

# Contoh implementasi DSS (*Decision Support System*) dengan metode SAW (*Simple Additive Weighting*) menggunakan PHP dan MySQL untuk pemilihan manajer IT

Metode *Simple Additive Weighting* (SAW) merupakan salah satu metode untuk menyelesaikan masalah *Multi-Attribute Decision Making* (MADM) dengan mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut.

[ author : [cahya dsn](#) , published on : [February 6, 2015](#) updated on : [August 16, 2018](#) ]

[minerva](#) [donasi](#) 

**Mau lihat artikel lainnya?** Dapatkan artikel-artikel lain seputar pemrograman website di [sini](#), dan dapatkan ide-ide baru

## Pendahuluan

langkah-langkah SAW

Kelebihan Kekurangan

## Studi Kasus dan Perhitungan

Kecocokan Alternatif dan Kriteria

Bobot Kriteria

Perhitungan Manual

## Aplikasi PHP

Persiapan Database

Koneksi Database

Langkah-langkah SAW

## Simpulan

## Pustaka

# 1. Pendahuluan

Metode *Simple Additive Weighting* (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn 1967). SAW dapat dianggap sebagai cara yang paling mudah dan intuitif untuk menangani masalah *Multiple Criteria Decision-Making* MCDM, karena fungsi linear additive dapat mewakili preferensi pembuat keputusan (*Decision-Making, DM*). Hal tersebut dapat dibenarkan, namun, hanya ketika asumsi *preference independence* (Keeney & Raiffa 1976) atau *preference separability* (Gorman 1968) terpenuhi.

Metode SAW membutuhkan proses normalisasi matriks keputusan ( $X$ ) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making* (MADM). MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Metode SAW ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya

## Pre-requisites

- Pemahaman terhadap dasar-dasar Sistem Pengambilan Keputusan
- Pemahaman terhadap dasar-dasar teknologi Web,HTML dan CSS
- Pemahaman terhadap dasar-dasar basis data/database, terutama query SQL pada MySQL/mariaDB
- Pemahaman terhadap dasar-dasar pemrograman PHP, terutama fungsi-fungsi koneksi database dan pengelolaan tipe data array

## 1.1. Langkah-langkah SAW

Langkah Penyelesaian *Simple Additive Weighting* (SAW) adalah sebagai berikut :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu  $C_i$
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria ( $x$ ).
3. Membuat matriks keputusan berdasarkan kriteria( $C_i$ ), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi  $R$ .
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi  $R$  dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik ( $A_i$ ) sebagai solusi.

Churchman dan Ackoff (1954) adalah yang pertama kali menggunakan Metode SAW untuk menangani masalah pemilihan portofolio. Metode SAW mungkin adalah metode yang paling terkenal dan paling banyak digunakan untuk *Multiple Attribute Decision Making* MADM. Metode SAW, karena kesederhanaannya, adalah metode yang paling populer dalam masalah MADM dan alternatif terbaik bisa diturunkan dari persamaan berikut:

$$A^* = \left\{ u_i(x) \mid \max_i u_i(x) \mid i = 1, 2, \dots, n \right\},$$

atau kesenjangan alternatif dapat ditingkatkan untuk membangun alternatif baru terbaik  $A^*$  untuk mencapai tingkat yang dicita-citakan/diinginkan pada setiap kriteria. Dan juga :

$$u_i(x) = \sum_{j=1}^n w_j r_{ij}(x)$$

Dimana  $U_i(x)$  menunjukkan kegunaan(*utility*) dari alternatif ke-  $i$ , dan  $i=1, 2, \dots, n$ ; . Sedangkan  $w_j$  menunjukkan bobot(*weight*) dari kriteria ke  $j$ . Dalam persamaan tersebut  $r_{ij}(x)$  adalah peringkat ternormalisasi yang dipilih dari alternatif ke-  $i$  yang berkaitan dengan kriteria ke-  $j$  untuk semua unit yang sepadan; dengan asumsi semua kriteria ada independen. Selain itu, peringkat ternormalisasi yang dipilih  $r_{ij}(x)$  dari alternatif ke-  $i$  yang berhubungan dengan kriteria ke-  $j$  dapat didefinisikan sebagai berikut:

#### Bentuk 1

- Untuk kriteria keuntungan *benefit* (lebih besar lebih baik),  $r_{ij}(x) = x_{ij} / x_j^*$ , dengan  $x_j^* = \max_i x_{ij}$  atau jadikan  $x_j^*$  sebagai tingkat yang diinginkan, dan dengan syarat  $0 \leq r_{ij}(x) \leq 1$
- Untuk kriteria kerugian *cost* (lebih kecil lebih baik),  $r_{ij}(x) = (1/x_{ij}) / (1/x_j^*) = (\max_i x_j^*) / (x_{ij})$  atau tetap jadikan  $x_j^*$  sebagai tingkat yang diinginkan.

#### Bentuk 2

- Untuk kriteria keuntungan *benefit* (lebih besar lebih baik),  $r_{ij}(x) = (x_{ij} - x_j^-) / (x_j^* - x_j^-)$ , dengan  $x_j^* = \max_i x_{ij}$  dan  $x_j^- = \min_i x_{ij}$  atau jadikan  $x_j^*$  sebagai tingkat yang diinginkan(terbaik) dan  $x_j^-$  sebagai tingkat yang paling tidak diinginkan(terburuk).
- Untuk kriteria kerugian *cost* (lebih kecil lebih baik),  $r_{ij}(x) = (x_j^- - x_{ij}) / (x_j^- - x_j^*)$

Oleh karena itu, kinerja disintesisnya adalah :

$$p_i = \sum_{j=1}^m w_j r_{ij}$$

Dimana  $p_i$  adalah nilai kinerja sintesis atau nilai preferensi dari alternatif ke-  $i$ ;  $w_j$  menunjukkan bobot dari kriteria ke  $j$ ;  $r_{ij}$  adalah peringkat ternormalisasi yang dipilih dari alternatif ke-  $i$  terhadap kriteria ke-  $j$  untuk menjadi unit sepadan; dan kriteria-kriteria-nya diasumsikan independen satu sama lain. Jika unit matriks kinerja adalah unit sepadan, kita tidak perlu untuk mentransfer matriks data ke dalam skala penilaian ternormalisasi yang terpilih.

## 1.2. Kelebihan dan Kekurangan

### 1.2.1. Kelebihan dari Metode SAW

Kelebihan dari metode simple additive weighting dibanding dengan model pengambil keputusan lainnya terletak pada kemampuannya untuk melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan, selain itu SAW juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada karena adanya proses perankingan setelah menentukan bobot untuk setiap atribut ( Kusumadewi 2006 ).

1. Menentukan nilai bobot untuk setiap atribut kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif.

2. Penilaian akan lebih tepat karena didasarkan pada nilai kriteria dari bobot preferensi yang sudah ditentukan.

### 1.2.2. Kekurangan dari Metode SAW

1. Perhitungan dilakukan dengan menggunakan bilangan *crisp* maupun *fuzzy*.
2. Adanya perbedaan perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai *benefit* dan *cost*)

## 2. Studi Kasus dan Perhitungan

This document using **Dynamic Content Technology™** for enrichment sample case and reading experience

- Data yang digunakan BUKAN merupakan data *real*, tapi data yang *digenerate* secara otomatis/random/acak dari sistem
- Data dan Nilai Perhitungan yang ditampilkan akan SELALU BERBEDA jika halaman di *refresh/reload*
- Jumlah Data Alternatif ditampilkan secara acak/*random* antara 5 s.d 10

Penerapan Metode **Simple Additive Weighting** (SAW) diharapkan mampu untuk membantu dalam menentukan Manajer IT baru dari beberapa kandidat karyawan yang diajukan, untuk mengelola dan memimpin departemen IT

Sebuah perusahaan IT membutuhkan seorang manager IT yang akan diarahkan untuk mengelola dan memimpin departemen IT. Ada **8** kandidat yang akan dipilih dari hasil interview yang sudah dilakukan oleh tim HRD dan Management yang akan dijadikan alternatif; yaitu A<sub>1</sub>: Lina P., A<sub>2</sub>: A. Alfian, A<sub>3</sub>: Yuna D., A<sub>4</sub>: M. Tantri, A<sub>5</sub>: M. Zaki, A<sub>6</sub>: Bella M., A<sub>7</sub>: James, dan A<sub>8</sub>: Nina P. .

Ada 5 kriteria dasar yang menjadi acuan dalam pengambilan keputusan, yaitu:

- C<sub>1</sub>: Penguasaan Aspek Teknis (skala 1-10)
- C<sub>2</sub>: Pengalaman Kerja (dalam tahun)
- C<sub>3</sub>: Interpersonal Skill (skala 1-10)
- C<sub>4</sub>: Usia (dalam tahun)
- C<sub>5</sub>: Status Perkawinan (5:blm menikah, 8:menikah tanpa tanggungan, 10: menikah dgn tanggungan)

### 2.1 Kecocokan Alternatif dan Kriteria

Tabel 1 menunjukkan rating kecocokan dari setiap alternatif dengan setiap kriteria. Nilai setiap alternatif pada setiap atribut diberikan berdasarkan data riil.

TABEL 1: Rating kecocokan dari setiap alternatif pada setiap kriteria

Alternatif	Kriteria				
	C1	C2	C3	C4	C5
A1 Lina P.	9	1	8.5	38	8
A2 A. Alfian	8	1	8	43	10
A3 Yuna D.	7.5	8.5	7.5	41	10
A4 M. Tantri	6.5	7	6.5	32	8
A5 M. Zaki	6	6	8.5	40	5
A6 Bella M.	8	6.5	8.5	31	8
A7 James	8.5	4.5	7.5	38	5
A8 Nina P.	7	7.5	7	35	5

## 2.2. Bobot Kriteria

Pengambil keputusan memberi bobot preferensi dari setiap kriteria sebagai:  $W=(1.9, 2.1, 3, 2, 2.5)$  dengan masing-masing jenisnya (keuntungan/*benefit* atau biaya/*cost*) seperti dalam tabel 2 berikut:

TABEL 2: Kriteria yang ditentukan

Kriteria	Deskripsi	Bobot	Atribut
C <sub>1</sub>	Penguasaan Aspek teknis	1.9	benefit
C <sub>2</sub>	Pengalaman Kerja	2.1	benefit
C <sub>3</sub>	Interpersonal Skill	3	benefit
C <sub>4</sub>	Usia	2	cost
C <sub>5</sub>	Status Perkawinan	2.5	cost

## 2.3. Perhitungan Manual

Pada bagian ini akan dijelaskan mengenai perhitungan peluang/probabilitas dari data training yang diberikan. Penjelasannya sendiri disajikan secara langkah per langkah agar mudah dipahami.

### 2.3.1. Matriks Keputusan (X)

Pertama-tama berdasarkan nilai data kecocokan antara alternatif dan kriteria pada [TABEL 1](#) dapat dibuatkan matriks keputusan (X) sebagai berikut:

$$\begin{bmatrix} 9 & 1 & 8.5 & 38 & 8 \\ 8 & 1 & 8 & 43 & 10 \end{bmatrix}$$

7.5	8.5	7.5	41	10
6.5	7	6.5	32	8
6	6	8.5	40	5
8	6.5	8.5	31	8
8.5	4.5	7.5	38	5
7	7.5	7	35	5

## 2.3.2. Perhitungan Matriks Ternormalisasi (R)

Tahap berikutnya adalah melakukan perhitungan normalisasi untuk mendapatkan matriks nilai ternormalisasi (R), dengan ketentuan :

Untuk normalisasi nilai, jika faktor/attribute kriteria bertipe *cost* maka digunakan rumusan:

$$R_{ij} = ( \min\{X_{ij}\} / X_{ij} )$$

sedangkan jika faktor/attribute kriteria bertipe *benefit* maka digunakan rumusan:

$$R_{ij} = ( X_{ij} / \max\{X_{ij}\} )$$

Sehingga dapat dihitung nilai-nilai ternormalisasi-nya (R) untuk tiap kriteria dan alternatif sebagai berikut:

### 2.3.2.1. Kriteria Penguasaan Aspek teknis

Pada kriteria Penguasaan Aspek teknis mempunyai tipe *benefit*, maka dicari nilai maksimum-nya ( $\max(X_{ij})$ ) terlebih dahulu; dalam hal ini diperoleh  $\max(X_{ij}) = 9$ ; yaitu didapat dari nilai tertinggi pada kolom ke-1. Nilai sehingga ternormalisasi-nya adalah dengan membagi nilai masing-masing alternatif dengan nilai maksimum kolom tersebut seperti perhitungan berikut:

$$R_{11} = 9/9 = 1$$

$$R_{21} = 8/9 = 0.89$$

$$R_{31} = 7.5/9 = 0.83$$

$$R_{41} = 6.5/9 = 0.72$$

$$R_{51} = 6/9 = 0.67$$

$$R_{61} = 8/9 = 0.89$$

$$R_{71} = 8.5/9 = 0.94$$

$$R_{81} = 7/9 = 0.78$$

### 2.3.2.2. Kriteria Pengalaman Kerja

Pada kriteria Pengalaman Kerja mempunyai tipe *benefit*, maka dicari nilai maksimum-nya ( $\max(X_{ij})$ ) terlebih dahulu; dalam hal ini diperoleh  $\max(X_{ij}) = 8.5$ ; yaitu didapat dari nilai tertinggi pada kolom ke-2. Nilai sehingga ternormalisasi-nya adalah dengan membagi nilai masing-masing alternatif dengan nilai maksimum kolom tersebut seperti perhitungan berikut:

$$R_{12} = 1/8.5 = 0.12$$

$$R_{22} = 1/8.5 = 0.12$$

$$R_{32} = 8.5/8.5 = 1$$

$$R_{42} = 7/8.5 = 0.82$$

$$R_{52} = 6/8.5 = 0.71$$

$$R_{62} = 6.5/8.5 = 0.76$$

$$R_{72} = 4.5/8.5 = 0.53$$

$$R_{82} = 7.5/8.5 = 0.88$$

### 2.3.2.3. Kriteria Interpersonal Skill

Pada kriteria Interpersonal Skill mempunyai tipe **benefit**, maka dicari nilai maksimum-nya ( $\max(X_{ij})$ ) terlebih dahulu; dalam hal ini diperoleh  $\max(X_{ij}) = 8.5$ ; yaitu didapat dari nilai tertinggi pada kolom ke-3. Nilai sehingga ternormalisasi-nya adalah dengan membagi nilai masing-masing alternatif dengan nilai maksimum kolom tersebut seperti perhitungan berikut:

$$R_{13} = 8.5/8.5 = 1$$

$$R_{23} = 8/8.5 = 0.94$$

$$R_{33} = 7.5/8.5 = 0.88$$

$$R_{43} = 6.5/8.5 = 0.76$$

$$R_{53} = 8.5/8.5 = 1$$

$$R_{63} = 8.5/8.5 = 1$$

$$R_{73} = 7.5/8.5 = 0.88$$

$$R_{83} = 7/8.5 = 0.82$$

### 2.3.2.4. Kriteria Usia

Pada kriteria Usia mempunyai tipe **cost**, maka dicari nilai minimum-nya ( $\min(X_{ij})$ ) terlebih dahulu; dalam hal ini diperoleh  $\min(X_{ij}) = 31$ ; yaitu didapat dari nilai terendah pada kolom ke-4. Nilai sehingga ternormalisasi-nya adalah dengan membagi nilai minimum kolom tersebut dengan nilai masing-masing alternatif seperti perhitungan berikut:

$$R_{14} = 31/38 = 0.82$$

$$R_{24} = 31/43 = 0.72$$

$$R_{34} = 31/41 = 0.76$$

$$R_{44} = 31/32 = 0.97$$

$$R_{54} = 31/40 = 0.78$$

$$R_{64} = 31/31 = 1$$

$$R_{74} = 31/38 = 0.82$$

$$R_{84} = 31/35 = 0.89$$

### 2.3.2.5. Kriteria Status Perkawinan

Pada kriteria Status Perkawinan mempunyai tipe **cost**, maka dicari nilai minimum-nya ( $\min(X_{ij})$ ) terlebih dahulu; dalam hal ini diperoleh  $\min(X_{ij}) = 5$ ; yaitu didapat dari nilai terendah pada kolom ke-5. Nilai sehingga ternormalisasi-nya adalah dengan membagi nilai minimum kolom tersebut dengan nilai masing-masing alternatif seperti perhitungan berikut:

$$R_{15} = 5/8 = 0.63$$

$$R_{25} = 5/10 = 0.5$$

$$R_{35} = 5/10 = 0.5$$

$$R_{45} = 5/8 = 0.63$$

$$R_{55} = 5/5 = 1$$

$$R_{65} = 5/8 = 0.63$$

$$R_{75} = 5/5 = 1$$

$$R_{85} = 5/5 = 1$$

Dari hasil-hasil perhitungan tersebut dapat dibuat matrik ternormalisasi (**R**) sebagai berikut:

1	0.12	1	0.82	0.63
0.89	0.12	0.94	0.72	0.5
0.83	1	0.88	0.76	0.5
0.72	0.82	0.76	0.97	0.63

0.67	0.71	1	0.78	1
0.89	0.76	1	1	0.63
0.94	0.53	0.88	0.82	1
0.78	0.88	0.82	0.89	1

### 2.3.3. Perhitungan Nilai Preferensi ( P )

Nilai preferensi ( P ) diperoleh dari penjumlahan perkalian nilai ternormalisasi ( R ) dengan bobot kriteria ( W ) untuk masing-masing Alternatif ( A ), sesuai dengan persamaan SAW-03. Perhitungan untuk masing-masing alternatif ( A ) adalah sebagai berikut:

$$A_1 = 1 * 1.9 + 0.12 * 2.1 + 1 * 3 + 0.82 * 2 + 0.63 * 2.5$$

$$A_1 = \mathbf{8.367}$$

$$A_2 = 0.89 * 1.9 + 0.12 * 2.1 + 0.94 * 3 + 0.72 * 2 + 0.5 * 2.5$$

$$A_2 = \mathbf{7.453}$$

$$A_3 = 0.83 * 1.9 + 1 * 2.1 + 0.88 * 3 + 0.76 * 2 + 0.5 * 2.5$$

$$A_3 = \mathbf{9.087}$$

$$A_4 = 0.72 * 1.9 + 0.82 * 2.1 + 0.76 * 3 + 0.97 * 2 + 0.63 * 2.5$$

$$A_4 = \mathbf{8.885}$$

$$A_5 = 0.67 * 1.9 + 0.71 * 2.1 + 1 * 3 + 0.78 * 2 + 1 * 2.5$$

$$A_5 = \mathbf{9.824}$$

$$A_6 = 0.89 * 1.9 + 0.76 * 2.1 + 1 * 3 + 1 * 2 + 0.63 * 2.5$$

$$A_6 = \mathbf{9.862}$$

$$A_7 = 0.94 * 1.9 + 0.53 * 2.1 + 0.88 * 3 + 0.82 * 2 + 1 * 2.5$$

$$A_7 = \mathbf{9.679}$$

$$A_8 = 0.78 * 1.9 + 0.88 * 2.1 + 0.82 * 3 + 0.89 * 2 + 1 * 2.5$$

$$A_8 = \mathbf{10.07}$$

### 2.3.4. Perangkingan

Dari hasil perhitungan nilai preferensi ( P ) sebelumnya, maka dapat dilakukan perangkingan dengan diurutkan berdasarkan nilai yang terbesar sebagai berikut:

$$A_8 = 10.07$$

$$A_6 = 9.862$$

$$A_5 = 9.824$$

$$A_7 = 9.679$$

$$A_3 = 9.087$$

$$A_4 = 8.885$$

$$A_1 = 8.367$$

$$A_2 = 7.453$$

Sehingga diperoleh hasil Alternatif **A<sub>8</sub> (Nina P.)** dengan nilai **10.07** menjadi yang terpilih sebagai manajer IT karena mempunyai nilai akhir perangkingan yang tertinggi



## 3. Aplikasi PHP

Sebagai pelengkap artikel Metode **Simple Additive Weighting** (SAW) pada bagian ini akan dibahas langkah-langkah dalam implementasinya dengan bahasa pemrograman PHP dan database MySQL/MariaDB.

### 3.1. Persiapan Database

Sebagai bahan pembelajaran aplikasi SAW ini; dibuat database (dalam hal ini menggunakan MySQL/MariaDB Database server) sebagai berikut:

```
CREATE DATABASE IF NOT EXISTS db_dss;  
USE db_dss;
```

Awalnya membuat dulu database dengan nama `db_dss` jika belum ada database dengan nama tersebut, kemudian gunakan database tersebut dengan memakai sintak `USE db_dss;`

Dalam hal ini, pembuatan database memakai *command console* dari database server yang bersangkutan

#### 3.1.1. Membuat Data Tabel Kriteria

Berdasarkan contoh kasus di atas dibuatkan tabel untuk data-data kriteria sebagai berikut:

```
DROP TABLE IF EXISTS saw_criterias;  
CREATE TABLE IF NOT EXISTS saw_criterias(  
  id_criteria TINYINT(3) UNSIGNED NOT NULL,  
  criteria VARCHAR(100) NOT NULL,  
  weight FLOAT NOT NULL,  
  attribute SET('benefit','cost'),  
  PRIMARY KEY(id_criteria)  
)ENGINE=MyISAM;  
  
INSERT INTO saw_criterias(id_criteria,criteria,weight,attribute)  
VALUES  
(1,'Penguasaan Aspek teknis',1.9,'benefit'),  
(2,'Pengalaman Kerja',2.1,'benefit'),  
(3,'Interpersonal Skill',3,'benefit'),  
(4,'Usia',2,'cost'),  
(5,'Status Perkawinan',2.5,'cost');
```

#### 3.1.2. Membuat Data Tabel Alternatif

Data-data mengenai kandidat yang akan dievaluasi seperti yang tertera pada contoh kasus di atas dapat di representasikan dalam tabel database sebagai berikut:

```
DROP TABLE IF EXISTS saw_alternatives;  
CREATE TABLE IF NOT EXISTS saw_alternatives(  
  id_alternative SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,  
  name VARCHAR(30) NOT NULL,
```

```

PRIMARY KEY(id_alternative)
) ENGINE=MyISAM;

INSERT INTO saw_alternatives(name)
VALUES
('Lina P.'),
('A. Alfian'),
('Yuna D.'),
('M. Tantri'),
('M. Zaki'),
('Bella M.'),
('James'),
('Nina P.');
```

Dalam tabel `saw_alternatives` tersebut hanya disimpan id dan nama alternatif-nya; dalam pengembangannya dapat ditambahkan atribut/properti data lainnya, semisal alamat, nomor telepon, email, dan sebagainya sesuai kebutuhan. Namun dalam contoh ini hanya diperlukan namanya saja.

### 3.1.3. Membuat Data Tabel Hasil Evaluasi

Berikutnya adalah membuat tabel yang berisi hasil evaluasi dari tiap-tiap kandidat(alternatif) terhadap kriteria-kriteria yang diberikan. Data-data yang ada pada contoh kasus di [atas](#) dapat di representasikan dalam tabel relasi antara tabel `saw_criteria` dengan tabel `saw_alternatives` sebagai berikut:

```

DROP TABLE IF EXISTS saw_evaluations;
CREATE TABLE IF NOT EXISTS saw_evaluations(
    id_alternative SMALLINT(5) UNSIGNED NOT NULL,
    id_criteria TINYINT(3) UNSIGNED NOT NULL,
    value FLOAT NOT NULL,
    PRIMARY KEY (id_alternative,id_criteria)
)ENGINE=MyISAM;

INSERT INTO saw_evaluations(id_alternative,id_criteria,value)
VALUES
(1,1,9)(1,2,1)(1,3,8.5)(1,4,38)(1,5,8),
(2,1,8)(2,2,1)(2,3,8)(2,4,43)(2,5,10),
(3,1,7.5)(3,2,8.5)(3,3,7.5)(3,4,41)(3,5,10),
(4,1,6.5)(4,2,7)(4,3,6.5)(4,4,32)(4,5,8),
(5,1,6)(5,2,6)(5,3,8.5)(5,4,40)(5,5,5),
(6,1,8)(6,2,6.5)(6,3,8.5)(6,4,31)(6,5,8),
(7,1,8.5)(7,2,4.5)(7,3,7.5)(7,4,38)(7,5,5),
(8,1,7)(8,2,7.5)(8,3,7)(8,4,35)(8,5,5);
```

## 3.2. Koneksi Ke Database Server

Dari database yang sudah dibuat, kita bisa membuat script php untuk membuat koneksi ke database server dengan extension mysqli sebagai berikut:

```

<?php
//-- konfigurasi database
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
```

```

$dbname = 'db_dss';
//-- koneksi ke database server dengan extension mysqli
$db = new mysqli($dbhost,$dbuser,$dbpass,$dbname);
//-- hentikan program dan tampilkan pesan kesalahan jika koneksi gagal
if ($db->connect_error) {
    die('Connect Error ('.$db->connect_errno.')'.$db->connect_error);
}
?>

```

Sesuaikan nilai-nilai `$dbhost`, `$dbuser`, `$dbpass` dan `$dbname` dengan konfigurasi database yg digunakan.

## 3.3. Langkah-Langkah SAW

Secara umum dalam SAW, langkah-langkah yang dilakukan dalam pengambilan keputusan adalah sebagai berikut:

- Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu  $C_i$
- Menentukan rating kecocokan setiap alternatif pada setiap kriteria
- Membuat matrik keputusan berdasarkan Kriteria  $C_i$
- Perankingan Nilai Vektor P

### 3.3.1. Menentukan Kriteria-kriteria Yang Akan Dijadikan Acuan ( $C_i$ )

Dari script sebelumnya dapat diteruskan untuk menampilkan kriteria-kriteria yg sudah ditentukan dengan mengambil data dari database dan ditampilkan dalam bentuk tabular/grid (bentuk tabel) dengan koding sebagai berikut:

```

<table border='1'>
  <caption>
    Tabel Kriteria C<sub>i</sub>
  </caption>
  <tr>
    <th>No</th>
    <th>Simbol</th>
    <th>Kriteria</th>
    <th>Atribut</th>
  </tr>
  <?php
    $sql='SELECT criteria,weight,attribute FROM saw_criterias';
    $result=$db->query($sql);
    $i=0;
    while($row=$result->fetch_object()){
      echo "<tr>
        <td class='right'>".(++$i)."</td>
        <td class='center'>C{$i}</td>
        <td>{$row->criteria}</td>
        <td>{$row->attribute}</td>
      </tr>\n";
    }
    $result->free();

```

```
?>
</table>
```

Hasil yang diperoleh adalah kurang lebih sebagai berikut:

Tabel Kriteria  $C_i$

No	Simbol	Kriteria	Atribut
1	$C_1$	Penguasaan Aspek teknis	benefit
2	$C_2$	Pengalaman Kerja	benefit
3	$C_3$	Interpersonal Skill	benefit
4	$C_4$	Usia	cost
5	$C_5$	Status Perkawinan	cost

### 3.3.2. Menentukan Bobot Kriteria

Setiap kriteria dalam semesta permasalahan diberikan bobot sesuai dengan pertimbangan pakar/pengambil keputusan -- dalam hal ini adalah dari HRD --, yang menunjukkan tingkat kepentingan suatu kriteria dibandingkan dengan kriteria lain yang ada di dalam semesta permasalahan ini

Bobot dari masing-masing kriteria diambil dari tabel `saw_criteria` dengan kode *script* sebagai berikut:

```
<table border='1'>
  <caption>
    Pembobotan Kriteria (w)
  </caption>
  <tr>
    <th>No</th>
    <th>Kriteria</th>
    <th>Bobot</th>
  </tr>
  <?php
    $sql="SELECT weight FROM saw_criteria ORDER BY id_criteria";
    $result=$db->query($sql);
    $i=0;
    $W=array();
    while($row=$result->fetch_object()){
      $W[]=$row->weight;
      echo "<tr class='center'>
        <td>".(++$i)."</td>
        <td>C{$i}</td>
        <td>{$row->weight}</td>
      </tr>";
    }
  ?>
</table>
```

Hasil dari *script* tersebut jika dijalankan di *browser* adalah seperti dibawah ini:

Tabel Pembobotan Kriteria (w)

No	Kriteria	Bobot
1	Penguasaan Aspek teknis	1.9

2	Pengalaman Kerja	2.1
3	Interpersonal Skill	3
4	Usia	2
5	Status Perkawinan	2.5

### 3.3.3. Menentukan Rating Kecocokan Setiap Alternatif Pada Setiap Kriteria

Teknik pembobotan pada setiap kriteria dari suatu alternatif dapat dilakukan dengan berbagai macam cara dan metode. Fase ini dikenal dengan istilah pra-proses. Namun bisa juga dengan cara secara sederhana dengan memberikan nilai pada masing-masing secara langsung berdasarkan persentasi nilai bobotnya. Sedangkan untuk yang lebih baik bisa digunakan fuzzy logic. Penggunaan Fuzzy logic, sangat dianjurkan bila kriteria yang dipilih mempunyai sifat yang relative, misal Umur, Panas, Tinggi, Baik atau sifat lainnya

```
<table border='1'>
  <caption>
    Rating Kecocokan Alternatif utk setiap Kriteria
  </caption>
  <tr>
    <th rowspan='2'>Alternatif</th>
    <th colspan='5'>Kriteria</th>
  </tr>
  <tr>
    <th>C1</th>
    <th>C2</th>
    <th>C3</th>
    <th>C4</th>
    <th>C5</th>
  </tr>
<?php
$sql="SELECT
    a.id_alternative,
    b.name,
    SUM(IF(a.id_criteria=1,a.value,0)) AS C1,
    SUM(IF(a.id_criteria=2,a.value,0)) AS C2,
    SUM(IF(a.id_criteria=3,a.value,0)) AS C3,
    SUM(IF(a.id_criteria=4,a.value,0)) AS C4,
    SUM(IF(a.id_criteria=5,a.value,0)) AS C5
FROM
    saw_evaluations a
    JOIN saw_alternatives b USING(id_alternative)
GROUP BY a.id_alternative
ORDER BY a.id_alternative";
$result=$db->query($sql);
while($row=$result->fetch_object()){
    echo "<tr class='center'>
        <th>A<sub>{$row->id_alternative}</sub> {$row->name}</th>
        <td>".round($row->C1,1). "</td>
        <td>".round($row->C2,1). "</td>
        <td>".round($row->C3,1). "</td>
```

```

        <td>".round($row->C4,1). "</td>
        <td>".round($row->C5,1). "</td>
    </tr>\n";
}
$result->free();
?>
</table>

```

Script PHP tersebut jika dijalankan akan menampilkan hasil di *browser* seperti berikut ini :

Rating Kecocokan Alternatif  
utk setiap Kriteria

Alternatif	Kriteria				
	C1	C2	C3	C4	C5
A <sub>1</sub> Lina P.	9	1	8.5	38	8
A <sub>2</sub> A. Alfian	8	1	8	43	10
A <sub>3</sub> Yuna D.	7.5	8.5	7.5	41	10
A <sub>4</sub> M. Tantri	6.5	7	6.5	32	8
A <sub>5</sub> M. Zaki	6	6	8.5	40	5
A <sub>6</sub> Bella M.	8	6.5	8.5	31	8
A <sub>7</sub> James	8.5	4.5	7.5	38	5
A <sub>8</sub> Nina P.	7	7.5	7	35	5

### 3.3.4. Membuat Matrik Keputusan (X) Berdasarkan Kriteria C<sub>i</sub>

```

<table border='1'>
    <caption>
        Matrik Keputusan(X)
    </caption>
    <tr>
        <th rowspan='2'>Alternatif</th>
        <th colspan='5'>Kriteria</th>
    </tr>
    <tr>
        <th>C1</th>
        <th>C2</th>
        <th>C3</th>
        <th>C4</th>
        <th>C5</th>
    </tr>
<?php
$sql="SELECT
    a.id_alternative,
    b.name,
    SUM(IF(a.id_criteria=1,a.value,0)) AS C1,
    SUM(IF(a.id_criteria=2,a.value,0)) AS C2,
    SUM(IF(a.id_criteria=3,a.value,0)) AS C3,
    SUM(IF(a.id_criteria=4,a.value,0)) AS C4,
    SUM(IF(a.id_criteria=5,a.value,0)) AS C5

```

```

FROM
    saw_evaluations a
    JOIN saw_alternatives b USING(id_alternative)
GROUP BY a.id_alternative
ORDER BY a.id_alternative";
$result=$db->query($sql);
$X=array(1=>array(),2=>array(),3=>array(),4=>array(),5=>array());
while($row=$result->fetch_object()){
    array_push($X[1],round($row->C1,2));
    array_push($X[2],round($row->C2,2));
    array_push($X[3],round($row->C3,2));
    array_push($X[4],round($row->C4,2));
    array_push($X[5],round($row->C5,2));
    echo "<tr class='center'>
        <th>A<sub>{$row->id_alternative}</sub> {$row->name}</th>
        <td>".round($row->C1,2). "</td>
        <td>".round($row->C2,2). "</td>
        <td>".round($row->C3,2). "</td>
        <td>".round($row->C4,2). "</td>
        <td>".round($row->C5,2). "</td>
    </tr>\n";
}
$result->free();
?>
</table>

```

### 3.3.5. Penghitungan Normalisasi (R)

```

<table border='1'>
    <caption>
        Matrik Ternormalisasi (R)
    </caption>
    <tr>
        <th rowspan='2'>Alternatif</th>
        <th colspan='5'>Kriteria</th>
    </tr>
    <tr>
        <th>C1</th>
        <th>C2</th>
        <th>C3</th>
        <th>C4</th>
        <th>C5</th>
    </tr>
<?php
$sql="SELECT
    a.id_alternative,
    SUM(
        IF(
            a.id_criteria=1,
            IF(

```

```

        b.attribute='benefit',
        a.value/"max($X[1]).",
        ".min($X[1])."/a.value)
    ,0)
    ) AS C1,
SUM(
    IF(
        a.id_criteria=2,
        IF(
            b.attribute='benefit',
            a.value/"max($X[2]).",
            ".min($X[2])."/a.value)
        ,0)
    ) AS C2,
SUM(
    IF(
        a.id_criteria=3,
        IF(
            b.attribute='benefit',
            a.value/"max($X[3]).",
            ".min($X[3])."/a.value)
        ,0)
    ) AS C3,
SUM(
    IF(
        a.id_criteria=4,
        IF(
            b.attribute='benefit',
            a.value/"max($X[4]).",
            ".min($X[4])."/a.value)
        ,0)
    ) AS C4,
SUM(
    IF(
        a.id_criteria=5,
        IF(
            b.attribute='benefit',
            a.value/"max($X[5]).",
            ".min($X[5])."/a.value)
        ,0)
    ) AS C5
FROM
    saw_evaluations a
    JOIN saw_criterias b USING(id_criteria)
GROUP BY a.id_alternative
ORDER BY a.id_alternative
";
$result=$db->query($sql);
$R=array();
while($row=$result->fetch_object()){
    $R[$row->id_alternative]=array($row->C1,$row->C2,$row->C3,$row->C4,$row->C5);

```



```

        echo "<tr class='center'>
            <th>A{$row->id_alternative}</th>
            <td>".round($row->C1,2). "</td>
            <td>".round($row->C2,2). "</td>
            <td>".round($row->C3,2). "</td>
            <td>".round($row->C4,2). "</td>
            <td>".round($row->C5,2). "</td>
        </tr>\n";
    }
?>
</table>

```

## Penghitungan Nilai Preferensi ( P )

Nilai preferensi ( P ) merupakan penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot W .

```

<table border='1'>
    <caption>
        Nilai Preferensi (P)
    </caption>
    <tr>
        <th>No</th>
        <th>Alternatif</th>
        <th>Hasil</th>
    </tr>
    <?php
    $P=array();
    $m=count($W);
    $no=0;
    foreach($R as $i=>$r){
        for($j=0;$j<$m;$j++){
            $P[$i]=(isset($P[$i])?$P[$i]:0)+$r[$j]*$W[$j];
        }
        echo "<tr class='center'>
            <td>".(++$no). "</td>
            <td>A{$i}</td>
            <td>{$P[$i]}</td>
        </tr>";
    }
?>
</table>

```

## Perangkingan

Proses perangkingan nilai preferensi ( P ) adalah dengan mengurutkan nilai akhir dari yang terbesar hingga yang terkecil sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik ( $A_i$ ) sebagai solusi

```

<table border='1'>
    <caption>

```

```

    Hasil Perangkingan
</caption>
<tr>
    <th>No</th>
    <th>Alternatif</th>
    <th>Hasil</th>
</tr>
<?php
arsort($P);
$no=0;
foreach($P as $i=>$p){
    echo "<tr class='center'>
        <td>".(++$no). "</td>
        <td>A{$i}</td>
        <td>{$p}</td>
    </tr>";
}
?>
</table>

```

## 4. Simpulan

Berikut simpulan yang dapat diambil dari contoh kasus dengan metode **Simple Additive Weighting** (SAW) tersebut:

- Metode **Simple Additive Weighting** (SAW) dapat digunakan untuk membantu proses pemilihan Manajer IT dari beberapa kandidat dengan kriteria-kriteria yang sudah ditentukan sebelumnya
- Dari beberapa kandidat terpilih alternatif **A<sub>8</sub> (Nina P.)** dengan total nilai perangkingan sebesar **10.07**

## 5. Pustaka

- Churchman, C.W., and R.L. Ackoff. (1954). **An approximate measure of value**. Journal of Operations Research Society of America 2 (1): 172-87
- Fishburn, P.C. (1970). **Utility theory for decision-making**. New York: Wiley
- Gorman, W.M. (1968a). **Conditions for additive separability**. Econometrica 36 (3/4): 605-9.
- Gorman, W.M. (1968b). **The structure of utility function**. Review of Economic Studies 35 (4):367-90.
- Keeney, R.L., and H. Raiffa. (1976). **Decision with multiple objectives: Preferences and value tradeoffs**. New York: John Wiley and Sons.
- Kusriani. (2007). **Konsep dan Aplikasi Sistem Pendukung Keputusan**. Yogyakarta: Penerbit Andi.
- Kusumadewi, S., Hartati, S., Harjoko, A., dan Wardoyo, R. (2006). **Fuzzy Multi-Attribute Decision Making (FUZZY MADM)**. Yogyakarta: Penerbit Graha Ilmu.

# Artikel Terkait

---

- [DSS AHP Method](#)
- [DSS CPI Method](#)
- [DSS MFEP Method](#)
- [DSS VIKOR Method](#)
- [DSS PROMETHEE Method](#)
- [DSS ELECTRE Method](#)
- [DSS ANP Method](#)
- [DSS Profile Matching Method](#)
- [DSS MOORA Method](#)
- [DSS TOPSIS Method](#)
- [DSS SMART Method](#)
- [DSS WP Method](#)
- [DSS MPE Method](#)

