0,8	0,8	1	0,5
Pengenalan Pola (A3)	0,25	0,8	0,6	0,8	0,67
Manajemen Basis Data (A4)	0,5	0,6	1	0,8	1
Keamanan Jaringan Komputer (A5)	0,33	1	0,8	0,8	0,67
Animasi dan Multimedia (A6)	1	0,6	0,6	0,6	1

Tabel 7.7 Matriks Normalisasi ®

Proses normalisasi dilakukan untuk menyetarakan skala setiap kriteria sebelum memasuki tahap perhitungan selanjutnya. Pada kriteria bertipe *benefit*, nilai normalisasi dihitung menggunakan rumus $r_{ij} = \frac{x_{ij}}{\max_{x_{ij}}}$, sehingga semakin besar nilai suatu alternatif, semakin baik hasil normalisasinya. Sebaliknya, untuk kriteria bertipe *cost*, digunakan rumus $r_{ij} = \frac{\min_{x_{ij}}}{x_{ij}}$, sehingga nilai yang lebih kecil akan menghasilkan nilai normalisasi yang lebih tinggi. Sebagai contoh, perhitungan normalisasi untuk alternatif A1 pada kriteria C1 yang

bertipe cost adalah $r_{11} = \frac{\min_{C1}}{x_{11}} = \frac{1}{4} = 0,25$. Sementara itu, pada kriteria C2 yang bertipe benefit, nilai normalisasi untuk alternatif A1 dihitung dengan rumus $r_{12} = \frac{x_{12}}{\max_{C2}} = \frac{5}{5} = 1,00$.

7.4 Sheet Nilai Akhir (Vi) dan Peringkat

Sheet keempat menampilkan hasil akhir perhitungan nilai preferensi dan peringkat alternatif.

Alternatif (Ai)	Nilai Preferensi (Vi)	C2 (Benefit)
Pembelajaran Mesin (A1)	0,7896	1
Data Science (A2)	0,7355	4
Pengenalan Pola (A3)	0,6489	6
Manajemen Basis Data (A4)	0,7541	2
Keamanan Jaringan Komputer (A5)	0,7458	3
Animasi dan Multimedia (A6)	0,7217	5

Tabel 7.8 Bobot Kriteria (Wj)

Kriteria (Cj)	C1	C2	C3	C4	C5
Bobot (Wj)	0,1845	0,2399	0,1677	0,2879	0,1198

Tabel 7.9 Nilai Preferensi dan Peringkat Akhir

Nilai preferensi setiap alternatif dihitung menggunakan rumus $V_i = \sum (W_j \times r_{ij})$, yaitu menjumlahkan hasil perkalian antara *bobot setiap kriteria* (Wj) dengan nilai normalisasi alternatif pada kriteria tersebut (rij). Perhitungan ini menghasilkan skor akhir yang mencerminkan performa keseluruhan alternatif berdasarkan seluruh kriteria. Sebagai contoh, nilai preferensi untuk alternatif A1 dihitung melalui operasi $V_1 = (0,1845 \times 0,25) + (0,2399 \times 1,00) + (0,1677 \times 1,00) + (0,2879 \times 1,00) + (0,1198 \times 0,40)$, yang menghasilkan $0,0461 + 0,2399 + 0,1677 + 0,2879 + 0,0479$. Jumlah keseluruhan dari komponen tersebut adalah $V_1 = 0,7896$, yang menjadi nilai preferensi akhir untuk alternatif A1.

Berdasarkan perhitungan di atas, mata kuliah Pembelajaran Mesin (A1) memperoleh nilai preferensi tertinggi (0,7896), sehingga menjadi rekomendasi terbaik untuk diambil pada

semester 5. Hal ini didukung oleh nilai maksimal pada hampir seluruh kriteria benefit (C2, C3, C4) dan nilai yang relatif baik pada kriteria cost (C1, C5). Urutan rekomendasi selanjutnya adalah Manajemen Basis Data, Keamanan Jaringan Komputer, Data Science, Animasi dan Multimedia, dan terakhir Pengenalan Pola.

8. Implementasi Program

Implementasi sistem rekomendasi mata kuliah pilihan ini dibangun sebagai aplikasi berbasis web yang mengintegrasikan backend menggunakan *FastAPI* (Python) dan frontend menggunakan *React.js*. Metode *Stepwise Weight Assessment Ratio Analysis* (SWARA) dan *Simple Additive Weighting* (SAW) diimplementasikan pada sisi backend untuk menjamin konsistensi dan objektivitas perhitungan rekomendasi.


8.1. Arsitektur Aplikasi

Aplikasi ini mengadopsi arsitektur *Model - View - Controller* (MVC) secara konseptual, dimana logika bisnis dan perhitungan MCDM (SWARA - SAW) dipisahkan dari antarmuka pengguna:

- a. Backend (*FastAPI*): Bertanggung jawab untuk menyediakan API endpoints untuk mendapatkan data kriteria, alternatif, dan melakukan perhitungan rekomendasi berdasarkan bobot masukan pengguna.
- b. Frontend (*React.js*): Bertanggung jawab untuk antarmuka pengguna, mengelola state aplikasi, menerima input bobot, dan menampilkan hasil rekomendasi.

8.2. Tampilan Aplikasi: Input Bobot Kriteria

Tampilan awal aplikasi memandu pengguna untuk memasukkan bobot kepentingan untuk setiap kriteria. Halaman ini dibagi menjadi tiga bagian utama: panduan pengisian, daftar alternatif, dan form input bobot.


Sistem Rekomendasi Mata Kuliah Pilihan Semester 5
Metode SWARA untuk Pemilihan Mata Kuliah Pilihan Semester 5 Fakultas Ilmu Komputer - Teknik Informatika

1 Input Bobot
2 Hasil Rekomendasi

Cara Pengisian Bobot

Berikan nilai 1-5 untuk setiap kriteria (5 = sangat penting, 1 = kurang penting)

Setiap nilai harus unik (tidak boleh ada yang sama)

Nilai tertinggi menunjukkan kriteria yang paling penting bagi Anda

Daftar Mata Kuliah Pilihan

Pembelajaran Mesin A1	Data Science A2
Pengenalan Pola A3	Manajemen Basis Data A4
Keamanan Jaringan Komputer A5	Animasi dan Multimedia A6

Tentukan Tingkat Kepentingan Kriteria

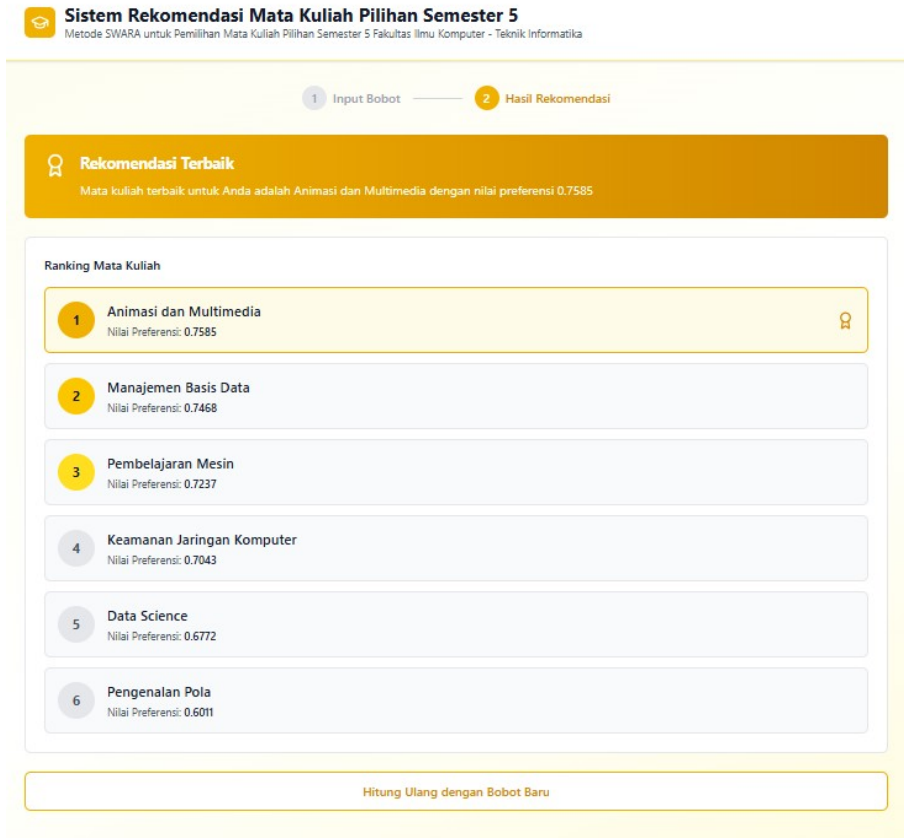
Prasyarat Akademik <small>Tipe: Cost</small>	5
Bidang Minat / Spesialisasi <small>Tipe: Benefit</small>	4
Kesinambungan Kurikulum <small>Tipe: Benefit</small>	3
Relevansi terhadap Dunia Kerja dan Industri <small>Tipe: Benefit</small>	2
Tingkat Kesulitan Materi <small>Tipe: Cost</small>	1

Hitung Rekomendasi

Gambar 8.1 - Halaman Input Bobot Kriteria

8.3. Tampilan Aplikasi: Hasil Rekomendasi

Setelah perhitungan berhasil, aplikasi akan beralih ke tampilan hasil rekomendasi (komponen Rekomendasi) untuk menampilkan peringkat akhir mata kuliah .



Gambar 8.2 - Halaman Hasil Rekomendasi

9. Analisis Hasil

Proses pengolahan data menggunakan kombinasi metode SWARA dan SAW menghasilkan bobot kriteria serta peringkat akhir alternatif mata kuliah pilihan. Tahap pertama dilakukan dengan metode SWARA untuk memperoleh bobot kriteria secara objektif berdasarkan tingkat prioritas yang ditentukan oleh responden. Berdasarkan nilai rata-rata penilaian pakar, kriteria Relevansi terhadap Dunia Kerja dan Industri (C4) menjadi kriteria paling penting dengan bobot akhir 0,2879, diikuti oleh Bidang Minat/Spesialisasi (C2) dengan bobot 0,2399. Sementara itu, Tingkat Kesulitan Materi (C5) memperoleh bobot paling rendah yaitu 0,1198. Hal ini menunjukkan bahwa pemilihan mata kuliah pilihan lebih dipengaruhi oleh manfaatnya dalam karier dan kesesuaian minat mahasiswa, sedangkan tingkat kesulitan bukan menjadi faktor dominan. Temuan ini sejalan dengan studi terdahulu yang menyatakan bahwa relevansi industri dan minat merupakan faktor utama dalam pengambilan keputusan akademik (Winata & Gustientiedina, 2025; Situmorang et al., 2025).

Tahap berikutnya menggunakan metode SAW untuk menghitung nilai preferensi alternatif berdasarkan bobot kriteria yang telah diperoleh. Proses normalisasi dilakukan sesuai

tipe kriteria (benefit atau cost), kemudian setiap nilai dikalikan dengan bobot SWARA. Hasil perhitungan menunjukkan bahwa alternatif A1 – Pembelajaran Mesin memiliki nilai preferensi tertinggi yaitu 0,7896, sehingga menjadi alternatif paling direkomendasikan. Alternatif berikutnya adalah A4 – Manajemen Basis Data (0,7541), diikuti A5 – Keamanan Jaringan Komputer (0,7458), dan A2 – Data Science (0,7355). Alternatif A3 – Pengenalan Pola dan A6 – Animasi dan Multimedia berada pada peringkat terendah.

Jika diperhatikan, tingginya nilai A1 disebabkan oleh performa maksimal pada hampir seluruh kriteria benefit (C2 – C4) serta nilai yang relatif baik pada kriteria cost (C1 dan C5). Hal ini mencerminkan bahwa mata kuliah Pembelajaran Mesin dianggap paling relevan dengan industri, sangat sesuai dengan minat mahasiswa informatika, serta memiliki kesinambungan kurikulum yang kuat. Hasil ini konsisten dengan penelitian Taherdoost (2023) yang menyatakan bahwa SAW efektif memberikan rekomendasi yang stabil ketika bobot kriteria didominasi faktor benefit.

Secara keseluruhan, hasil analisis menunjukkan bahwa metode SWARA–SAW mampu memberikan pemeringkatan alternatif secara terstruktur, transparan, dan konsisten dengan dasar teori pengambilan keputusan multikriteria.

10. Kesimpulan dan Saran

10.1 Kesimpulan

Berdasarkan hasil penelitian dan perhitungan menggunakan kombinasi metode SWARA dan SAW, dapat disimpulkan bahwa proses penentuan bobot dan pemeringkatan alternatif berjalan secara sistematis dan konsisten. Metode SWARA mampu menetapkan bobot kriteria berdasarkan tingkat prioritas responden, di mana Relevansi terhadap Dunia Kerja dan Industri (C4) menjadi kriteria paling berpengaruh dengan bobot 0,2879, sedangkan Tingkat Kesulitan Materi (C5) memperoleh bobot terendah yaitu 0,1198. Bobot ini kemudian digunakan dalam metode SAW untuk menghitung nilai preferensi setiap alternatif melalui proses normalisasi dan pembobotan. Hasil perhitungan menunjukkan bahwa Pembelajaran Mesin (A1) memiliki nilai preferensi tertinggi, yaitu 0,7896, sehingga menjadi mata kuliah pilihan yang paling direkomendasikan. Urutan rekomendasi lengkap berdasarkan nilai preferensi adalah A1, A4, A5, A2, A6, dan A3. Secara keseluruhan, kombinasi metode SWARA–SAW terbukti

efektif dalam mendukung proses pemilihan mata kuliah pilihan secara lebih terstruktur, objektif, dan transparan (Anonymous Student, 2023; Taherdoost, 2023).

10. 2 Saran

Penelitian ini masih dapat dikembangkan lebih lanjut agar hasil rekomendasinya semakin akurat dan memiliki cakupan yang lebih luas. Salah satu upaya yang dapat dilakukan adalah menambah jumlah responden atau pakar pada tahap SWARA, sehingga bobot kriteria yang dihasilkan menjadi lebih representatif dan mengurangi potensi bias. Selain itu, penambahan kriteria yang lebih spesifik—seperti metode pengajaran, ketersediaan dosen, atau beban tugas—dapat memberikan gambaran yang lebih komprehensif mengenai pertimbangan mahasiswa dalam memilih mata kuliah pilihan. Sistem ini juga berpotensi dikembangkan menjadi aplikasi berbasis web atau mobile yang terintegrasi dengan data akademik kampus, sehingga proses rekomendasi dapat diberikan secara otomatis dan real time. Untuk memastikan keandalan model, penggunaan metode MCDM lain seperti AHP, TOPSIS, atau MOORA sebagai pembanding dapat membantu memvalidasi konsistensi hasil yang diperoleh. Selain itu, sistem dapat diuji pada konteks pengambilan keputusan lain, seperti pemilihan konsentrasi, rekomendasi karier, atau evaluasi mata kuliah, sehingga model menjadi lebih fleksibel dan bermanfaat bagi berbagai kebutuhan pengguna.

11. Daftar Pustaka

- Fitriani, R., & Fadhilah, N. (2020). Sistem pendukung keputusan pemilihan mata kuliah menggunakan metode SAW. *Jurnal Informatika dan Sistem Informasi*, 7(2), 85–94.
- Ramadhani, F., & Satria, A. (2024). Implementasi Simple Additive Weighting (SAW) dan Stepwise Weight Assessment Ratio Analysis (SWARA) untuk memberikan alternatif terbaik dalam pemilihan jenis laptop. *Wahana Jurnal Pengabdian kepada Masyarakat*, 3(1), 32–39.
- Anonymous Student. (2022/2023). Analisis Metode SWARA: Langkah-Langkah dan Contoh Penerapan (Step-wise Weight Assessment Ratio Analysis Method – SWARA Method). Universitas Ahmad Dahlan.
- Taherdoost, H. (2023). *E-Research: Applying research methodologies and frameworks*. Emerald Publishing Limited.

Situmorang, R. A., Abdillah, R., Zulkarnaen, I., & Handoko, D. (2025). Implementasi Sistem Pendukung Keputusan Metode SWARA untuk Pemilihan Laptop Low-End Terbaik 2024 bagi Programmer. *JUKTISI Jurnal Komputer Teknologi Informasi Sistem Komputer*, 3(3), 885–893.

Sriyasa, I. W. (2023). Kombinasi Metode SWARA dan Simple Additive Weighting (SAW) Pemilihan Tempat Kursus. *Journal of Artificial Intelligence and Technology Information (JAITI)*, 1(4), 146-153.

Winata, A. L., & Gustientiedina. (2025). Penerapan Metode SWARA dan MOORA dalam Menentukan Insentif Karyawan. *Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi*, 7(1), 40–45.

12. Lampiran

Lampiran 1. Link Video Demo Website

<https://youtu.be/2HTi-Ce5WQc?si=MvuqdmXNv3oPVgz7>

Lampiran 2. Link Data Alternatif Mata Kuliah Pilihan Fakultas Ilmu Komputer Jurusan Teknik Informatika Universitas Sriwijaya

<https://docs.google.com/spreadsheets/d/1-Oma9gYfgkGVOUb-Sw2SHFbghPEehaoW/edit?usp=sharing&ouid=101562715900113381810&rtpof=true&sd=true>

Lampiran 3. Kode Program Metode SWARA dan SAW

```
%pip install numpy pandas
import numpy as np
import pandas as pd
```

```
# 1. DATA INISIALISASI
```

```
# Pembobotan Kriteria
```

```
# a. Alternatif (Mata Kuliah Pilihan)
```

```
alternatif = {
    'A1': 'Pembelajaran Mesin',
    'A2': 'Data Science',
    'A3': 'Pengenalan Pola',
    'A4': 'Manajemen Basis Data',
```

```

'A5': 'Keamanan Jaringan Komputer',
'A6': 'Animasi dan Multimedia'
}

# b. Data Kriteria, Tipe, Urutan (j) dan Nilai Komparatif (Sj)
# Urutan kriteria berdasarkan Nilai Rata-Rata (tj)
data_swara_input = [
    {'Kode': 'C4', 'Nama': 'Relevansi terhadap Dunia Kerja dan Industri', 'Tipe': 1, 'tj': 5.0, 'Sj': 0.0}, # Benefit
    {'Kode': 'C2', 'Nama': 'Bidang Minat / Spesialisasi', 'Tipe': 1, 'tj': 4.5, 'Sj': 0.2}, # Benefit
    {'Kode': 'C1', 'Nama': 'Prasyarat Akademik', 'Tipe': 0, 'tj': 4.0, 'Sj': 0.3}, # Cost
    {'Kode': 'C3', 'Nama': 'Kesesinambungan Kurikulum', 'Tipe': 1, 'tj': 3.5, 'Sj': 0.1}, # Benefit
    {'Kode': 'C5', 'Nama': 'Tingkat Kesulitan Materi', 'Tipe': 0, 'tj': 3.0, 'Sj': 0.4}, # Cost
]
df_swara = pd.DataFrame(data_swara_input).set_index('Kode')
kriteria_list = df_swara.index.tolist()
tipe_kriteria = df_swara['Tipe'].to_dict()

# c. Matriks Keputusan Awal (X)
# Urutan kolom Matriks X HARUS C1, C2, C3, C4, C5 untuk kebutuhan Normalisasi
data_x = [
    [4, 5, 5, 5, 5], # A1: Pembelajaran Mesin
    [3, 4, 4, 5, 4], # A2: Data Science
    [4, 4, 3, 4, 3], # A3: Pengenalan Pola
    [2, 3, 5, 4, 2], # A4: Manajemen Basis Data
    [3, 5, 4, 4, 3], # A5: Keamanan Jaringan Komputer
    [1, 3, 3, 3, 2] # A6: Animasi dan Multimedia
]

# Membuat DataFrame Matriks Keputusan (X) dengan urutan kolom C1, C2, C3, C4, C5
df_X_input = pd.DataFrame(data_x, index=alternatif.values(), columns=['C1', 'C2', 'C3', 'C4', 'C5'])

print("## Matriks Keputusan Awal (X):")
print(df_X_input)

2. PERHITUNGAN METODE SWARA (Bobot Wj)

# 2.1 Hitung Koefisien Perbandingan (Kj)
# Rumus:  $K_j = S_j + 1$ . Jika  $j=1$ ,  $K_j = 1$ 
Kj = []

```

```

for j, Sj in enumerate(df_swara['Sj']):
    # j adalah index (0, 1, 2, ...), j+1 adalah Urutan (1, 2, 3, ...)
    if (j + 1) == 1:
        Kj.append(1.0) # Urutan 1, Kj = 1
    else:
        Kj.append(Sj + 1)
df_swara['Kj (Koefisien Perbandingan)'] = Kj

# 2.2 Hitung Koefisien Berurutan (Qj)
# Rumus:  $Q_j = Q_{j-1} / K_j$ . Jika  $j=1$ ,  $Q_j = 1$ 
Qj = []
for j, Kj_val in enumerate(df_swara['Kj (Koefisien Perbandingan)']):
    if (j + 1) == 1:
        Qj.append(1.0) # Urutan 1, Qj = 1
    else:
        #  $Q_j = Q_{j-1} / K_j$ 
        Qj.append(Qj[-1] / Kj_val) # Qj[-1] adalah Qj-1
df_swara['Qj (Koefisien Berurutan)'] = Qj

# 2.3 Hitung Bobot Akhir (Wj)
# Rumus:  $W_j = Q_j / \text{Sigma}(Q_j)$ 
Sigma_Qj = sum(Qj)
Wj = [q / Sigma_Qj for q in Qj]
df_swara['Wj (Bobot Akhir)'] = Wj

# Tampilkan hasil perhitungan SWARA (untuk verifikasi dengan Gambar b)
pd.options.display.float_format = '{:.10f}'.format
print("## Hasil Perhitungan SWARA (Bobot Kriteria):")
df_swara[['Sj', 'Kj (Koefisien Perbandingan)', 'Qj (Koefisien Berurutan)', 'Wj (Bobot Akhir)']]

# 3. PERHITUNGAN METODE SAW (NORMALISASI DAN PREFERENSI)

# a. Normalisasi Matriks (R)
df_R = df_X_input.copy()

max_vals = df_X_input.max()
min_vals = df_X_input.min()

for j in df_X_input.columns:
    tipe = tipe_kriteria[j] # Ambil Tipe (1=Benefit, 0=Cost)

    if tipe == 1:
        # Benefit:  $r_{ij} = x_{ij} / \max(x_{ij})$ 

```



```

        df_R[j] = df_X_input[j] / max_vals[j]
    else:
        # Cost: rij = min(xij) / xij
        df_R[j] = min_vals[j] / df_X_input[j]

print("Matriks Normalisasi (R):")
df_R
# b. Perhitungan Nilai Preferensi (Vi) dan Perankingan
# Rumus: Vi = Sigma (Wj * rij)
R_matrix = df_R.values
W_vector = bobot_swara # Menggunakan bobot hasil SWARA (C1, C2, C3, C4, C5)

# Hitung Nilai Preferensi (Vi)
V_scores = R_matrix.dot(W_vector)

# Membuat DataFrame Hasil Akhir
results = pd.DataFrame({
    'Alternatif': alternatif.values(),
    'Nilai Preferensi (Vi)': V_scores
})

# Lakukan Perankingan (Peringkat 1 = Vi tertinggi)
results['Peringkat'] = results['Nilai Preferensi (Vi)'].rank(ascending=False,
method='min').astype(int)

# Urutkan dan Tampilkan Hasil
df_final = results.sort_values(by='Nilai Preferensi (Vi)',
ascending=False).reset_index(drop=True)

print("## Hasil Akhir Perankingan (SWARA-SAW):")
df_final

print("Kesimpulan:")
print(f'Keputusan Terbaik terbaik untuk pemilihan mata kuliah semester 5 di Fakultas
Ilmu Komputer Teknik Informatika adalah {df_final.iloc[0]['Alternatif']}")

```

Lampiran 4. Kode Program Backend API (Library FastAPI)

```

from fastapi import FastAPI, HTTPException
from fastapi.middleware.cors import CORSMiddleware
from pydantic import BaseModel, validator
from typing import Dict, List
import numpy as np

```

```

import pandas as pd

app = FastAPI(title="SWARA-SAW Course Recommendation API")

# CORS middleware untuk akses dari frontend
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

# DATA STATIS
ALTERNATIF = {
    'A1': 'Pembelajaran Mesin',
    'A2': 'Data Science',
    'A3': 'Pengenalan Pola',
    'A4': 'Manajemen Basis Data',
    'A5': 'Keamanan Jaringan Komputer',
    'A6': 'Animasi dan Multimedia'
}

KRITERIA_INFO = {
    'C1': {'nama': 'Prasyarat Akademik', 'tipe': 0}, # Cost
    'C2': {'nama': 'Bidang Minat / Spesialisasi', 'tipe': 1}, # Benefit
    'C3': {'nama': 'Kesesuaian Kurikulum', 'tipe': 1}, # Benefit
    'C4': {'nama': 'Relevansi terhadap Dunia Kerja dan Industri', 'tipe': 1}, # Benefit
    'C5': {'nama': 'Tingkat Kesulitan Materi', 'tipe': 0} # Cost
}

# Matriks Keputusan (Data Rating) - Tetap
MATRIKS_X = [
    [4, 5, 5, 5, 5], # A1: Pembelajaran Mesin
    [3, 4, 4, 5, 4], # A2: Data Science
    [4, 4, 3, 4, 3], # A3: Pengenalan Pola
    [2, 3, 5, 4, 2], # A4: Manajemen Basis Data
    [3, 5, 4, 4, 3], # A5: Keamanan Jaringan Komputer
    [1, 3, 3, 3, 2] # A6: Animasi dan Multimedia
]

# Nilai Sj default (bisa disesuaikan)
DEFAULT_SJ = {
    'C1': 0.3,

```

```

'C2': 0.2,
'C3': 0.1,
'C4': 0.0, # Kriteria paling penting
'C5': 0.4
}

```

Pydantic Models

```

class KriteriaBobotInput(BaseModel):

```

```

    C1: float

```

```

    C2: float

```

```

    C3: float

```

```

    C4: float

```

```

    C5: float

```

```

    @validator('C1', 'C2', 'C3', 'C4', 'C5')

```

```

    def validate_range(cls, v):

```

```

        if not (1 <= v <= 5):

```

```

            raise ValueError('Nilai tj harus antara 1 dan 5')

```

```

        return v

```

```

    @validator('C5')

```

```

    def validate_unique(cls, v, values):

```

```

        all_values = list(values.values()) + [v]

```

```

        if len(all_values) != len(set(all_values)):

```

```

            raise ValueError('Nilai tj harus unik (tidak boleh sama)')

```

```

        return v

```

```

class RekomendasiResponse(BaseModel):

```

```

    alternatif: str

```

```

    nilai_preferensi: float

```

```

    peringkat: int

```

```

class HasilResponse(BaseModel):

```

```

    bobot_kriteria: Dict[str, float]

```

```

    urutan_kriteria: List[str]

```

```

    rekomendasi: List[RekomendasiResponse]

```

```

    kesimpulan: str

```

FUNGSI PERHITUNGAN

```

def hitung_swara(tj_dict: Dict[str, float]) -> Dict[str, float]:

```

```

    """

```

```

    Menghitung bobot kriteria menggunakan metode SWARA

```

```

    """

```

```

# Urutkan kriteria berdasarkan tj (tertinggi ke terendah)
sorted_kriteria = sorted(tj_dict.items(), key=lambda x: x[1], reverse=True)

# Siapkan data untuk perhitungan
df_data = []
for i, (kode, tj_val) in enumerate(sorted_kriteria):
    if i == 0:
        sj = 0.0 # Kriteria paling penting
    else:
        # Ambil Sj dari kode sebelumnya (atau gunakan default)
        sj = DEFAULT_SJ.get(kode, 0.1)

    df_data.append({
        'Kode': kode,
        'tj': tj_val,
        'Sj': sj
    })

df_swara = pd.DataFrame(df_data)

# Hitung Kj
Kj = []
for i, row in df_swara.iterrows():
    if i == 0:
        Kj.append(1.0)
    else:
        Kj.append(row['Sj'] + 1)
df_swara['Kj'] = Kj

# Hitung Qj
Qj = []
for i, kj_val in enumerate(df_swara['Kj']):
    if i == 0:
        Qj.append(1.0)
    else:
        Qj.append(Qj[-1] / kj_val)
df_swara['Qj'] = Qj

# Hitung Wj
sigma_qj = sum(Qj)
Wj = [q / sigma_qj for q in Qj]
df_swara['Wj'] = Wj

# Return bobot dalam format dict dengan urutan C1-C5

```

```

    bobot_dict = dict(zip(df_swara['Kode'], df_swara['Wj']))
    return bobot_dict, df_swara['Kode'].tolist()

def hitung_saw(bobot_dict: Dict[str, float]) -> pd.DataFrame:
    """
    Menghitung ranking menggunakan metode SAW
    """
    # Buat DataFrame Matriks X
    df_X = pd.DataFrame(MATRIKS_X,
                        index=ALTERNATIF.values(),
                        columns=['C1', 'C2', 'C3', 'C4', 'C5'])

    # Normalisasi
    df_R = df_X.copy()
    max_vals = df_X.max()
    min_vals = df_X.min()

    for kode in df_X.columns:
        tipe = KRITERIA_INFO[kode]['tipe']
        if tipe == 1: # Benefit
            df_R[kode] = df_X[kode] / max_vals[kode]
        else: # Cost
            df_R[kode] = min_vals[kode] / df_X[kode]

    # Hitung Nilai Preferensi
    bobot_array = np.array([bobot_dict[k] for k in ['C1', 'C2', 'C3', 'C4', 'C5']])
    V_scores = df_R.values.dot(bobot_array)

    # Buat DataFrame hasil
    results = pd.DataFrame({
        'Alternatif': ALTERNATIF.values(),
        'Nilai Preferensi': V_scores
    })

    results['Peringkat'] = results['Nilai Preferensi'].rank(ascending=False,
method='min').astype(int)
    results = results.sort_values(by='Nilai Preferensi',
ascending=False).reset_index(drop=True)

    return results

# API ENDPOINTS

```

```

@app.get("/")
def root():
    return {
        "message": "SWARA-SAW Course Recommendation API",
        "version": "1.0",
        "endpoints": {
            "/kriteria": "GET - Daftar kriteria",
            "/alternatif": "GET - Daftar mata kuliah",
            "/hitung": "POST - Hitung rekomendasi"
        }
    }

@app.get("/kriteria")
def get_kriteria():
    """Mendapatkan daftar kriteria"""
    return {
        "kriteria": [
            {
                "kode": kode,
                "nama": info["nama"],
                "tipe": "Benefit" if info["tipe"] == 1 else "Cost"
            }
            for kode, info in KRITERIA_INFO.items()
        ]
    }

@app.get("/alternatif")
def get_alternatif():
    """Mendapatkan daftar mata kuliah"""
    return {
        "alternatif": [
            {"kode": kode, "nama": nama}
            for kode, nama in ALTERNATIF.items()
        ]
    }

@app.post("/hitung", response_model=HasilResponse)
def hitung_rekomendasi(bobot: KriteriaBobotInput):
    """
    Menghitung rekomendasi mata kuliah berdasarkan bobot kriteria
    """
    try:
        # Konversi input ke dict
        tj_dict = bobot.dict()

```

```

# Hitung SWARA
bobot_kriteria, urutan_kriteria = hitung_swara(tj_dict)

# Hitung SAW
hasil_ranking = hitung_saw(bobot_kriteria)

# Format response
rekomendasi = [
    RekomendasiResponse(
        alternatif=row['Alternatif'],
        nilai_preferensi=round(row['Nilai Preferensi'], 6),
        peringkat=row['Peringkat']
    )
    for _, row in hasil_ranking.iterrows()
]

kesimpulan = f"Mata kuliah terbaik untuk Anda adalah {hasil_ranking.iloc[0]
['Alternatif']} dengan nilai preferensi {hasil_ranking.iloc[0]['Nilai Preferensi']:.4f}"

return HasilResponse(
    bobot_kriteria={k: round(v, 6) for k, v in bobot_kriteria.items()},
    urutan_kriteria=urutan_kriteria,
    rekomendasi=rekomendasi,
    kesimpulan=kesimpulan
)

except Exception as e:
    raise HTTPException(status_code=400, detail=str(e))

if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8000)

```

Lampiran 5. Kode Program Frontend Interface (Library React JS)

```

import { StrictMode } from "react";
import { createRoot } from "react-dom/client";
import "./style/style.css";
import Home from "./page/Home";
createRoot(document.getElementById("root")).render(
    <StrictMode>

```

```

    <Home />
  </StrictMode>
);

import { GraduationCap } from "lucide-react";

const Header = () => {
  return (
    <header className="bg-white border-b border-yellow-200 shadow-sm">
      <div className="max-w-6xl mx-auto px-4 py-6">
        <div className="flex items-center gap-3">
          <div className="bg-yellow-500 p-2 rounded-lg">
            <GraduationCap className="w-6 h-6 text-white" />
          </div>
          <div>
            <h1 className="text-2xl font-bold text-gray-900">
              Sistem Rekomendasi Mata Kuliah Pilihan Semester 5
            </h1>
            <p className="text-sm text-gray-600">
              Metode SWARA untuk Pemilihan Mata Kuliah Pilihan Semester 5
              Fakultas Ilmu Komputer - Teknik Informatika
            </p>
          </div>
        </div>
      </div>
    </header>
  );
};

export default Header;

const Progress = ({ step }) => {
  return (
    <div className="mb-8">
      <div className="flex items-center justify-center gap-4">
        <div
          className={`flex items-center gap-2 ${
            step === 1 ? "text-yellow-600" : "text-gray-400"
          }`}
        >
          <div
            className={`w-8 h-8 rounded-full flex items-center justify-center font-
semibold ${
              step === 1 ? "bg-yellow-500 text-white" : "bg-gray-200"
            }`}
          >

```



```

    }}
  >
  1
</div>
  <span className="font-medium">Input Bobot</span>
</div>
<div className="w-16 h-0.5 bg-gray-300"></div>
<div
  className={`flex items-center gap-2 ${
    step === 2 ? "text-yellow-600" : "text-gray-400"
  }`}
>
  <div
    className={`w-8 h-8 rounded-full flex items-center justify-center font-
semibold ${
      step === 2 ? "bg-yellow-500 text-white" : "bg-gray-200"
    }`}
  >
    2
  </div>
  <span className="font-medium">Hasil Rekomendasi</span>
</div>
</div>
</div>
);
};

export default Progress;

import { TrendingUp } from "lucide-react";

const InfoCard = () => {
  return (
    <div className="bg-yellow-50 border border-yellow-200 rounded-lg p-5">
      <div className="flex items-start gap-3">
        <TrendingUp className="w-5 h-5 text-yellow-600 flex-shrink-0 mt-0.5" />
        <div>
          <h3 className="font-semibold text-gray-900 mb-2">
            Cara Pengisian Bobot
          </h3>
          <ul className="text-sm text-gray-700 space-y-1">
            <li>
              Berikan nilai 1-5 untuk setiap kriteria (5 = sangat penting, 1 =
kurang penting)
            </li>
          </ul>
        </div>
      </div>
    </div>
  );
};

```

```

        </li>
        <li>Setiap nilai harus unik (tidak boleh ada yang sama)</li>
        <li>
            Nilai tertinggi menunjukkan kriteria yang paling penting bagi Anda
        </li>
    </ul>
</div>
</div>
</div>
</div>
);
};

export default InfoCard;

import { BarChart3 } from "lucide-react";

const AlternatifPreview = ({ alternatif }) => {
    return (
        <div className="bg-white rounded-lg border border-gray-200 p-6">
            <h3 className="font-semibold text-gray-900 mb-4 flex items-center gap-2">
                <BarChart3 className="w-5 h-5 text-yellow-600" />
                Daftar Mata Kuliah Pilihan
            </h3>
            <div className="grid grid-cols-2 md:grid-cols-2 gap-3">
                {alternatif.map((alt) => (
                    <div
                        key={alt.kode}
                        className="bg-gray-50 rounded-lg px-4 py-3 border border-gray-200"
                    >
                        <p className="text-sm font-medium text-gray-900">{alt.nama}</p>
                        <p className="text-xs text-gray-500 mt-1">{alt.kode}</p>
                    </div>
                ))}
            </div>
        </div>
    );
};

export default AlternatifPreview;

import { CheckCircle2, Loader2 } from "lucide-react";

const InputBobot = ({
    kriteria,

```

```

    bobot,
    loading,
    handleSubmit,
    handleBobotChange,
  }) => {
    return (
      <div className="bg-white rounded-lg border border-gray-200 p-6">
        <h3 className="font-semibold text-gray-900 mb-5">
          Tentukan Tingkat Kepentingan Kriteria
        </h3>
        <div className="space-y-4">
          {kriteria.map((k) => (
            <div
              key={k.kode}
              className="flex items-center gap-4 p-4 bg-gray-50 rounded-lg border border-
gray-200"
            >
              <div className="flex-1">
                <label className="font-medium text-gray-900 block mb-1">
                  {k.nama}
                </label>
                <p className="text-xs text-gray-500">
                  Tipe: { " " }
                <span
                  className={
                    k.tipe === "Benefit" ? "text-green-600" : "text-blue-600"
                  }
                >
                  {k.tipe}
                </span>
              </p>
            </div>
            <div className="flex items-center gap-3">
              <input
                type="range"
                min="1"
                max="5"
                step="0.5"
                value={bobot[k.kode] || 3}
                onChange={(e) => handleBobotChange(k.kode, e.target.value)}
                className="w-32 accent-yellow-500"
              />
              <input
                type="number"

```

```

        min="1"
        max="5"
        step="0.5"
        value={bobot[k.kode] || 3}
        onChange={(e) => handleBobotChange(k.kode, e.target.value)}
        className="w-16 px-3 py-2 border border-gray-300 rounded-lg text-center
font-semibold text-gray-900 focus:ring-2 focus:ring-yellow-500 focus:border-
transparent"
      />
    </div>
  </div>
  )))}
</div>

```

```

<button
  onClick={handleSubmit}
  disabled={loading}
  className="w-full mt-6 bg-yellow-500 hover:bg-yellow-600 text-white font-
semibold py-3 px-6 rounded-lg transition-colors disabled:bg-gray-300
disabled:cursor-not-allowed flex items-center justify-center gap-2"
>
  {loading ? (
    <
      <Loader2 className="w-5 h-5 animate-spin" />
      Menghitung Rekomendasi...
    </>
  ) : (
    <
      <CheckCircle2 className="w-5 h-5" />
      Hitung Rekomendasi
    </>
  )}
</button>
</div>
);
};

```

```
export default InputBobot;
```

```
import { Award } from "lucide-react";
```

```

const Rekomendasi = ({ hasil, getRankColor, handleReset }) => {
  return (
    <div className="space-y-6">

```

```

    {/ * Kesimpulan */}
    <div className="bg-gradient-to-r from-yellow-500 to-yellow-600 rounded-lg p-6
text-white">
      <div className="flex items-start gap-4">
        <Award className="w-8 h-8 flex-shrink-0" />
        <div>
          <h3 className="font-bold text-xl mb-2">Rekomendasi Terbaik</h3>
          <p className="text-yellow-50">{hasil.kesimpulan}</p>
        </div>
      </div>
    </div>
  </div>

```

```

    {/ * Ranking Mata Kuliah */}
    <div className="bg-white rounded-lg border border-gray-200 p-6">
      <h3 className="font-semibold text-gray-900 mb-4">
        Ranking Mata Kuliah
      </h3>
      <div className="space-y-3">
        {hasil.rekomendasi.map((rek) => (
          <div
            key={rek.alternatif}
            className={`p-4 rounded-lg border-2 ${
              rek.peringkat === 1
                ? "bg-yellow-50 border-yellow-500"
                : "bg-gray-50 border-gray-200"
            }`}
          >
            <div className="flex items-center gap-4">
              <div
                className={`w-12 h-12 ${getRankColor(
                  rek.peringkat
                )} rounded-full flex items-center justify-center font-bold text-lg`}
                >
                {rek.peringkat <= 3 ? "text-gray-900" : "text-gray-600"}
              </div>
              <div>
                {rek.peringkat}
              </div>
            </div>
            <div className="flex-1">
              <h4 className="font-semibold text-gray-900 text-lg">
                {rek.alternatif}
              </h4>
              <p className="text-sm text-gray-600 mt-1">
                Nilai Preferensi: {""}
                <span className="font-semibold text-gray-900">

```

```

        {rek.nilai_preferensi.toFixed(4)}
      </span>
    </p>
  </div>
  {rek.peringkat === 1 && (
    <Award className="w-6 h-6 text-yellow-600" />
  )}
</div>
</div>
  )))}
</div>
</div>

  {/* Action Buttons */}
  <div className="flex gap-3">
    <button
      onClick={handleReset}
      className="flex-1 bg-white border-2 border-yellow-500 text-yellow-600 font-
semibold py-3 px-6 rounded-lg hover:bg-yellow-50 transition-colors"
    >
      Hitung Ulang dengan Bobot Baru
    </button>
  </div>
</div>
);
};

export default Rekomendasi;

const Footer = () => {
  return (
    <footer className="bg-white border-t border-gray-200 mt-16">
      <div className="max-w-6xl mx-auto px-4 py-6 text-center text-sm text-
gray-600">
        <p>Sistem Rekomendasi Mata Kuliah Pilihan menggunakan Metode
SWARA</p>
        <p className="mt-1">Fakultas Ilmu Komputer - Teknik Informatika</p>
      </div>
    </footer>
  );
};

export default Footer;

```

```

import React, { useState, useEffect } from "react";
import {
  GraduationCap,
  TrendingUp,
  AlertCircle,
  CheckCircle2,
  Loader2,
  Award,
  BarChart3,
} from "lucide-react";
import { getKriteria, getAlternatif, hitungRekomendasi } from "../service/api";
import Header from "../components/Header";
import Progress from "../components/Progress";
import InfoCard from "../components/InfoCard";
import AlternatifPreview from "../components/AlternatifPreview";
import InputBobot from "../components/InputBobot";
import Rekomendasi from "../components/Rekomendasi";
import Footer from "../components/Footer";

function Home() {
  const [kriteria, setKriteria] = useState([]);
  const [alternatif, setAlternatif] = useState([]);
  const [bobot, setBobot] = useState({});
  const [hasil, setHasil] = useState(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);
  const [step, setStep] = useState(1);

  useEffect(() => {
    fetchInitialData();
  }, []);

  const fetchInitialData = async () => {
    try {
      const [kriteriaData, alternatifData] = await Promise.all([
        getKriteria(),
        getAlternatif(),
      ]);

      setKriteria(kriteriaData.kriteria);
      setAlternatif(alternatifData.alternatif);

      // Initialize bobot dengan nilai default
      const initialBobot = {};

```

```

    kriteriaData.kriteria.forEach((k, index) => {
      initialBobot[k.kode] = 5 - index;
    });
    setBobot(initialBobot);
  } catch (err) {
    setError(err.message);
  }
};

const handleBobotChange = (kode, value) => {
  const numValue = parseFloat(value);
  if (numValue >= 1 && numValue <= 5) {
    setBobot((prev) => ({ ...prev, [kode]: numValue }));
    setError(null);
  }
};

const validateBobot = () => {
  const values = Object.values(bobot);
  const uniqueValues = new Set(values);

  if (uniqueValues.size !== values.length) {
    setError("Nilai bobot harus unik (tidak boleh ada yang sama)");
    return false;
  }

  if (values.some((v) => v < 1 || v > 5)) {
    setError("Nilai bobot harus antara 1 sampai 5");
    return false;
  }

  return true;
};

const handleSubmit = async () => {
  if (!validateBobot()) return;

  setLoading(true);
  setError(null);

  try {
    const result = await hitungRekomendasi(bobot);
    setHasil(result);
    setStep(2);
  }
};

```



```

    } catch (err) {
      setError(err.message);
    } finally {
      setLoading(false);
    }
  };

```

```

const handleReset = () => {
  setStep(1);
  setHasil(null);
  setError(null);
  fetchInitialData();
};

```

```

const getRankColor = (peringkat) => {
  if (peringkat === 1) return "bg-yellow-500";
  if (peringkat === 2) return "bg-yellow-400";
  if (peringkat === 3) return "bg-yellow-300";
  return "bg-gray-200";
};

```

```

return (
  <div className="min-h-screen bg-gradient-to-br from-yellow-50 via-white to-yellow-100">
    {/* Header */}
    <Header />

    <main className="max-w-6xl mx-auto px-4 py-8">
      {/* Progress Indicator */}
      <Progress step={step} />

      {/* Error Alert */}
      {error && <Error error={error} />}

      {/* Step 1: Input Bobot */}
      {step === 1 && (
        <div className="space-y-6">
          {/* Info Card */}
          <InfoCard />

          {/* Alternatif Preview */}
          <AlternatifPreview alternatif={alternatif} />

          {/* Input Bobot */}

```

```

        <InputBobot
          kriteria={kriteria}
          bobot={bobot}
          loading={loading}
          handleSubmit={handleSubmit}
          handleBobotChange={handleBobotChange}
        />
      </div>
    )}

    { /* Step 2: Hasil Rekomendasi */ }
    { step === 2 && hasil && (
      <Rekomendasi
        hasil={hasil}
        getRankColor={getRankColor}
        handleReset={handleReset}
      />
    ) }
  </main>

  { /* Footer */ }
  <Footer />
</div>

);
}

export default Home;

```