

**SISTEM PENDUKUNG KEPUTUSAN PENILAIAN KINERJA GURU
BERBASIS *MULTI CRITERIA DECISION MAKING***

THESIS

**Oleh:
ANDREAS SETIYONO
NIM. 19841013**



**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022**

**SISTEM PENDUKUNG KEPUTUSAN PENILAIAN KINERJA GURU
BERBASIS *MULTI CRITERIA DECISION MAKING***

THESIS

**Oleh:
ANDREAS SETIYONO
NIM. 19841013**



**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022**

**SISTEM PENDUKUNG KEPUTUSAN PENILAIAN KINERJA GURU
BERBASIS *MULTI CRITERIA DECISION MAKING***

THESIS

**Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi salah satu Persyaratan dalam
Memperoleh Gelar Magister Komputer (M.Kom)**

**Oleh:
ANDREAS SETIYONO
NIM. 19841013**

**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022**

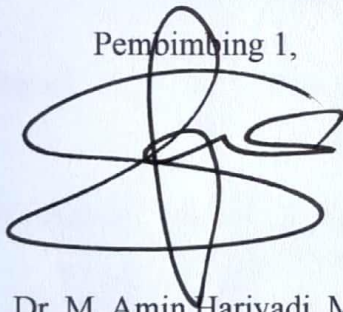
**SISTEM PENDUKUNG KEPUTUSAN PENILAIAN KINERJA GURU
BERBASIS *MULTI CRITERIA DECISION MAKING***

THESIS

**Oleh:
ANDREAS SETIYONO
NIM. 19841013**

Telah diperiksa dan disetujui untuk diuji:
Tanggal: 07 Desember 2022

Pembimbing 1,



Dr. M. Amin Hariyadi, MT
NIP. 19670118 200501 1 001

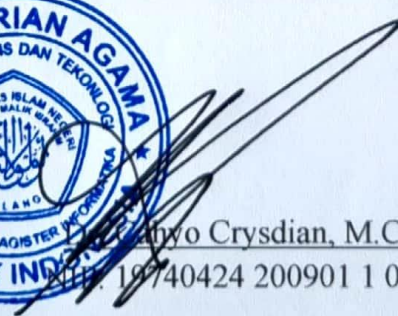
Pembimbing 2,



Dr. Sri Harini, M.Si
NIP. 19731014 200112 2 002

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Anvo Crysodian, M.Cs
NIP. 19740424 200901 1 008

**SISTEM PENDUKUNG KEPUTUSAN PENILAIAN KINERJA GURU
BERBASIS *MULTI CRITERIA DECISION MAKING***

THESIS

**Oleh:
ANDREAS SETIYONO
NIM. 19841013**

Telah Dipertahankan di depan Dewan Penguji Thesis
Dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal: 07 Desember 2022

Susunan Dewan Penguji

Penguji Utama : Dr. Cahyo Crysdiyan, M.Cs
NIP. 19740424 200901 1 008

Tanda Tangan

()

Ketua Penguji : Dr. Fachrul Kurniawan, ST., M. MT., IPM
NIP. 19771020 200912 1 001

()

Sekretaris Penguji: Dr. M. Amin Hariyadi, MT
NIP. 19670118 200501 1 001


()

Anggota Penguji : Dr. Sri Harini, M.Si
NIP. 19731014 200112 2 002

()

Mengetahui dan Mengesahkan
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdiyan, M.Cs
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Andreas Setiyono
NIM : 19841013
Program Studi : Magister Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa Thesis saya yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar Pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Thesis ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 27 Desember 2022
Yang membuat Pernyataan



Andreas Setiyono
NIM. 19841013

MOTTO

“We Work as Dedication to Alloh SWT and For The Future Civilized Generation”

HALAMAN PERSEMBAHAN

Terima kasih kepada Dosen pembimbing yang telah sabar mendampingi saya serta mengarahkan saya dalam melakukan penulisan karya ilmiah ini.

Karya ini saya persembahkan untuk kedua orang tua saya. Orang Tua yang telah mendukung secara maksimal dalam penulisan karya ilmiah ini

Terima kasih kepada keluarga saya, istri dan ketiga putri saya yang telah memberikan semangat dan dukungan untuk menyelesaikan karya tulis ilmiah ini

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Syukur alhamdulillah penulis haturkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Thesis ini dengan baik.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan jazakumullah ahsanal jaza' kepada semua pihak yang telah membantu terselesaikannya Thesis ini. Ucapan terima kasih ini penulis sampaikan kepada

1. Bapak Dr. M. Amin Hariyadi, MT dan Ibu Dr. Sri Harini, M.Si, selaku dosen pembimbing Thesis, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
2. Segenap sivitas akademika Program Studi Magister Informatika, terutama seluruh Bapak/ Ibu dosen, terima kasih atas segenap ilmu dan bimbingannya.
3. Ayahanda dan Ibunda tercinta yang senantiasa memberikan doa dan restunya kepada penulis dalam menuntut ilmu.
4. Istri dan anak-anak penulis yang selalu memberikan semangat kepada penulis untuk menyelesaikan Thesis ini.
5. Semua pihak yang ikut membantu dalam menyelesaikan Thesis ini baik berupa materiil maupun moril.

Penulis menyadari bahwa dalam penyusunan Thesis ini masih terdapat kekurangan dan penulis berharap semoga Thesis ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. Amin Ya Rabbal Alamin.

Wassalamu'alaikum Wr. Wb.

Malang, 27 Desember 2022
Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN.....	iv
MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR.....	vii
DAFTAR ISI	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xiv
ABSTRAK.....	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah.....	5
1.6 Sistematika Penulisan.....	5
BAB II STUDI PUSTAKA	7
2.1 Penilaian Kinerja Guru	7
2.2 Sistem Pendukung Keputusan	9
2.2.1 Database Manajemen	11
2.2.2 Model Base	11
2.2.3 User Interface	12
2.2.4 Pengembangan Sistem Pendukung Keputusan	12
2.3 Multi Criteria Decision Making	14
2.4 Metode Analytical Hierarchy Proses.....	14
2.5 Metode Weighted Sum Model.....	19
2.6 Metode Weighted Product Model	20
2.7 Metode Tecnique for Other Reference By Similarity to Ideal Solution.....	22
2.8 Perhitungan Ketidaksesuaian Perangkingan.....	25
2.9 Penelitian Terkait.....	26
2.10 Penilaian Kinerja Guru dalam Perspektif AL-Quran	28

BAB III METODE ANALYTICAL HIERARCHY PROSES (AHP).....	30
3.1 Desain Metode Analytical Hierarchy Proses	30
3.2 Implementasi Metode Analytical Hierarchy Proses	31
3.3 Uji Coba Analytical Hierarchy Proses	47
 BAB IV METODE WEIGHTED SUM MODEL (WSM).....	 55
4.1 Desain Metode Weighted Sum Model	55
4.2 Implementasi Metode Weighted Sum Model.....	55
4.3 Uji Coba Weighted Sum Model	68
 BAB V METODE WEIGHTED PRODUCT MODEL (WPM)	 76
5.1 Desain Metode Weighted Product Model	76
5.2 Implementasi Metode Weighted Product Model.....	76
5.3 Uji Coba Weighted Product Model	89
 BAB VI METODE METODE TECHNIQUE FOR OTHERS PREFERENCE BY SIMILARITY TO IDEAL SOLUTION (TOPSIS).....	 96
6.1 Desain Metode TOPSIS	96
6.2 Implementasi Metode TOPSIS	97
6.3 Uji Coba Metode TOPSIS	110
 BAB VII PEMBAHASAN	 119
7.1 Perbandingan metode AHP, WSM, WPM, dan TOPSIS	119
7.2 Perbandingan antar metode (dengan nilai kriteria yang sama).....	120
7.3 Perbandingan antar metode (dengan nilai kriteria yang tidak sama).....	121
7.4 Perbandingan hasil ranking manual dengan metode AHP, WSM, WPM, dan TOPSIS	124
 BAB VIII KESIMPULAN	 126
8.1 Kesimpulan	126
8.2 Saran	126
 DAFTAR PUSTAKA.....	 109

DAFTAR GAMBAR

Gambar 2.1 Tiga Komponen Besar SPK.....	11
Gambar 2.2 Fase-fase Pengembangan SPK	13
Gambar 2.3 Struktur Hierarki AHP	16
Gambar 3.1 Desain Metode Analytical Hierarchy Proses (AHP)	30
Gambar 3.2 Desain Database.....	31
Gambar 3.3 Hirarki AHP SPK kinerja Guru	33
Gambar 3.4 Form Input Nilai Peserta.....	34
Gambar 3.5 Kode Program Persentase Nilai Peserta	34
Gambar 3.6 Kode Program Pengkategorian Nilai Peserta	35
Gambar 3.7 Kode Program Function Query Nilai Peserta	36
Gambar 3.8 Tampilan Form Bobot Kriteria	38
Gambar 3.9 Kode Program Bobot Kriteria.....	38
Gambar 3.10 Kode Program Memanggil Function Ganti Bobot Kriteria	38
Gambar 3.11 Kode Program Function Query Ganti Bobot Kriteria.....	39
Gambar 3.12 Kode Program Memanggil Penilaian AHP.....	39
Gambar 3.13 Kode Program Function Query View Penilaian AHP	39
Gambar 3.14 Kode Program Pengolahan Penilaian AHP	40
Gambar 3.15 Kode Program Menghitung Perbandingan Berpasangan.....	41
Gambar 3.16 Kode Program Jumlah Perbandingan Berpasangan.....	41
Gambar 3.17 Kode Program Pembagian Nilai Berpasangan	41
Gambar 3.18 Kode Program Menghitung Nilai Priority	41
Gambar 3.19 Kode Program Perkalian Nilai Priority.....	42
Gambar 3.20 Kode Program Consistency Index (CI).....	42
Gambar 3.21 Kode Program Consistency Ratio (CR).....	42
Gambar 3.22 Kode Program Perangkingan	43
Gambar 3.23 Contoh Perangkingan Bersama.....	43
Gambar 3.24 Kode Program Untuk Mengambil Hasil Perhitungan Manual	44
Gambar 3.25 Kode Program Untuk Mencari Rangking Bersama	44
Gambar 3.26 Kode Program Mengambil hasil perangkingan dengan metode AHP. 45	
Gambar 3.27 Kode Program <i>hamming distance</i>	45

Gambar 4.1 Desain Metode Weighted Sum Model (WSM)	55
Gambar 4.2 Form Input Nilai Peserta.....	56
Gambar 4.3 Kode Program Persentase Nilai Peserta	57
Gambar 4.4 Kode Program Pengkategorian Nilai Peserta	57
Gambar 4.5 Kode Program Penyimpan Nilai Peserta	58
Gambar 4.6 Kode Program Function Query Nilai Peserta	58
Gambar 4.7 Kode Program Form Bobot Kriteria.....	59
Gambar 4.8 Tampilan Form Bobot Kriteria	60
Gambar 4.9 Kode Program Bobot Kriteria.....	60
Gambar 4.10 Kode Program Memanggil Function Ganti Bobot Kriteria	60
Gambar 4.11 Kode Program Function Query Ganti Bobot Kriteria.....	61
Gambar 4.12 Kode Program Memanggil Penilaian WSM	61
Gambar 4.14 Kode Program Memanggil Penilaian WSM	61
Gambar 4.15 Kode Program Pengolahan Penilaian WSM.....	62
Gambar 4.16 Kode Program Mencari Nilai Max Untuk Setiap Kriteria.....	63
Gambar 4.17 Kode Program Normalisasi Alternatif WSM	63
Gambar 4.18 Kode Program Menghitung Jumlah Total Bobot.....	63
Gambar 4.19 Kode Program Normalisasi Bobot Kriteria WSM.....	64
Gambar 4.20 Kode Program Perhitungan Nilai WSM	64
Gambar 4.21 Kode Program Perangkingan	64
Gambar 4.22 Contoh Perangkingan Bersama.....	65
Gambar 4.23 Kode Program Untuk Mencari Rangking Bersama	66
Gambar 4.24 Kode Program Mengambil hasil perangkingan metode WSM.....	67
Gambar 4.25 Kode Program <i>hamming distance</i>	68
Gambar 5.1 Desain Metode Weighted Product Model (WPM)	76
Gambar 5.2 Form Input Nilai Peserta.....	77
Gambar 5.3 Kode Program Persentase Nilai Peserta	78
Gambar 5.4 Kode Program Pengkategorian Nilai Peserta	78
Gambar 5.5 Kode Program Penyimpan Nilai Peserta	79
Gambar 5.6 Kode Program Function Query Nilai Peserta	79
Gambar 5.7 Kode Program Form Bobot Kriteria.....	80
Gambar 5.8 Tampilan Form Bobot Kriteria	81

Gambar 5.9 Kode Program Bobot Kriteria.....	81
Gambar 5.11 Kode Program Function Query Ganti Bobot Kriteria.....	82
Gambar 5.12 Kode Program Memanggil Penilaian WPM	82
Gambar 5.14 Kode Program Memanggil Penilaian WPM	82
Gambar 5.15 Kode Program Pengolahan Penilaian WPM.....	83
Gambar 5.16 Kode Program Penjumlahan Bobot KriteriaWPM	83
Gambar 5.17 Kode Program Normalisasi Bobot Kriteria WPM.....	84
Gambar 5.18 Kode Program Pemangkat Alternatif WPM.....	84
Gambar 5.19 Kode Program Nilai Vektor S WPM	84
Gambar 5.20 Kode Program Nilai WPM	85
Gambar 5.21 Kode Program Perangkingan	85
Gambar 5.22 Contoh Perangkingan Bersama.....	86
Gambar 5.23 Kode Program Untuk Mengambil Hasil Perhitungan Manual	86
Gambar 5.24 Kode Program Untuk Mencari Rangking Bersama	87
Gambar 5.25 Kode Program Mengambil hasil perangkingan metode WPM.....	87
Gambar 5.26 Kode Program <i>hamming distance</i>	88
Gambar 6.1 Desain TOPSIS	96
Gambar 6.2 Form Input Nilai Peserta.....	97
Gambar 6.3 Kode Program Persentase Nilai Peserta	98
Gambar 6.4 Kode Program Pengkategorian Nilai Peserta	98
Gambar 6.5 Kode Program Penyimpan Nilai Peserta	99
Gambar 6.6 Kode Program Function Query Nilai Peserta	100
Gambar 6.7 Kode Program Form Bobot Kriteria	100
Gambar 6.8 Tampilan Form Bobot Kriteria	101
Gambar 6.9 Kode Program Bobot Kriteria.....	101
Gambar 6.10 Kode Program Memanggil Function Ganti Bobot Kriteria	101
Gambar 6.11 Kode Program Function Query Ganti Bobot Kriteria.....	102
Gambar 6.12 Kode Program Memanggil Penilaian TOPSIS	102
Gambar 6.14 Kode Program Memanggil Penilaian Query Database.....	102
Gambar 6.15 Kode Program Pengolahan Penilaian TOPSIS	103
Gambar 6.16 Kode Program Normalisasi Alternatif TOPSIS.....	104
Gambar 6.17 Kode Program Solusi Ideal Positif TOPSIS	104

Gambar 6.18 Kode Program Solusi Ideal Negative TOPSIS	105
Gambar 6.19 Kode Program Jarak Positif TOPSIS	105
Gambar 6.20 Kode Program Jarak Negative TOPSIS	106
Gambar 6.21 Kode Program Perangkingan	106
Gambar 6.22 Contoh Perangkingan Bersama.....	107
Gambar 6.23 Kode Program Untuk Mengambil Hasil Perhitungan Manual	107
Gambar 6.24 Kode Program Untuk Mencari Rangking Bersama	108
Gambar 6.25 Kode Program Mengambil hasil perangkingan metode TOPSIS	108
Gambar 6.26 Kode Program <i>hamming distance</i>	109
Gambar 7.1 Perbandingan Metode AHP, WSM, WPM, dan TOPSIS	120
Gambar 7.2 Perbandingan Metode AHP, WSM, WPM, dan TOPSIS dengan Bobot Kriteria Sama	121
Gambar 7.3 Perbandingan Metode AHP, WSM, WPM, dan TOPSIS dengan Bobot Kriteria Sama	122
Gambar 7.4 Perbandingan Bobot Kriteria Sama dan Tidak Sama	123
Gambar 7.5 Gambar 7.5 Perbandingan penghitungan manuan dan metode	125

DAFTAR TABEL

Tabel 2.1 Tabel Kompetensi Penilaian Kinerja Guru.....	8
Tabel 2.2 Skala Penilaian Perbandingan Berpasangan.....	16
Tabel 2.3 Indeks Random Consistency (IR) berdasarkan ukuran matriks	19
Tabel 2.4 Daftar penelitian yang relevan.....	26
Tabel 3.1 Pengujian AHP	47
Tabel 3.2 Bobot Kriteria Pengujian 1 AHP	48
Tabel 3.3 Hasil Pengujian 1 AHP	49
Tabel 3.4 Bobot Kriteria Pengujian 2 AHP	49
Tabel 3.5 Hasil Pengujian 2 AHP	50
Tabel 3.6 Bobot Kriteria Pengujian 3 AHP	50
Tabel 3.7 Hasil Pengujian 3 AHP.....	51
Tabel 3.8 Bobot Kriteria Pengujian 4 AHP	51
Tabel 3.9 Hasil Pengujian 4 AHP.....	52
Tabel 3.10 Bobot Kriteria Pengujian 5 AHP	52
Tabel 3.11 Hasil Pengujian 5 AHP.....	53
Tabel 3.12 Bobot Kriteria Pengujian 6 AHP	53
Tabel 3.13 Hasil Pengujian 6 AHP.....	54
Tabel 4.1 Pengujian WSM	68
Tabel 4.2 Bobot Kriteria Pengujian 1 WSM	70
Tabel 4.3 Hasil Pengujian 1 WSM	70
Tabel 4.4 Bobot Kriteria Pengujian 2 WSM	71
Tabel 4.5 Hasil Pengujian 2 WSM	71
Tabel 4.6 Bobot Kriteria Pengujian 3 WSM	71
Tabel 4.7 Hasil Pengujian 3 WSM	72
Tabel 4.8 Bobot Kriteria Pengujian 4 WSM	72
Tabel 4.9 Hasil Pengujian 4 WSM	73
Tabel 4.10 Bobot Kriteria Pengujian 5 WSM	73
Tabel 4.11 Hasil Pengujian 5 WSM	74
Tabel 4.12 Bobot Kriteria Pengujian 6 WSM	74
Tabel 4.13 Hasil Pengujian 6 WSM	74

Tabel 5.1 Pengujian WPM	89
Tabel 5.2 Bobot Kriteria Pengujian 1 WPM	91
Tabel 5.3 Hasil Pengujian 1 WPM	91
Tabel 5.4 Bobot Kriteria Pengujian 2 WPM	92
Tabel 5.5 Hasil Pengujian 2 WPM	92
Tabel 5.6 Bobot Kriteria Pengujian 3 WSM	92
Tabel 5.7 Hasil Pengujian 3 WSM	93
Tabel 5.8 Bobot Kriteria Pengujian 4 WSM	93
Tabel 5.9 Hasil Pengujian 4 WSM	94
Tabel 5.10 Bobot Kriteria Pengujian 5 WSM	94
Tabel 5.11 Hasil Pengujian 5 WSM	94
Tabel 5.12 Bobot Kriteria Pengujian 6 WSM	95
Tabel 5.13 Hasil Pengujian 6 WSM	95
Tabel 6.1 Pengujian TOPSIS	111
Tabel 6.2 Bobot Kriteria Pengujian 1 TOPSIS	112
Tabel 6.3 Hasil Pengujian 1 TOPSIS	113
Tabel 6.4 Bobot Kriteria Pengujian 2 TOPSIS	114
Tabel 6.5 Hasil Pengujian 2 TOPSIS	114
Tabel 6.6 Bobot Kriteria Pengujian 3 TOPSIS	114
Tabel 6.7 Hasil Pengujian 3 TOPSIS	115
Tabel 6.8 Bobot Kriteria Pengujian 4 TOPSIS	115
Tabel 6.9 Hasil Pengujian 4 TOPSIS	116
Tabel 6.10 Bobot Kriteria Pengujian 5 TOPSIS	116
Tabel 6.11 Hasil Pengujian 5 TOPSIS	117
Tabel 6.12 Bobot Kriteria Pengujian 6 TOPSIS	117
Tabel 6.13 Hasil Pengujian 6 TOPSIS	118
Tabel 7.1 Perbandingan Metode AHP, WSM, WPM dan Topsis	119
Tabel 7.2 Perbandingan Metode AHP, WSM, WPM dan Topsis dengan Bobot Kriteria Sama	121
Tabel 7.3 Perbandingan Metode AHP, WSM, WPM dan Topsis dengan Bobot Kriteria Tidak sama	122
Tabel 7.4 Perbandingan Hasil Rangking Manual dengan Metode	124

ABSTRAK

Setiyono, Andreas. 2022. **SISTEM PENDUKUNG KEPUTUSAN PENILAIAN KINERJA GURU BERBASIS *MULTI CRITERIA DECISION MAKING***. Thesis. Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulan Malik Ibrahim Malang. Pembimbing: (1) Dr. M. Amin Hariyadi, MT (2) Dr. Sri Harini, M.Si

Kata Kunci : *Multi Criteria Decision Making*, Penilaian Kinerja Guru, Sistem Pendukung Keputusan.

Guru merupakan pendidik professional yang mempunyai peran dan tugas penting untuk mendidik dan mencerdaskan anak-anak penerus bangsa. Untuk itu dibutuhkan penilaian terhadap proses pembelajaran yang dilakukan oleh seorang guru. Penilaian tersebut dengan Penilaian Kinerja Guru (PKG). Dalam PKG terdapat 14 kriteria yang nilai. 14 Kriteria tersebut dikelompokkan menjadi kelompok pedagogic, kelompok kepribadian, kelompok sosial dan kelompok professional. Selama ini, PKG di beberapa sekolah masih dilakukan secara manual. Dalam penelitian ini dibuat Sistem Pendukung Keputusan (SPK) untuk proses PKG. SPK yang dibuat berbasis *Multi Criteria Decision Making* (MCDM). Metode MCDM yang digunakan adalah *Analytical Hierarchy Process* (AHP), *Weighted Sum Model* (WSM), *Weighted Product Model* (WPM), *Technique For Others Preference by Similarity to Ideal Solution* (TOPSIS). Metode-metode MCDM tersebut akan dibandingkan untuk mengetahui tingkat ketidaksesuaian posisi perbandingan dengan hasil penilaian manual. Hasil perbandingan tersebut digunakan untuk penentuan metode terbaik untuk digunakan pada Sistem Pendukung Keputusan Penilaian Kinerja Guru. Berdasarkan hasil pengujian dengan pembobotan kriteria yang berbeda menunjukkan bahwa WSM merupakan metode dengan tingkat ketidaksesuaian terkecil dan AHP adalah metode dengan tingkat ketidaksesuaian terbesar. Berikut ini adalah rata-rata tingkat ketidaksesuaian (jumlah alternatif adalah 25) antara metode : AHP = 15,1, WSM = 4,5, WPM = 10,8 dan TOPSIS 9,9.

ABSTRACT

Setiyono, Andreas. 2022. ***MULTI-CRITERIA DECISION MAKING SUPPORT SYSTEM FOR TEACHER PERFORMANCE ASSESSMENT*** Thesis. Master of Informatics Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang Advisors: (1) Dr. M. Amin Hariyadi, MT (2) Dr. Sri Harini, M.Sc.

Keywords: *multi-criteria decision making*, teacher performance assessment, and decision support system.

Teachers are professional educators who have an important role and task in educating the nation's future children. As a result, a teacher-led assessment of the learning process is required. The assessment is the Teacher Performance Assessment (PKG). PKG has 14 criteria that have been evaluated. 14 These criteria are grouped into pedagogic groups, personality groups, social groups, and professional groups. So far, PKG at several schools is still done manually. In this research, a decision support system (SPK) was created for the PKG process. SPK is based on *multi-criteria decision making* (MCDM). The MCDM method used is the *Analytical Hierarchy Process* (AHP), *Weighted Sum Model* (WSM), *Weighted Product Model* (WPM), and *Technique for Others' Preference by Similarity to the Ideal Solution* (TOPSIS). These MCDM methods will be compared to find out the degree of discrepancy between the ranking positions and the results of the manual assessment. The results of this comparison are used to determine the best method for use in the Teacher Performance Assessment Decision Support System. Based on the test results with different criteria weighting, it is clear that WSM has the lowest level of discrepancy and AHP has the highest level of discrepancy. Following are the average levels of non-conformance (the number of alternatives is 25) between the methods: AHP = 15.1, WSM = 4.5, WPM = 10.8, and TOPSIS = 9.9.

سيتيونو، أندرياس. 2022. نظام تقييم قرار المعايير المتعددة يعني قرار المعايير المتعددة. أطروحة. ماجستير في برنامج دراسة المعلومات كلية العلوم والتكنولوجيا بجامعة الدولة الإسلامية مولان مالك إبراهيم مالانج. المشرف: (1) د. م. أمين هريادي ، MT (2) د. سري هاريني ، م

الكلمات الرئيسية: معايير اتخاذ القرار المتعدد ، تقييم أداء المعلم ، نظام دعم القرار .

المعلمون هم معلمون محترفون لديهم أدوار ومهام مهمة لتعليم وإذلال الأطفال الخلف في البلاد. هذا يتطلب تقييم عملية التعلم التي يقوم بها المعلم. التقييم مع تقييم أداء المعلم (PKG). في PKG هناك 14 معيارًا هي القيم. 14 يتم ترطيب هذه المعايير في مجموعات تربوية ومجموعات شخصية ومجموعات اجتماعية ومجموعات مهنية. خلال هذا الوقت ، كان PKG لا يزال يتم يدويًا. في هذه الدراسة تم إنشاء نظام دعم القرار (SPK) لعملية PKG. يتم إجراء SPK بناءً على اتخاذ القرار متعدد المعايير (MCDM). طريقة MCDM المستخدمة هي عملية التسلسل الهرمي (AHP) ، نموذج المجموع المرجح (WSM) ، نموذج المنتج المرجح (WPM) ، تقنية للأخيرين تفضيلات بواسطة (TOPSIS). ستتم مقارنة طرق MCDM لمعرفة مستوى موقف عدم الاستبعاد مع نتائج التقييم اليدوي. يتم استخدام نتائج المقارنة لتحديد أفضل طريقة لاستخدامها في نظام دعم قرار تقييم أداء المعلم. استنادًا إلى نتائج الاختبارات بمعايير الترجيح المختلفة ، يُظهر WSM طريقة ذات أصغر مستوى من عدم التطابق و AHP هي الطريقة ذات أعلى مستوى من عدم التطابق. فيما يلي متوسط مستوى عدم الامتثال (عدد البدائل هو 25) بين الطرق: $AHP = 15$ ، 1 ، $WSM = 4.5$ ، $WPM = 10.8$ ، و $TOPSIS = 9.9$.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Guru merupakan pendidik yang harus bekerja secara profesional dalam menjalankan tugas mendidik dan mencerdaskan anak-anak penerus bangsa. Guru adalah elemen kunci pada sistem pendidikan, khususnya di Sekolah Dasar (SD). Untuk itu diperlukan proses penilaian untuk menjamin tugas mendidik dan mencerdaskan anak-anak tersebut dilakukan dengan baik dan berkualitas. Proses penilaian tersebut disebut dengan Penilaian Kinerja Guru (PKG).

PKG bertujuan untuk menilai kinerja guru dalam menjalankan tugas mendidik dan mencerdaskan telah baik dan memuaskan atau belum. Kinerja yang telah memenuhi standar dapat dikatakan bahwa kinerja tersebut telah baik. Hal itu dengan yang terdapat dalam Al Qur'an surat At Tawbah ayat 105:

وَقُلْ أَعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ وَسَتُرَدُّونَ إِلَىٰ عِلْمِ الْغَيْبِ وَالشَّهَادَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ

Dan katakanlah: "Bekerjalah kamu, maka Allah dan Rasul-Nya serta orang-orang mu'min akan melihat pekerjaanmu itu, dan kamu akan dikembalikan kepada (Allah) Yang Mengetahui akan yang ghaib dan yang nyata, lalu diberitakan-Nya kepada kamu apa yang telah kamu kerjakan".

Dalam ayat ini menjelaskan kinerja adalah hasil secara kualitas dan kuantitas dalam menjalankan tugas berdasarkan tanggung jawab yang dilakukan oleh seorang guru.

Kinerja guru merupakan hal yang terpenting dalam mewujudkan proses pendidikan yang efektif khususnya untuk menciptakan sikap disiplin dan meningkatkan kualitas hasil dari kegiatan belajar peserta didik. Wujud kinerja guru yang baik adalah terlaksananya tugas utama dan fungsi seorang guru. Wujud kinerja guru yang baik diukur melalui Penilaian Kinerja Guru (PKG). PKG dilakukan oleh penilai dan kepala sekolah berdasarkan instrumen pengarsipan dokumen pedoman pelaksanaan PKG untuk guru kelas atau guru mata pelajaran dimasing-masing lembaga, dimana kriteria penilaian yang digunakan adalah 4 (empat) kompetensi, yang pertama pedagogik dengan 7 (tujuh) sub kriteria, kompetensi kedua kepribadian dengan 3 (tiga) sub kriteria, kompetensi ketiga sosial dengan 2 (dua) sub kriteria dan kompetensi keempat profesional dengan 2 (dua) sub kriteria (Hartanto, 2019)

Saat ini, PKG di beberapa lembaga atau sekolah dasar masih dilaksanakan dan dijalankan secara manual dengan mengisi formulir berisi instrumen PKG serta tanpa metode saat melakukan penilaian. Untuk itu, sebagai upaya dalam mempercepat proses penilaian kinerja dibutuhkan sistem penilaian kinerja guru yang akan membantu kepala sekolah pada proses pengambilan keputusan PKG.

Untuk mengatasi permasalahan tersebut, sistem yang dibangun berupa Sistem Pendukung Keputusan (SPK). SPK merupakan sistem dengan tujuan untuk membantu seorang atau beberapa pimpinan dalam menyelesaikan masalah yang dihadapi. Dimana dalam menyelesaikan masalah tersebut maka harus diubah menjadi suatu model solusi (McLeod, 2007) (Schell, 2007). Untuk menemukan

sebuah alternatif terbaik terhadap permasalahan yang melibatkan lebih dari satu alternatif berdasarkan kriteria tertentu didalam sistem pendukung keputusan bisa menggunakan suatu metode *Multi Criteria Decision Making* (MCDM) (Kusumadewi, 2006).

MCDM adalah suatu metode yang banyak dipergunakan untuk menyelesaikan masalah pengambilan keputusan. Dalam menyelesaikan masalah pengambilan keputusan, MCDM melakukan dengan cara memilih pilihan terbaik berdasarkan kriteria-kriteria yang telah ditetapkan sebelumnya oleh pengambil keputusan. MCDM terdiri dari beberapa metode antara lain metode *Analitycal Hierarchy Process* (AHP), *Weighted Product Model* (WPM), *Weighted Sum Model* (WSM), *Technique For Others Preference by Similarity to Ideal Solution* (TOPSIS) dan lain-lain.

Metode-metode MCDM yang akan digunakan pada penelitian ini yaitu AHP, WSM, WPM dan TOPSIS. Dimana hasil dari metode-metode MCDM tersebut akan dibandingkan. Dalam melakukan perbandingan antar hasil metode MCDM tersebut, metode atau algoritma yang digunakan adalah *Hamming Distance*. Tujuan penggunaan *Hamming Distance* adalah untuk mengetahui tingkat ketidaksesuaian posisi pemilihan atau perangkingan alternatif hasil metode-metode MCDM dengan hasil perangkingan secara manual menggunakan formulir cetakan instrumen PKG.

Berdasarkan uraian diatas, untuk menentukan penilaian kinerja guru secara optimal berdasarkan dengan kriteria yang ditentukan, diperlukan penelitian pengembangan SPK memakai metode AHP, WSM, WPM dan TOPSIS. Hasil dari

metode-metode tersebut kemudian dibandingkan dan ditentukan metode terbaik untuk menyelesaikan masalah penilaian kinerja guru. Diharapkan sistem yang dikembangkan berdasarkan metode terbaik tersebut dapat menjadi alat pendukung keputusan bagi pihak terkait, khususnya bagi kepala sekolah dan penilai sebagai dasar pengambilan keputusan.

1.2. Pernyataan Masalah

Pernyataan masalah pada penelitian ini yaitu Bagaimana mengembangkan SPK penilaian kinerja guru menggunakan MCDM dengan membandingkan metode *Analitycal Hierarchy Process (AHP)*, *Weighted Product Model (WPM)*, *Weighted Sum Model (WSM)*, dan *Technique For Others Preference by Similarity to Ideal Solution (TOPSIS)*, sehingga menghasilkan penilaian kinerja guru yang paling baik ?

1.3. Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan SPK penilaian kinerja guru menggunakan metode *Analitycal Hierarchy Process (AHP)*, *Weighted Product Model (WPM)*, *Weighted Sum Model (WSM)*, dan *Technique For Others Preference by Similarity to Ideal Solution (TOPSIS)* untuk memperoleh keputusan penilaian kinerja guru dalam menentukan guru terbaik.

1.4. Manfaat Penelitian

Diharapkan penelitian ini akan memberikan manfaat untuk memberikan informasi tentang SPK penilaian kinerja guru untuk masalah penentuan perangkan dan penentuan jumlah jam mengajar.

1.5. Batasan Masalah

Adapun batasan masalah yang yang menjadi dasar batasan adalah

1. Menggunakan dataset pada penilaian kinerja guru di SD Anak saleh.
2. Keluaran yang dihasilkan oleh sistem hanya senagai alat bantu dalam mendukung pengambil keputusan penilaian kinerja guru kelas/ mata pelajaran dalam menentukan guru terbaik pada SD Anak Saleh Malang sebagai obyek penelitian.

1.6. Sistematika Penulisan

Sistematika penulisan yang digunakan yaitu :

BAB I PENDAHULUAN (*INTRODUCTION*)

Bab I ini merupakan bab pendahuluan yang terdiri dari latar belakang, pernyataan masalah, tujuan, batasan masalah, dan sistematika penulisan. Isi pada bab pendahuluan ini akan berisi gambaran tentang maksud dan tujuan dilakukannya penelitian ini, yang jika disampaikan secara singkat adalah mengembangkan SPK untuk penilaian kinerja guru dengan menerapkan metode AHP, WSM, WPM dan TOPSIS untuk membantu dalam membuat keputusan pada kasus penilaian kinerja guru.

BAB II STUDI PUSTAKA (*LITERATUR REVIEW*)

Pada Studi Pustaka mengemukakan teori-teori dan karya ilmiah yang yang berkaitan terhadap proses serta metode yang dipakai. Dimana teori tentang proses serta metode yang dipakai tersebut didapatkan dari beberapa sumber referensi yang valid.

BAB III s/d N. METODOLOGI PENELITIAN (*RESEARCH METHODOLOGY*)

Pada bab metodologi penelitian berisi berbagai hal mulai dari pola dan rancangan, materi atau bahan, alat, proses, sampai pada kegiatan analisis hasil penelitian.

BAB N+1. PEMBAHASAN (*DISCUSSION*)

Bab pembahasan berisi hasil yang didapatkan. Dimana hasilnya bisa dalam bentuk penjelasan teoritis yang bersifat kualitatif, kuantitatif atau bahkan secara statistik. Hasil uji coba pada metode-metode utama akan dijabarkan dan dilakukan perbandingan dengan hasil uji coba metode utama lain. Penjabaran tersebut dilakukan melalui perpektif informatika serta diinspirasi Al-Qur'an dan Hadist.

BAB N+2. KESIMPULAN (*CONCLUSION*)

Pada bab ini akan dijelaskan tentang kesimpulan penelitian dan berisikan saran-saran yang disampaikan untuk perbaikan serta hal yang dapat dilakukan untuk penelitian selanjutnya.

BAB II

STUDI PUSTAKA (*LITERATUR REVIEW*)

2.1. Penilaian Kinerja Guru

Penilaian Kinerja Guru (PKG) merupakan penilaian terhadap poin-poin kegiatan tugas utama yang harus dilaksanakan oleh guru sebagai langkah dalam pembinaan karir, kepangkatan, dan jabatannya. Pengertian PKG tersebut adalah pengertian sesuai yang terdapat pada Peraturan Menteri Negara Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 16 Tahun 2009. Dalam melaksanakan tugas utamanya, guru harus mempunyai penguasaan dan penerapan pengetahuan yang baik serta memiliki keterampilan dalam melaksanakan tugas utama tersebut. Hal itu sesuai dengan kompetensi yang dibutuhkan oleh seorang guru sebagaimana tertuang dalam Peraturan Menteri Pendidikan Nasional Nomor 16 Tahun 2007 tentang Standar Kualifikasi Akademik dan Kompetensi Guru. Kompetensi yang berupa penguasaan dan penerapan pengetahuan yang baik serta memiliki keterampilan tersebut akan sangat berpengaruh dalam pencapaian kualitas proses pembelajaran pada siswa, dan akan sangat membantu dalam melaksanakan tugas tambahan di sekolah.

Sistem PK Guru merupakan sistem yang ditujukan untuk menilai mengenali kemampuan guru saat menjalankan tugas dalam unjuk kerja dengan cara mengukur kemampuan melaksanakan kompetensi.

Data-data yang dipakai Tim Direktorat Profesi Pendidik Dirjen PMPTK Tahun 2010 sebagai berikut (PMPTK, 2010):

- a. Instrumen PKG
- b. Lembar pernyataan yang berisi cara menilai, indikator, dan kompetensi PK Guru Kelas/Mata Pelajaran (Tim Direktorat Profesi Pendidik Dirjen PMPTK, 2010).

Tabel 2.1 Tabel Kompetensi Penilaian Kinerja Guru

Kompetensi	Cara Menilai
Pedagogik	
1. Menguasai karakteristik peserta didik.	Pengamatan & Pemantauan
2. Menguasai teori belajar dan prinsip-prinsip pembelajaran yang mendidik.	Pengamatan
3. Pengembangan kurikulum	Pengamatan
4. Kegiatan pembelajaran yang mendidik.	Pengamatan
5. Pengembangan potensi peserta didik.	Pengamatan & Pemantauan
6. Komunikasi dengan peserta didik.	Pengamatan
7. Penilaian dan evaluasi	Pengamatan
Kepribadian	
8. Bertindak sesuai dengan norma agama, hukum, sosial dan kebudayaan nasional.	Pengamatan & Pemantauan
9. Menunjukkan pribadi yang dewasa dan teladan.	Pengamatan & Pemantauan
10. Etos kerja, tanggung jawab yang tinggi, rasa bangga menjadi guru.	Pengamatan & Pemantauan
Sosial	
11. Bersikap inklusif, bertindak obyektif serta tidak diskriminatif.	Pengamatan & Pemantauan
12. Komunikasi dengan sesama guru, tenaga kependidikan, orang tua, peserta didik dan masyarakat.	Pengamatan
Profesional	
13. Penguasaan materi, struktur, konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu.	Pengamatan
14. Mengembangkan keprofesional melalui tindakan yang reflektif.	Pengamatan

Keterangan

Pengamatan yaitu tindakan guna menilai kinerja guru dengan cara diskusi sebelum dilakukan pengamatan, dimana pengamatan dilakukan pada saat menjalankan proses pembelajaran, dan akan dilakukan diskusi kembali setelahnya.

Pemantauan yaitu tindakan guna menilai kinerja guru dengan cara memeriksa dokumen, melakukan tanya jawab dengan guru tersebut, serta tanya jawab dengan warga sekolah.

2.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) adalah sistem untuk menyajikan informasi, model, dan dapat manipulasi data, dimana sistem tersebut dibuat interaktif. Sistem tersebut berguna untuk memberi saran dalam mengambil keputusan pada masalah semiterstruktur dan tidak terstruktur (Turban, 2005).

SPK merupakan sistem yang memiliki dan mampu memecahkan masalah dan mampu mengkomunikasikan masalah dalam kondisi semistruktural dan tak terstruktur. Sistem ini dapat digunakan pada situasi semi terstruktur dan situasi tidak terstruktur untuk membantu pengambilan keputusan dan tidak ada orang yang mengetahui dengan pasti keputusan yang benar dan seharusnya dilakukan (Daihani, 2001).

Sprague dan Watson mengutarakan bahwa SPK harus mempunyai lima karakteristik utama sebagai sistem yaitu (Sprague, 1993):

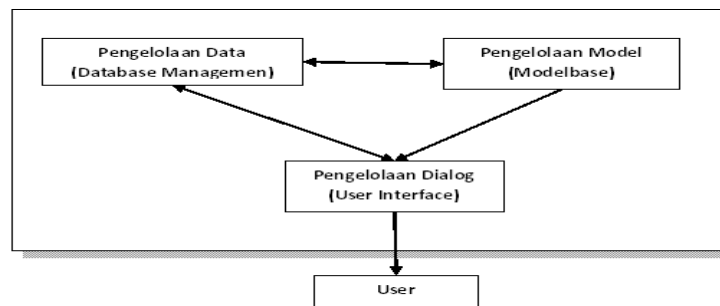
- a. Berbasis pada komputer

- b. Digunakan guna memberi saran untuk membantu para pengambil keputusan
- c. Digunakan untuk mencari solusi pada masalah-masalah rumit dan tidak mungkin dapat diselesaikan melalui kalkulasi manual
- d. Mencarian solusi dilakukan dengan cara mensimulasikan dengan interaktif
- e. Komponen utama adalah berupa data dan model analisis

Tujuan SPK dibuat adalah (Kursini, 2007):

- a. Untuk menyelesaikan masalah semi terstruktur sehingga dapat memberikan saran kepada manajer untuk mengambil keputusan.
- b. Tidak ditujukan untuk mengganti peran manajer, tetapi hanya sebagai hal pendukung dari keputusan yang akan diambil manajer.
- c. Menambah efektifitas dari manajer dalam mengambil keputusan dan lebih baik daripada hanya meningkatkan efektivitas.
- d. Dapat melakukan komputasi dengan lebih cepat dan biaya rendah karena kecepatan komputasi komputer.

SPK terdiri dari 3 komponen besar. 3 komponen besar itu adalah *database management, model base* dan *software system / user interface*.



Gambar 2.1 Tiga Komponen Besar SPK

2.2.1. Database Management

Subsistem *Database Management* adalah yang terkoneksi dengan basis data. Data datanya sendiri bisa dari dalam maupun luar lingkungan. SPK membutuhkan data yang relevan sesuai masalah yang akan dicari solusinya.

2.2.2. Model Base

Subsistem *Model Base* berupa model yang menggambarkan masalah menjadi format kuantitatif (sebagai contoh model matematika) untuk menjadi dasar simulasi atau pengambilan keputusan, yang terdiri dari tujuan masalah (objektif), komponen terkait, batasan dan hal-hal yang berhubungan lainnya. Dengan subsistem *Model Base*, pengambil keputusan dapat melakukan analisa dengan menyeluruh dengan cara pengembangan dan perbandingan alternatif yang menjadi solusi.

2.2.3. User Interface

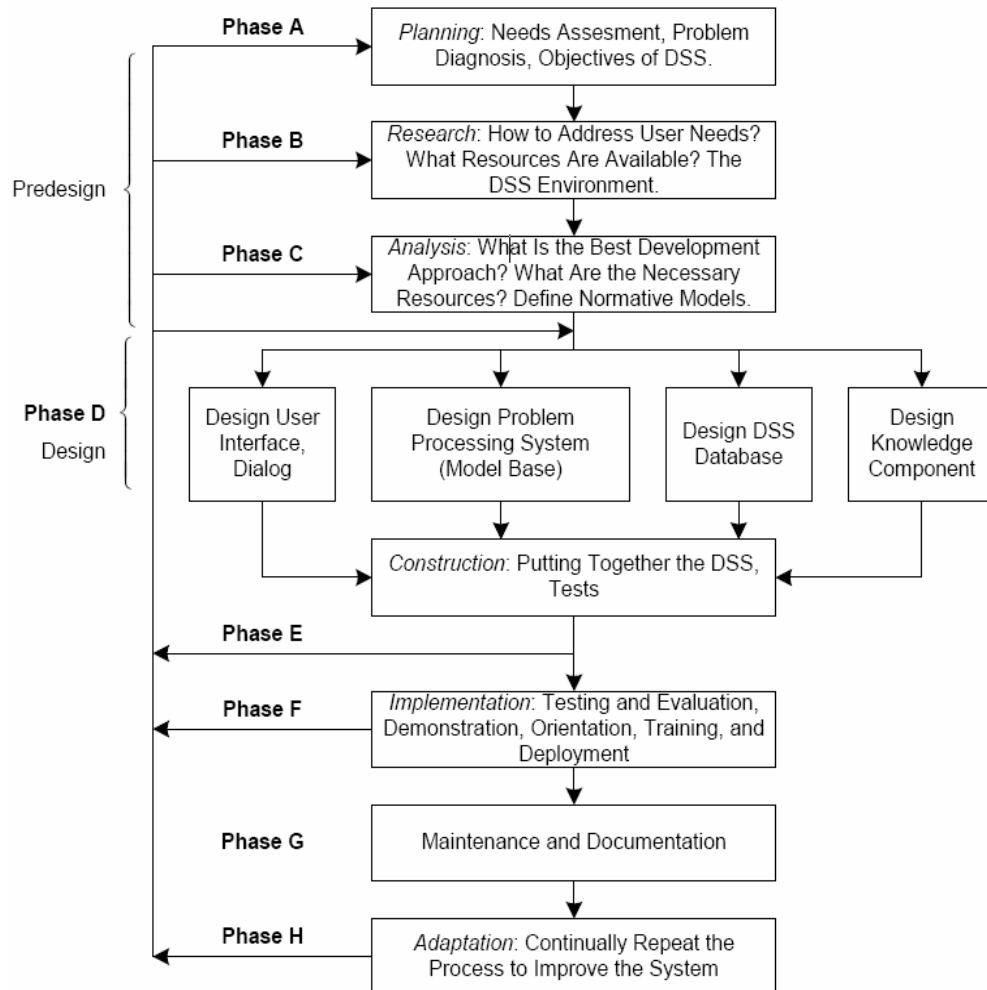
Subsistem *User interface* disebut juga dengan subsistem dialog. Subsistem ini yang menghubungkan subsistem *Database Management* dan Subsistem *Model Base*, yang sebelumnya telah digambarkan dalam bentuk model dan dimengerti oleh komputer. Subsistem ini berfungsi untuk menampilkan keluaran dari sistem serta untuk media melakukan inputan ke dalam SPK.

2.2.4. Pengembangan Sistem Pendukung Keputusan

Proses pengembangan SPK tersaji dalam bagan fase-fase pengembangan dalam Gambar 2.2.

Gambar 2.2 menjelaskan proses pengembangan SPK yang meliputi meliputi:

1. Fase pra-desain yang terdiri dari perencanaan, penelitian, dan analisis. Dalam perencanaan meliputi perencanaan kebutuhan, diagnosis masalah, dan menentukan tujuan. Dalam fase penelitian meliputi penelitian terhadap apa keinginan user, sumber daya yang dapat digunakan, dan lingkungan di mana SPK akan diimplementasikan. Dalam analisis meliputi analisis pendekatan yang akan digunakan dan model yang sesuai.
2. Fase desain meliputi desain *user interface*, desain proses, desain database, dan desain *knowledge base* yang akan digunakan.
3. Fase konstruksi meliputi integrasi menggunakan tool yaitu dengan menterjemahkan desain ke dalam bahasa pemrograman yang dipilih.
4. Fase implementasi meliputi pengujian dan evaluasi, demonstrasi, pelatihan, dan pemasangan.
5. Fase pemeliharaan dan dokumentasi.
6. Fase adaptasi, adalah fase pembaharuan yang dilakukan sebagai tanggapan atas perubahan yang diinginkan pengguna.



Gambar 2.2 Fase-fase Pengembangan SPK (Subakti, 2002)

2.3 Multi Criteria Decision Making (MCDM)

Multi-criteria decision making (MCDM) adalah metode untuk mengambil keputusan berdasarkan alternatif-alternatif yang ada. Terdapat dua macam kategori dari MCDM, antara lain (Raharjo, 2000):

1. *Multiple Objective Decision Making* (MODM) untuk memecah permasalahan perencanaan atau perancangan. Pemecahan masalah dilakukan dengan teknik matematika berupa optimasi. Pemecahan

masalah dengan metode ini biasanya dilakukan untuk jumlah alternatif yang sangat banyak atau bahkan tak terhingga.

2. *Multiple Attribute Decision Making* (MADM) untuk memecahkan masalah yang berhubungan dengan pemilihan. Pada memecahkan masalah ini, tidak banyak memerlukan teknik analisa matematika dan biasanya digunakan pada pemilihan yang memiliki jumlah alternatif yang kecil.

Dalam hal ini *Analytical Hierarchy Process* (AHP), *Weighted Sum Model* (WSM), *Technique For Others Preference by Similarity to Ideal Solution* (TOPSIS) dan *Weighted Product Model* (WPM) termasuk kedalam MADM.

2.4 Metode Analytical Hierarchy Proses (AHP)

Metode ini dibuat diawal 1970. Metode ini dibuat oleh Prof. Thomas Lorie Saaty. Beliau bersal dari Wharton Business School. Metode ini dibuat untuk dapat mencari peringkat, prioritas urutan atau perangkingan dari alternatif-alteratif solusi untuk masalah yang akan dipecahkan (Sinaga, 2009). Prinsip dari AHP merupakan teori umum yang berhubungan dengan pengukuran, dimana pengukuran tersebut dibutuhkan untuk memutuskan skala rasio pada saat dilakukan saat perbandingan berpasangan pada kasus distrit ataupun kontinu. Prinsip-prinsip dalam memecahkan masalah menggunakan AHP adalah (Saaty, 2000)

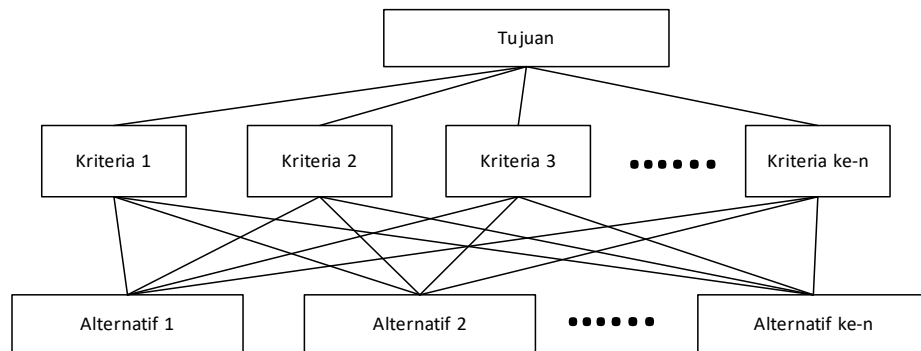
1. *Decomposition* (diselesaikan dalam bentuk hierarki) merupakan prinsip yang menjelaskan bahwa dengan memisahkan elemen-elemen menjadi

lebih kecil dan bisa dipahami dengan mudah maka sistem yang besar dan rumit dapat dipahami dengan lebih gampang.

2. *Comparative judgment* (menilai alternatif dan menilai kriteria) merupakan prinsip yang menjelaskan bahwa perbandingan berpasangan dilakukan untuk menilai alternatif dan menilai kriteria, sehingga skala perbandingan pada masing-masing kriteria dapat diketahui. Untuk aturan penggunaan skala perbandingan dapat dilihat pada Tabel 2.2.
3. *Synthesis of priority* (dengan cara memutuskan prioritas)
4. *Logical Consistency* (dapat diartikan sebagai konsistensi logis)

Langkah-langkah penyelesaian pengambilan keputusan menggunakan AHP yaitu (Saaty, 2000):

1. Menjelaskan permasalahan diikuti dengan menentukan pemecahan masalah yang ingin dicapai, sehingga dapat digunakan untuk membuat hierarki pemecahan masalah. Penyusunan hierarki dilakukan dengan mengisi level teratas dengan tujuan dan mengisi level-level dibawahnya dengan kriteria-kriteria sebagai indikator penilaian, setiap kriteria bisa memiliki subkriteria dibawahnya (Saaty, 1980). Contoh struktur hierarki AHP bisa dilihat pada Gambar 2.3.



Gambar 2.3 Struktur Hierarki AHP

2. Membuat prioritas pada setiap elemen dengan membuat matriks perbandingan berpasangan.
 - a. Membandingkan elemen dengan cara berpasangan berdasarkan kriteria yang diharapkan dengan menjadikannya matriks perbandingan berpasangan.
 - b. Mengisi matriks perbandingan berpasangan dengan angka sesuai tingkat kepentingan relatif satu elemen dengan elemen lainnya. Penentuan angka kepentingan relatif sesuai dengan skala perbandingan pada Tabel 2.2. dibawah ini.

Tabel 2.2 Skala Penilaian Perbandingan Berpasangan

Intensitas Kepentingan	Artinya	Keterangan
1	Sama penting (<i>Equal Importance</i>)	Elemen memiliki tingkat kepentingan yang sama
3	Sedikit lebih penting (<i>Weak importance of one over</i>)	Elemen yang satu sedikit lebih penting dari pasangannya
5	Lebih penting (<i>Essential or strong importance</i>)	Elemen yang satu lebih penting dibandingkan dengan pasangannya
7	Sangat penting	Satu elemen jelas lebih

Intensitas Kepentingan	Artinya	Keterangan
	<i>(Demonstrated importance)</i>	mutlak penting daripada elemen lainnya
9	mutlak lebih penting <i>(Extreme importance)</i>	Satu elemen mutlak penting daripada elemen lainnya
2, 4, 6, 8	<i>Intermediate values between the two adjacent judgements</i>	Nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan
Resiprokal	Kebalikan	Jika elemen i memiliki salah satu angka diatas ketika dibandingkan elemen j, maka j memiliki kebalikannya ketika dibanding elemen i

3. Menghitung nilai prioritas (Sintesis)

Untuk mendapatkan keseluruhan prioritas dilakukan pertimbangan pada perbandingan berpasangan disintesis. Dimana Langkah-langkah untuk melakukannya yaitu :

- a. Nilai-nilai pada matriks dijumlahkan.
- b. Menghitung normalisasi matriks dengan cara membagi nilai-nilai pada kolom dengan jumlah total nilai pada kolom.
- c. Menghitung nilai rata-rata dengan cara menambahkan nilai-nilai dalam satu baris dengan jumlah elemen.

4. Menghitung konsistensi

Untuk menghitung konsistensi, Langkah-langkah yaitu :

- a. Nilai dikolom satu dikalikan dengan nilai prioritas relatif yang dimiliki elemen pertama, nilai dikolom kedua dikalikan dengan nilai prioritas relatif yang dimiliki elemen kedua dan seterusnya.
- b. Melakukan penjumlahan pada tiap baris.

- c. Membagi hasil langkah sebelumnya (penjumlahan baris) dengan prioritas relatif.
- d. Menghitung λ maks dengan cara menjumlahkan hasil bagi diatas dengan jumlah elemen.
- e. Menghitung *Consistency Index* (CI), dimana rumus yang digunakan adalah:

$$CI = \frac{(\lambda \text{ maks} - n)}{n-1} \quad (2.1)$$

Keterangan:

CI = *Consistency Index*

n = banyaknya elemen

λ maks = *Eigen value*

- f. Menghitung *Consistency Ratio* (CR), dimana rumus yang digunakan adalah :

$$CR = \frac{CI}{IR} \quad (2.2)$$

Keterangan:

CR = *Consistency Ratio*

CI = *Consistency Index*

IR = *Indeks Random Consistency*

Nilai IR ditentukan dari ukuran matriks yang akan dibandingkan. Nilai

IR tersebut dapat dilihat pada Tabel 2.3.

Tabel 2.3 *Indeks Random Consistency (IR)* berdasarkan ukuran matriks

Ukuran Matriks (n)	Indeks Random /IR (inkonsistensi)
2	0
3	0.58
4	0.9
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49

Hasil dikatakan konsisten jika nilai CR yang diperoleh lebih kecil atau sama dengan 0,1 ($CR \leq 0,1$), sedangkan jika melebihi nilai tersebut maka hasil dikatakan tidak konsisten dan perhitungan perbandingan berpasangan harus diulang.

2.5 Metode Weighted Sum Model (WSM)

Weighted Sum Model (WSM) adalah metode yang untuk menyelesaikan masalah pengambilan keputusan mutli-kriteria yang paling dikenal dan paling sederhana. WSM digunakan untuk melakuka penilaian terhadap alternatif berdasarkan kriteria keputusan. Alternatif yang memilki penilaian terbaik adalah yang memiliki hasil total nilai kinerja maksimal (Rizka, 2015). WSM adalah salah satu bagian dari MCDM yang melakukan evaluasi nilai terhadap setiap alternatif. WSM banyak dipakai untuk menyelesaikan masalah dimensi tunggal. Pada WSM, perhitungan skor dari alternatif dilakukan mengalikan jumlah alternatif dengan

bobot untuk tiap kriteria (Irwansyah, 2022). Langkah-langkah pengambilan keputusan dengan metode WSM adalah (F. Parhusip, 2018):

1. Mengidentifikasi kriteria dan Alternatif.
2. Melakukan perhitungan Nilai WSM-Score, dimana dilakukan dengan rumus:

$$A_i^{WSM-Score} = \sum_{j=1}^n w_j x_{ij} \quad (2.3)$$

Keterangan

n = Jumlah kriteria

W_j = Bobot dari setiap kriteria

X_{ij} = Nilai matrks x

3. Langkah terakhir adalah dilakukan perangkingan

2.6 Metode Weighted Product Model (WPM)

Weighted Product Model (WPM) merupakan metode MCDM yang dalam memecahkan masalah dilakukan dengan perkalian dalam mengaitkan rating atribut, sebelum melakukan perkalian rating atribut harus dilakuukan pemangkatan dengan bobot atribut terlebih dahulu (Kusumadewi, 2006). Langkah tersebut serupa dengan langkah normalisasi. Untuk menggunakan WP dalam SPK dengan tujuan melakukan perangkingan alternatif, langkah pertama yang harus dijalankan adalah input nilai alternatif, setelah itu dilakukan pemrosesan pada bobot kriteria, memangkatkan vektor sehigga didapatkan hasil keputusan alternatif.

Langkah-langkah pengambilan keputusan dengan metode WPM adalah (Kusumadewi, 2006):

1. Memasukkan parameter penilaian yaitu berupa kriteria.
2. Dilakukan normalisasi pada bobot melalui pencarian nilai relatif bobot awal (W_j). Dimana tingkat kepentingan relatif pada setiap kriteria ditunjukkan oleh nilai bobot awal (W_0). W_0 tersebut kemudian dilakukan normalisasi dengan tujuan agar total nilai relatif bobot awal $\sum W_j = 1$. Rumus normalisasi yang digunakan adalah rumus 2.4.

$$W_j = \frac{W_j}{\sum W_j} \quad (2.4)$$

3. Menghitung nilai preferensi pada tiap alternatif A_i , langkah ini bertujuan untuk mendapatkan vektor S . Untuk menghitung nilai preferensi pada alternatif A_i langkah pertama yang dilakukan adalah pada kriteria ke j (x_{ij}) dilakukan penginisialisasian nilai rating ke- i . Langkah pertama tersebut akan menghasilkan nilai rating kinerja pada setiap kandidat. Dimana nilai rating kinerja tersebut yang akan dilakuakn pemangkatan dengan nilai relative bobot w_j (yang merupakan hasil langkah ke 2). Untuk atribut keuntungan (benefit), w_j bernilai positif. Untuk atribut biaya (cost), w_j bernilai negatif. Rumus 2.5 adalah rumus yang digunakan untuk perhitungan nilai preferensi ditiap alternatif A_i (vektor S).

$$S_i = \prod_{j=1}^n x_{ij}^{w_j} \quad (2.5)$$

Keterangan:

S = Preferensi alternatif atau vektor S

W = Bobot kriteria/subkriteria

X = Nilai kriteria

n = Banyaknya kriteria

j = Kriteria

i = Alternatif

4. Untuk mendapat nilai vektor V, yang perlu dilakukan adalah menghitung nilai preferensi relative di tiap alternatif. Untuk melakukan perhitungan tersebut rumus yang digunakan adalah rumus 2.6

$$V_i = \frac{\prod_{j=1}^n w_{ij}^{w_j}}{\prod_{j=1}^n (x_{ij}^*)^{w_j}} \quad (2.6)$$

Keterangan

V = Preferensi alternatif atau vektor V

W = Bobot kriteria/subkriteria

X = Nilai Kriteria

n = Banyaknya kriteria

i = Alternatif

j = Kriteria

5. Langkah terakhir yang perlu dilakukan yaitu dilakukan pembagian nilai V pada tiap alternatif berdasarkan pada ketentuan nilai standar ($V(A^*)$), dimana hasilnya adalah R. Untuk mendapatkan keputusan, maka perlu dilakukan pengurutan alternatif terbaik.

2.7 Metode Technique for Other Reference By Similarity to Ideal Solution (TOPSIS)

Hwang dan Yoong yang memperkenalkan *Technique for Order Preference by Similarity to an Ideal Solution* (TOPSIS) ini. TOPSIS termasuk kedalam *Multi Criteria Decision Making* (MCDM), dimana metode ini dapat membantu memberi saran pengambilan keputusan untuk masalah multi-kriteria. Oleh karena, TOPSIS sering dipakai dalam memecah masalah pengambilan keputusan dibanyak permasalahan. Konsep pemilihan alternatif terbaik pada TOPSIS ditentukan oleh jarak terjauh untuk solusi ideal negative (A-) dan jarak terpendek untuk solusi ideal positif (A+). Berikut ini adalah tahapan perhitungan menggunakan TOPSIS (Kursini, 2007):

1. Nilai atribut dilakukan normalisasi terlebih dahulu dengan menggunakan rumus 2.7 sehingga dapat terbentuk matriks ternormalisasi (R).

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.7)$$

Dengan $i=1,2,\dots, n$; dan $j=1,2,\dots, m$

2. Membentuk matriks ternormalisasi terbobot (Y) dengan cara mengalikan bobot dan nilai ditiap atribut dengan memakai rumus 2.8.

$$y_{ij} = w_i r_{ij} \quad (2.8)$$

3. Menyusun matrik solusi ideal positif (A⁺) dan negatif (A⁻) dengan memakai rumus dibawah ini

$$A^+ = y_1^+, y_2^+, \dots, y_j^+ \quad (2.9)$$

$$A^- = y_1^-, y_2^-, \dots, y_j^- \quad (2.10)$$

Dengan ketentuan

$$y_j^+ = \begin{cases} \max_i y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \min_i y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

$$y_j^- = \begin{cases} \min_i y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \max_i y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

4. Mencari selisih jarak antara nilai setiap alternatif dan matriks solusi ideal positif dengan menggunakan rumus 2.9. Kemudian Mencari selisih jarak antara nilai setiap alternatif dan matriks solusi ideal negatif dengan menggunakan rumus 2.10.

$$D_i^+ = \sqrt{\sum_{j=1}^m (y_j^+ - y_{ij})^2} \quad (2.9)$$

$$D_i^- = \sqrt{\sum_{j=1}^m (y_{ij} - y_j^-)^2} \quad (2.10)$$

5. Mencari nilai preferensi yang dimiliki oleh tiap alternatif. Nilai preferensi merupakan jarak terdekat suatu alternatif dengan / antara solusi ideal. Untuk tiap alternatif (V_i), Nilai preferensi didapat dengan rumus 2.11.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad (2.11)$$

Alternatif A_i yang akan lebih menjadi pilihan adalah yang memiliki nilai V_i lebih besar.

2.8 Perhitungan Ketidaksesuaian Perangkingan

Algoritma *Hamming Distance* dapat digunakan untuk menghitung ketidaksesuaian perangkingan. Algoritma *Hamming Distance* adalah algoritma *Approximate String* yang dikembangkan oleh Richar Hamming ditahun 1950 (S. H. Andreas Budiman, 2016). Algoritma ini awalnya digunakan untuk mengetahui jumlah ketidaksesuaian urutan dua buah biner, dimana biner yang dibandingkan tersebut memiliki panjang atau jumlah yang sama (S. H. Andreas Budiman, 2016). Jumlah ketidaksesuaian yang semakin kecil pada hasil algoritma *hamming distance* menandakan bahwa dua buah biner memiliki kemiripan yang semakin besar. Contoh penerapan *hamming distance* (Estining N. S. P., 2013):

- H (11111, 00000), menghasilkan *hamming distance* sebesar 5 dikarenakan tidak ada string yang sama dan jumlah string adalah 5.
- H (11000, 00010), menghasilkan *hamming distance* sebesar 3 dikarenakan string pada posisi 1, 2 dan 4 berbeda.

Jarak Hamming adalah metrik yang menyatakan jarak antara dua objek dengan jumlah ketidaksesuaian di antara pasangan variabel. Jumlah ketidaksesuaian yang semakin kecil pada hasil algoritma *hamming distance* menandakan bahwa dua buah biner memiliki kemiripan yang semakin besar.

Contoh penerapan *hamming distance*:

		• $H(11000, 00010) = 3$		
1	1	0	0	0
0	0	0	1	0
Tidak sama	Tidak sama		Tidak sama	

- Jumlah ketidaksesuaian adalah 3 dikarenakan string pada posisi 1, 2 dan 4 berbeda.

2.9 Penelitian Terkait

Penelitian-penelitian yang telah ada baik yang menggunakan metode sama maupun yang relevan dengan penelitian ini terdapat di Tabel 2.4:

Tabel 2.4 Daftar penelitian yang relevan

No	Objek Penelitian	Peneliti	Metode	Hasil
1	Penentuan Lokasi Pemasaran Produk dengan Menerapkan Metode AHP dan Weighted Product	I Wayan Kayun Suwastika, Ni Luh Ayu Kartika Yuniastari Sarja, 2016	AHP dan WP	Dari kedua metode yaitu AHP dan WP didapat hasil bahwa pada AHP alternatif lokasi yang terpilih adalah lokasi 1 (A1) sedangkan pada WP alternatif lokasi yang terpilih adalah lokasi 5 (A5). Dengan nilai AHP untuk lokasi 1 adalah 0.30841836, dan nilai WP untuk lokasi 5 adalah 0.23597
2	Sistem Pemilihan Perumahan Dengan Metode Kombinasi Fuzzy C-Means Clustering Dan Simple Additive Weighting	Tri Sandhika Jaya, Tesis, 2012	Fuzzy C-Means Clustering Dan SAW	SPK ini dapat digunakan untuk bantuan memilih perumahan dengan mudah dan cepat. Dari 10 kali pengujian menghasilkan 9 kali sesuai dengan hasil yang diinginkan dan 1 yang tidak sesuai.
3	Pengaruh Metode Evaluasi Penawaran Pengadaan Barang/Jasa Pemerintah Terhadap Hasil Pekerjaan Dengan Pendekatan Analytical Hierarchy	Dwi Sukarme, Tesis, 2011	AHP	Perbedaan karakteristik sistem gugur dengan sistem nilai terdapat pada sistem nilai (daftar simak dan <i>passing grade</i> /ambang lulus) memiliki akibat lebih bagus dari pada sistem gugur disaat melakukan pelelangan bahkan sampai berakhirnya pekerjaan. Pada <i>passing grade</i> /ambang lulus dan

No	Objek Penelitian	Peneliti	Metode	Hasil
	Process			daftar simak, faktor yang dominan yaitu faktor keuangan (55,1%), teknis (18,4%), administrasi (6,7%), pernyataan minat (6,7%), pakta integritas (6,5%), dan kualifikasi (6,7%); 3) Masih terdapat kesalahan dalam melaksanakan pekerjaan sebagaimana yang ditunjukkan pada saat validasi data.
4	Implementasi Metode AHP Dan WP Dalam Sistem Pendukung Keputusan Pemilihan Susu Formula Balita	Robby Rachmatullah, Heribertus Ary Setyadi, 2015	AHP dan WP	SPK ini dibuat dengan tujuan memberi saran pada kasus orang tua yang akan menentukan susu untuk anak
5	Sistem Pendukung Keputusan Penilaian Kinerja Guru SMA Menggunakan Metode Analytical Hierarchy Process (AHP) Dan Weighted Product (WP)	Wilman Alfarezi, Ariefiandi, Gunawan Abdillah, Ridwan Ilyas, 2017	AHP dan WP	SPK ini dapat menjadi solusi untuk menyelesaikan kasus penentuan guru dengan kinerja terbaik dan dapat digunakan untuk mengganti cara manual yang selama ini digunakan pihak sekolah.
6	Penerapan Metode Analytical Process (AHP) Pada Sistem Pendukung Keputusan Pemilihan Laptop	Sylvia Hartati Saragih, 2013	AHP	AHP adalah metode pada SPK yang dapat memecahkan masalah penentuan keputusan multikriteria dan bisa diterapkan untuk menyelesaikan masalah penentuan laptop yang akan dipilih.
7	Implementasi Metode Simple Additive Weighting (SAW) Untuk Pemilihan Pekerjaan Bidang Informatika	Tonni Limbong, 2014	SAW	SAW menghitung bobot tiap alternatif, nilai kriteria dibobotkan kemudian dilakukan pembagian dengan bidang pekerjaan yang telah terhubung dengan nilai akademik.
8	Penerapan Metode Weighted Product Model Untuk Seleksi Calon Karyawan	Sri Lestari, 2013	WP	WPM bisa dipakai dalam memecahkan kasus multidemensi yang salah satunya adalah penyeleksian calon karyawan, seperti pada penelitian ini.
9	Metode Weighted Product (WP) Dalam Sistem Pendukung Keputusan	Basri, 2017	WP	Dengan pengembangan SPK dalam masalah pemberian beasiswa ini bisa sebagai alat bantu untuk pihak terkait dilingkungan sekolah saat

No	Objek Penelitian	Peneliti	Metode	Hasil
	Penerimaan Beasiswa Prestasi			menentukan pemberian beasiswa berprestasi, dan dapat dengan mudah menentukan siswa yang pantas mendapatkan beasiswa berprestasi serta memudahkan dalam membuat validasi terhadap keaslian laporan

2.10 Penilaian Kinerja Guru dalam Perspektif Al-Qur'an

Islam memiliki sudut pandang bahwa kinerja mempunyai makna yaitu kemauan dan kesungguhan untuk menjalankan tugas. Hal ini seperti yang terdapat pada surat At-Taubah 105 :

Artinya: Dan katakanlah: "Bekerjalah kamu, maka Allah dan Rasul-Nya serta orang-orang mu'min akan melihat pekerjaanmu itu, dan kamu akan dikembalikan kepada (Allah) Yang Mengetahui akan yang ghaib dan yang nyata, lalu diberitakan-Nya kepada kamu apa yang telah kamu kerjakan". (QS, Taubah: 105).

Sedangkan surat Al-Maidah ayat 35 memuat tentang kinerja yaitu:

Artinya: Hai orang-orang yang beriman, bertakwalah kepada Allah dan carilah jalan yang mendekatkan diri kepada-Nya, dan berjihadlah pada jalan-Nya, supaya kamu mendapat keberuntungan. (QS. Al-Maidah: 35).

Islam mempunyai rambu-rambu untuk ummatnya yaitu saat menjalankan dan melakukan pekerjaan dituntut untuk berusaha dengan sungguh-sungguh, hal itu adalah hal yang mutlak. Sungguh-sungguh dalam menjalankan pekerjaan termasuk jihad. Orang yang memiliki kesungguhan saat bekerja, tidak hanya dilihat oleh manusia, bahkan Allah SWT juga menghargainya sebagai orang yang mulia atas prestasi kerjanya serta memperoleh kemuliaan.

Berdasarkan riwayat Abu Daud, dari Umar Ra, Nabi SAW bersabda:

Artinya: "Dari Umar Ibn al-Khaththab ra., dia berkata: Rasulullah telah bersabda bahwa amal-amal (itu sah bila disertai dengan niat). Dan bahwa bagi setiap orang (mendapatkan apa yang diniatkan). Maka barang siapa hijrahnya kepada Allah dan Rasul-Nya, maka hijrahnya kepada Allah dan Rasul-Nya. Dan barang siapa yang hijrahnya kepada harta dunia yang dicarinya atau seorang wanita yang dinikahinya, maka hijrahnya kepada apa yang dihijrahinya." (H.R. Bukhari Muslim).

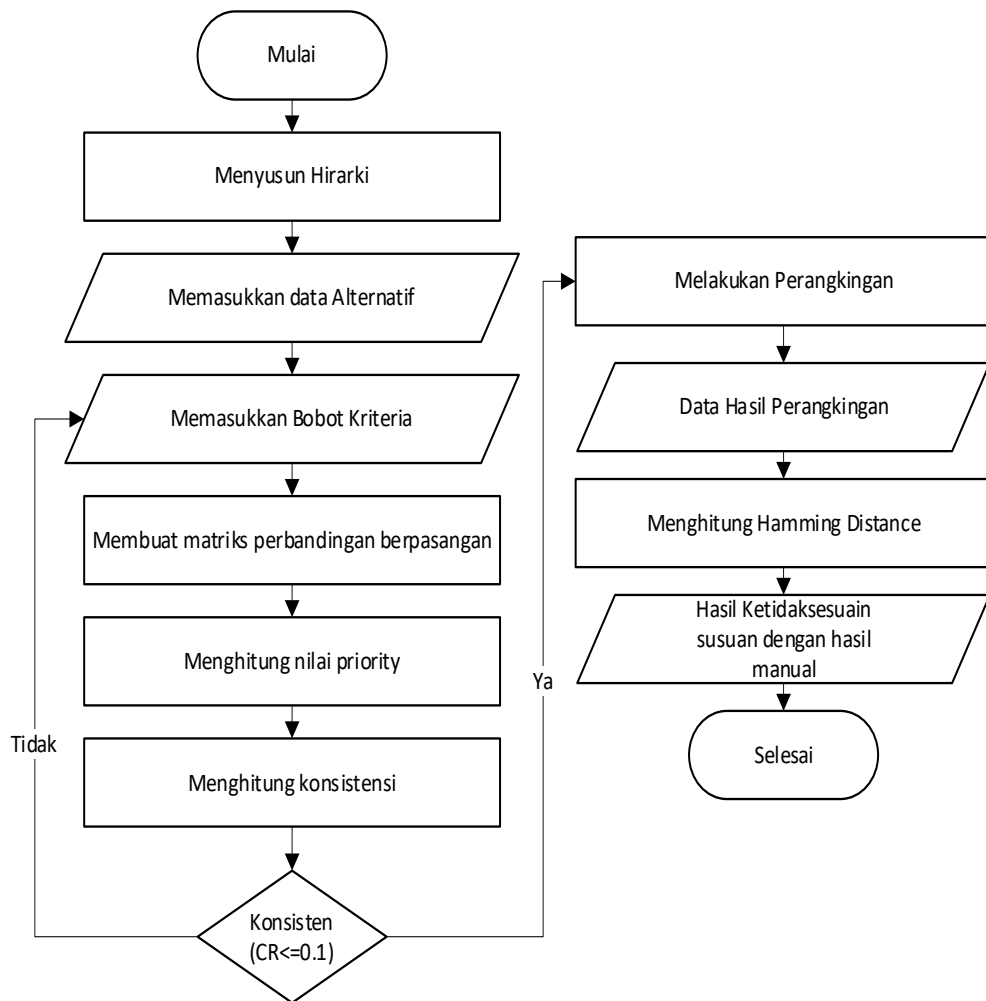
Pada hadist diatas terdapat pesan tentang kesungguhan, dimana untuk setiap aktivitas atau kegiatan harus dijalankan dengan niat yang lahir dari dalam hati. Aktivitas yang dikerjakan dengan penuh kesungguhan dilahirkan dari niat yang sungguh-sungguh dan benar. Aktivitas yang dilakukan akan menghasilkan sesuatu yang baik karena telah niatkan dalam hatinya. Artinya awal seseorang melakukan aktivitas sampai dengan tuntas semua dan berhasil harus dilakukan dengan kesungguhan dan niat yang benar.

BAB III

METODE ANALYTICAL HIERARCHY PROSES (AHP)

3.1 Desain Metode Analytical Hierarchy Proses (AHP)

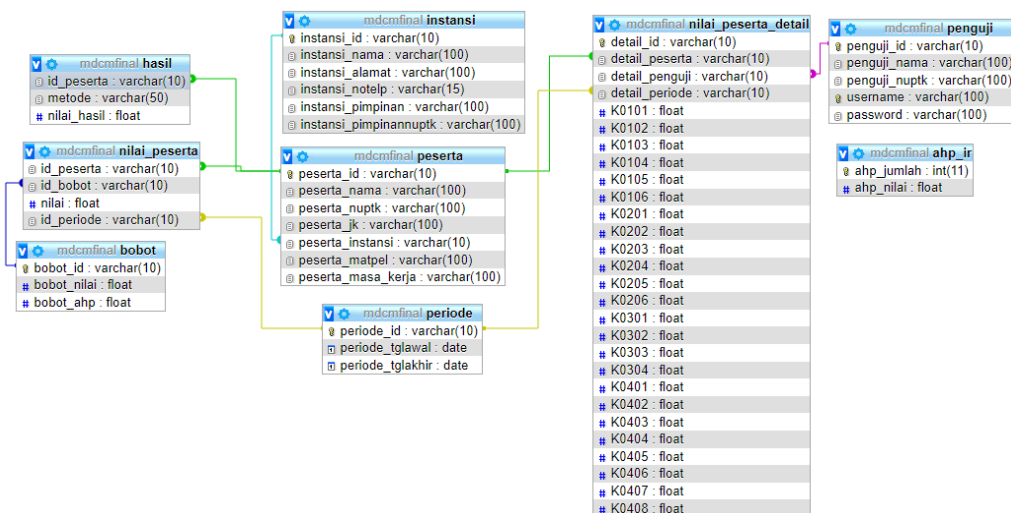
Desain metode *analytical hierarchy proses* (AHP) terdapat di Gambar 3.1. Desain itu AHP tersebut terdiri dari alur proses perhitungan AHP, perbandingan hasil AHP dan hasil perbandingan ketidaksesuaian hasil *hamming distance*.



Gambar 3.1 Desain Metode Analytical Hierarchy Proses (AHP)

3.2 Implementasi Metode Analytical Hierarchy Proses (AHP)

Metode AHP akan diimplementasikan dengan memakaia bahasa pemrograman PHP dan dan *database management sistem* yang digunakan adalah MySQL. Desain database adalah seperti Gambar 3.2.



Gambar 3.2 Desain Database

Berikut ini adalah tahapan dalam pengimplementasian metode AHP:

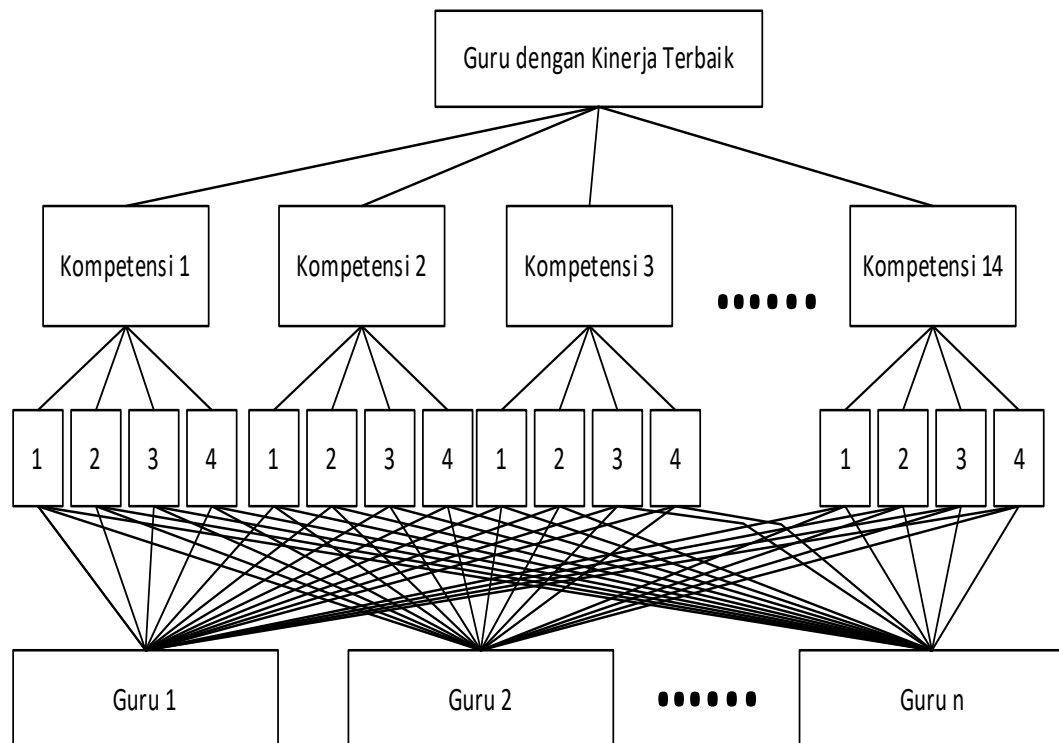
1. Menyusun hirarki

Hirarki AHP yang akan diterapkan untuk SPK penilaian kinerja guru ini memiliki 14 kriteria dan tidak memiliki sub kriteria. Tidak ditetapkan sub kriteria karena untuk menghasilkan pembobotan kriteria yang sama dengan metode multi criteria decision making lainnya. 14 kriteria merupakan 14 kompetensi untuk menghitung kinerja guru. 14 kriteria tersebut adalah

- Mengenal karakteristik peserta didik (C01),
- Menguasai teori belajar dan prinsip-prinsip pembelajaran yang mendidik (C02),

- Pengembangan kurikulum (C03),
- Kegiatan pembelajaran yang mendidik (C04),
- Mengembangkan potensi peserta didik (C05),
- Komunikasi dengan peserta didik (C06),
- Penilaian dan evaluasi (C07),
- Bertindak sesuai dengan norma agama, hukum, sosial dan kebudayaan nasional (C08),
- Menunjukkan pribadi yang dewasa dan teladan (C09),
- Etos kerja, tanggung jawab yang tinggi, rasa bangga menjadi guru (C10),
- Bersikap inklusif, bertindak obyektif, serta tidak diskriminatif (C11),
- Komunikasi dengan sesama guru, tenaga kependidikan, orang tua, peserta didik, dan masyarakat (C12),
- Penguasaan materi, struktur, konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu (C13)
- dan mengembangkan keprofesionalan melalui tindakan yang reflektif (C14).

Masing-masing kriteria diatas memiliki nilai antara 1-4 sesuai dengan hasil perhitungan persentase penilaian poin dari setiap kriteria. Sehingga susunan hirarki AHP yang diterapkan adalah seperti Gambar 3.3



Gambar 3.3 Hirarki AHP SPK kinerja Guru

2. Memasukkan data alternatif

Data alternatif terdiri dari data peserta dan nilai peserta. Nilai peserta terdiri dari 14 kompetensi. Nilai 14 kompetensi tersebut adalah hasil perhitungan dari poin-poin penilaian dimasing-masing kompetensi. Pada poin penilaian tersebut dinilai dengan 3 kategori yaitu

- Nilai 0 untuk tidak terpenuhi
- Nilai 1 untuk terpenuhi Sebagian
- Nilai 2 untuk seluruhnya terpenuhi

Untuk memasukkan data alternatif digunakan form seperti Gambar 3.4

Gambar 3.4 Form Input Nilai Peserta

Dari form tersebut kemudian poin-poin penilaian dihitung menggunakan rumus:

- Skor maksimum = jumlah indikator x 2
- Persentase = $(\text{total skor} / \text{skor maksimum}) \times 100 \%$

Kode program untuk melakukan perhitungan diatas adalah

```

$temp = 0;
for($f=$simpantiti2[$ttkawal]; $f < $simpantiti2[$ttkakhir]; $f++){
    $temp=$temp+$simpannilaipoinkomp[$f];
}
$simpanpersenkomp[$i] = round(($temp/((($simpantiti2[$ttkakhir]-$simpantiti2[$ttkawal])*2))*100), 0);
$simpanakhirkomp[$i] = viewpersenkomp($simpanpersenkomp[$i]);
$total = $total + $simpanakhirkomp[$i];

```

Gambar 3.5 Kode Program Persentase Nilai Peserta

Setelah persentase didapatkan maka dilakukan pengkategorian dengan kondisi:

```

function viewpersenkomp($persen) {
    $hasil = 0;
    if($persen >= 0 && $persen <= 25){
        $hasil = 1;
    }else if($persen > 25 && $persen <= 50){
        $hasil = 2;
    }else if($persen > 50 && $persen <= 75){
        $hasil = 3;
    }else if($persen > 75 && $persen <= 100){
        $hasil = 4;
    }
    return $hasil;
}

```

Gambar 3.6 Kode Program Pengkategorian Nilai Peserta

Sehingga didapatkan nilai untuk 14 kompetensi tersebut dengan rentang nilai antara 1 – 4. Nilai inilah yang akan menjadi nilai dari 14 kompetensi yang dimiliki setiap alternatif. Data alternatif disimpan pada tabel “nilai_peserta”. Berikut adalah kode program untuk melakukan penyimpanan nilai tersebut

```

addnilai($detail_peserta, 'C01', $simpanakhirkomp[0], $detail_periode);
addnilai($detail_peserta, 'C02', $simpanakhirkomp[1], $detail_periode);
addnilai($detail_peserta, 'C03', $simpanakhirkomp[2], $detail_periode);
addnilai($detail_peserta, 'C04', $simpanakhirkomp[3], $detail_periode);
addnilai($detail_peserta, 'C05', $simpanakhirkomp[4], $detail_periode);
addnilai($detail_peserta, 'C06', $simpanakhirkomp[5], $detail_periode);
addnilai($detail_peserta, 'C07', $simpanakhirkomp[6], $detail_periode);
addnilai($detail_peserta, 'C08', $simpanakhirkomp[7], $detail_periode);
addnilai($detail_peserta, 'C09', $simpanakhirkomp[8], $detail_periode);
addnilai($detail_peserta, 'C10', $simpanakhirkomp[9], $detail_periode);
addnilai($detail_peserta, 'C11', $simpanakhirkomp[10], $detail_periode);
addnilai($detail_peserta, 'C12', $simpanakhirkomp[11], $detail_periode);
addnilai($detail_peserta, 'C13', $simpanakhirkomp[12], $detail_periode);
addnilai($detail_peserta, 'C14', $simpanakhirkomp[13], $detail_periode);

```

Gambar 3.6 Kode Program Penyimpanan Nilai Peserta

Kode program function “addnilai” yang berisi query untuk menyimpan kedatabase

```
function addnilai($aidpeserta, $aidbobot, $anilai, $aperiode) {
    include('config.php');
    $query = "INSERT INTO nilai_peserta (id_peserta, id_bobot, nilai, id_periode) "
        . "VALUES('$aidpeserta', '$aidbobot', '$anilai', '$aperiode)";
    mysqli_query($db, $query);
    return mysqli_affected_rows($db);
}
```

Gambar 3.7 Kode Program Function Query Nilai Peserta

3. Memasukkan bobot kriteria

Dalam menentukan bobot pada setiap kriteria tidak ada aturan baku untuk menentukan kriteria mana yang menempati bobot tertinggi atau bobot terendah. Pada penelitian ini penulis mengasumsikan memberikan pembobotan pada kriteria berdasarkan acuan atau asumsi pribadi sebagai seorang DM (Decision Maker) / (PK) Pengambil Keputusan dengan melihat indikator pada masing-masing kriteria yang mana makin tinggi nilai kompetensi yang diberikan oleh DM akan makin tinggi bobot kepentingannya, sehingga didapatkan hasil sebagai berikut;

Nilai/Bobot	Tingkat Kepentingan	Kriteria/Kompetensi
1	(Sama Penting)	
3	(Sedikit Lebih Penting)	C08, C09, C11, C13, C14
5	(Jelas Lebih Penting)	C03, C04, C10
7	(Sangat Lebih Penting)	C02, C07, C12
9	(Sangat Jelas Lebih Penting)	C01, C05, C06

Dikarenakannya tidak adanya aturan baku pada penentuan bobot kriteria maka penulis memberikan form isian untuk menentukan bobot masing-masing kriteria berdasarkan pada asumsi dari DM (Decision Maker) / PK (Pengambil Keputusan).

Berikut adalah kode program untuk mengatur bobot dari kriteria.

```

$book = ambileditbobot($_GET['kode']);
$edit = mysqli_fetch_assoc($book);
$dafkom = viewdafkomp();
?>
<div class="row">
  <div class="col-auto mr-auto"><h4>Silahkan isi untuk mengubah kriteria dengan ID :<?php echo $edit['bobot_id'];?></h4></div>
  <div class="col-auto"><a href="bobot_view.php" class="btn btn-info">Lihat Data</a></div>
</div>
<form method="post">
  <div class="form-group">
    <label>ID Kriteria</label>
    <input type="text" class="form-control" name="id" value="<?php echo $edit['bobot_id'];?>" disabled/>
    <input type="hidden" name="id" value="<?php echo $edit['bobot_id'];?>" />
  </div>
  <div class="form-group">
    <label>Kriteria</label>
    <input type="text" class="form-control" name="nama" value="<?php echo $dafkom[$_GET['komp']];?>" disabled/>
  </div>
  <div class="form-group">
    <label>Bobot</label>
    <select class="form-control" name="nilai">
      <option value="1" <?php if($edit['bobot_nilai'] == '1'){ echo 'selected'; }?>>Sangat Rendah</option>
      <option value="2" <?php if($edit['bobot_nilai'] == '2'){ echo 'selected'; }?>>Rendah</option>
      <option value="3" <?php if($edit['bobot_nilai'] == '3'){ echo 'selected'; }?>>Cukup</option>
      <option value="4" <?php if($edit['bobot_nilai'] == '4'){ echo 'selected'; }?>>Tinggi</option>
      <option value="5" <?php if($edit['bobot_nilai'] == '5'){ echo 'selected'; }?>>Sangat Tinggi</option>
    </select>
  </div>
  <button type="submit" class="btn btn-primary" name="kirim">
    Ubah
  </button>
</form>

```

Gambar 3.7 Kode Program Form Bobot Kriteria

Berikut ini adalah tampilan dari form untuk mengatur bobot kriteria

The screenshot shows a web application interface for setting criteria weights. The page title is "SD ANAK SALEH" and the user is "admin". The main content area shows a form for editing criteria with ID "C01". The criteria is "Kompetensi 1 : Mengenal Karakteristik Peserta Didik". The weight is set to "Sangat Tinggi" in a dropdown menu. A "Lihat Data" button is visible in the top right corner.

Gambar 3.8 Tampilan Form Bobot Kriteria

Form ini akan mengirim data melalui kode program dibawah ini :

```

if (isset($_POST['kirim'])) {
    $id = $_POST['id'];
    $nilai = $_POST['nilai'];
    $nilai2 = 0;
    if($nilai == 1){
        $nilai2 = 1;
    }else if($nilai == 2){
        $nilai2 = 3;
    }else if($nilai == 3){
        $nilai2 = 5;
    }else if($nilai == 4){
        $nilai2 = 7;
    }else if($nilai == 5){
        $nilai2 = 9;
    }
    $success = updatebobot($id, $nilai, $nilai2);
    if ($success > 0) {
        header('Location: bobot_view.php');
    } else {
        echo ' <div class="alert alert-danger" role="alert">
            <strong>Maaf</strong>, data gagal diinputkan, silahkan coba inputkan kembali.
            </div>';
    }
}

```

Gambar 3.9 Kode Program Bobot Kriteria

Pada kode program diatas bobot terlebih dahulu akan di konversi sesuai pembobotan pada AHP yaitu

Nilai 1	Dikonversi menjadi 1 (Sama Penting)
Nilai 2	Dikonversi menjadi 3 (Sedikit Lebih Penting)
Nilai 3	Dikonversi menjadi 5 (Jelas Lebih Penting)
Nilai 4	Dikonversi menjadi 7 (Sangat Lebih Penting)
Nilai 5	Dikonversi menjadi 9 (Sangat Jelas Lebih Penting)

Setelah nilai dikoversi, nilai pembobotan dari kriteria tersebut akan disimpan ke database dengan memanggil function “updatebobot” melalui kode program dibawah ini

```
“$success = updatebobot($id, $nilai, $nilai2); ”
```

Gambar 3.10 Kode Program Memanggil Function Ganti Bobot Kriteria

Dibawah ini adalah isi function “updatebobot” yang berfungsi untuk menyimpan data ke database yaitu table “bobot”

```
function updatebobot($aid, $anilai, $anilai2) {
    include('config.php');
    $query = "UPDATE bobot "
        . "SET bobot_nilai='$anilai', bobot_ahp='$anilai2' "
        . "WHERE bobot_id='$aid'";
    mysqli_query($db, $query);
    return mysqli_affected_rows($db);
}
```

Gambar 3.11 Kode Program Function Query Ganti Bobot Kriteria

4. Membuat matriks perbandingan berpasangan

Langkah pada tahap ini dari mengambil data dari database. Untuk mengambil data yang dibutuhkan dari database menggunakan function dibawah ini:

```
$tampil = viewpenilaianahp();
```

Gambar 3.12 Kode Program Memanggil Penilaian AHP

Function tersebut berisi:

```
function viewpenilaianahp(){
    include('config.php');
    $query = "SELECT b.peserta_id, b.peserta_nama, c.bobot_id, a.nilai, c.bobot_ahp
        FROM nilai_peserta a
        LEFT JOIN peserta b ON a.id_peserta = b.peserta_id
        LEFT JOIN bobot c ON a.id_bobot = c.bobot_id";
    $result = mysqli_query($db, $query);
    return $result;
}
```

Gambar 3.13 Kode Program Function Query View Penilaian AHP

Data hasil query tersebut kemudian diolah untuk memisahkan data kriteria dan data bobot dengan kode program dibawah ini:

```

$data      =array();
$kriterias =array();
$bobot     =array();
$setnama  =array();
if ($stampil) {
} while ($row=$tampil->fetch_object()){
}   if(!isset($data[$row->peserta_id])){
}   $data[$row->peserta_id]=array();
}   }
}   if(!isset($data[$row->peserta_id][$row->bobot_id])){
}   $data[$row->peserta_id][$row->bobot_id]=array();
}   }
}   $bobot[$row->bobot_id]=$row->bobot_ahp;
}   $data[$row->peserta_id][$row->bobot_id]=$row->nilai;
}   $kriterias[]=$row->bobot_id;
}   $setnama[$row->peserta_id]=$row->peserta_nama;
}   }
}   }
}   $kriteria      =array_unique($kriterias);
}   $jml_kriteria  =count($kriteria);
}

```

Gambar 3.14 Kode Program Pengolahan Penilaian AHP

Dari kode diatas didapat data dibawah ini:

\$data dipakai sebagai penyimpan data-data alternatif

\$bobot dipakai sebagai penyimpan data bobot kriteria

\$kriteria dipakai sebagai penyimpan data nama kriteria (berupa id)

\$setnama dipakai sebagai penyimpan nama alternatif

Berikut ini adalah kode program untuk menghitung perbandingan berpasangan

```

$hitung1 = array();
foreach ($bobot as $k1 => $a1){
    echo '<tr><td><b>'. $k1. '</b></td>';
    foreach ($bobot as $k2 => $a2){
        $selisinilai = sqrt(pow($a1-$a2, 2));
        if($a1<$a2){
            $kata = "1/[". $k2. "][". $k1. "] =";
            $ptemp = 1/($selisinilai+1);
        }
        if($a1>$a2){
            $kata = "bobot asli =";
            $ptemp = $selisinilai+1;
        }
        if($k1==$k2){
            $kata = "";
            $ptemp = 1;
        }
        $hitung1[$k1][$k2] = $ptemp;
        echo '<td><div class="text-xs">'. $kata. '</div>'. round($ptemp, 4). '</td>';
    }
    echo '</tr><tr>';
}

```

Gambar 3.15 Kode Program Menghitung Perbandingan Berpasangan

Menjumlahkan hasilnya

```

$totalhitung1 = array();
$temp = 0;
$temp1 = "";
echo '<td><b>Jumlah</b></td>';
foreach ($bobot as $k1 => $a1){
    $temp = 0;
    foreach ($bobot as $k2 => $a2){
        $temp1 = $k1;
        $temp = $temp + $hitung1[$k2][$k1];
    }
    echo '<td><b>' . round($temp, 4) . '</b></td>';
    $totalhitung1[$temp1]=$temp;
}

```

Gambar 3.16 Kode Program Jumlah Perbandingan Berpasangan

5. Menghitung nilai priority

Membuat matriks nilai kriteria yang merupakan hasil pembagian nilai berpasangan dengan jumlah hasil.

```

$temp = 0;
foreach ($bobot as $k2 => $a2){
    if($totalhitung1[$k2] == 0){
        $temptemp = 0;
    }else{
        $temptemp = $hitung1[$k1][$k2]/$totalhitung1[$k2];
    }
    $temp = $temp + $temptemp;
    echo '<td><div class="text-xs">' . $k1 . ' / jumlah[' . $k2 . '] =</div>' . round($temptemp, 4) . '</td>';
}

```

Gambar 3.17 Kode Program Pembagian Nilai Berpasangan

Menghitung nilai prioritas (Priority Vector) yang merupakan nilai rata-rata.

```

$temptemp = $temp/$jml_kriteria;
$ratahitung1[$k1]=$temptemp;

```

Gambar 3.18 Kode Program Menghitung Nilai Priority

6. Menghitung konsistensi

Menghitung jumlah hasil perkalian nilai perpasangan dengan nilai prioritas

```

$sumhitungl = array();
foreach ($bobot as $k1 => $a1){
    echo '<tr><td><b>'. $k1. '</b></td>';
    $temp = 0;
    foreach ($bobot as $k2 => $a2){
        $temptemp = $hitungl[$k1][$k2]*$ratahitungl[$k2];
        $temp = $temp + ($temptemp);
        echo '<td><div class="text-xs">'. $k2. ' * rata-rata ['. $k1. ']=</div>'.round($temptemp, 4). '</td>';
    }
    $sumhitungl[$k1]=$temp;
    echo '<td><b>'.round($temp, 4). '</b></td></tr>';
}

```

Gambar 3.19 Kode Program Perkalian Nilai Priority

Menghitung Consistency Index (CI)

```

$temp = 0;
foreach ($bobot as $k1 => $a1){
    $temp = $temp + ($sumhitungl[$k1]/$ratahitungl[$k1]);
}
$t = 1/$jml_kriteria*$temp;
$ci = ($t-$jml_kriteria)/($jml_kriteria-1);

```

Gambar 3.20 Kode Program Consistency Index (CI)

Mencari Consistency Ratio (CR) untuk mengetahui tingkat konsistensi

```

$tampil2 = ambilir($jml_kriteria);
$htampil2 = mysqli_fetch_assoc($tampil2);
echo '<tr><td>IR</td><td>'. $htampil2['ahp_nilai']. '</td></tr>';
$cr = ($ci/$htampil2['ahp_nilai']);
echo '<tr><td>Consistency Ratio (CR)</td><td>'. $cr;
if($cr<=0.1){
    echo " [konsisten]</td></tr>";
}else{
    echo " [tidak konsisten]</td></tr>";
}

```

Gambar 3.21 Kode Program Consistency Ratio (CR)

7. Melakukan perangkingan

```

$i = 0;
arsort($hasilfinal);
foreach($hasilfinal as $x=>$x_value) {
    echo "<tr>
        <td>" . (++$i) . "</td>
        <td>{$x}</td>
        <td>${setnama[$x]}</td>
        <td>".round($x_value, 4). "</td>";
}

```

Gambar 3.22 Kode Program Perangkingan

8. Menghitung *hamming distance*

Untuk menghitung *hamming distance* yang digunakan untuk mengetahui tingkat ketidaksesuan perangkingan dengan hasil perhitungan manual, maka langkah yang harus dilakukan adalah:

1. Merangking hasil perhitungan manual dan mencari guru yang memiliki nilai yang sama untuk ditetapkan sebagai rangking bersama.

Perangkingan

No	ID Peserta	Nama	Nilai
1	PS0021 (1)	Moch. Yusroni, S.Pd.	37
2	PS0014 (1)	Diah Anita Ludfiana, S.Pd.	37
3	PS0017 (1)	Yeni Yusella, S.Pd.	37
4	PS0022 (4)	Nihayatul Hasanah, S.Pd	36

Gambar 3.23 Contoh Perangkingan Bersama

Gambar diatas adalah contoh hasil perangkingan manual. Dalam hasil perangkingan atas terdapat rangking bersama yaitu no. 1, 2 dan 3 atau Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. merupakan rangking bersama karena memiliki nilai yang sama yaitu 37. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd jika dirangkingkan, maka mereka ada dirangking 1, 2 dan 3. Sedangkan Nihayatul Hasanah, S.Pd ada pada rangking 4.

Kode program untuk mengambil hasil perangkingan perhitungan manual seperti yang terdapat pada gambar 3.24. Pada kode program tersebut hasil perhitungan manual akan disimpan pada variabel `stmt4`.

```
$stmt4 = viewnilaidetailperangkingan();
```

Gambar 3.24 Kode Program Untuk Mengambil Hasil Perhitungan Manual Setelah dilakukan pengambilan perangkingan manual, pada data perangkingan manual tersebut akan dicari guru yang memiliki nilai yang sama untuk untuk ditetapkan sebagai rangking bersama. Gambar 3.25 merupakan kode program menjalankan perbandingan.

```
$i = 0;
$ranking = 1 ;
$perban = -1;
while ($data2 = mysqli_fetch_assoc($stmt4)) {
    if($perban == -1){
        $perban = $data2['total'];
    }
    if($perban != $data2['total']){
        $ranking = $i+1;
        $perban = $data2['total'];
    }
    $simpan_hsl_manual[$i] = $data2['detail_peserta'];
    $simpan_n_manual[$data2['detail_peserta']] = $ranking;
    $i++;
};
```

Gambar 3.25 Kode Program Untuk Mencari Rangking Bersama

2. Mengambil hasil perangkingan dengan metode AHP

Hasil perangkingan ini yang akan dibandingkan dengan hasil prangkingan manual. Kode program untuk Mengambil hasil perangkingan dengan metode AHP adalah :

```
$stmt = viewsimpanhasil("AHP");
$i = 0;
while ($data2 = mysqli_fetch_assoc($stmt)) {
    $simpan_hsl_ahp[$i] = $data2['id_peserta'];
    $i++;
};
```

Gambar 3.26 Kode Program Mengambil hasil perangkingan dengan metode AHP

3. Mencari nilai *hamming distance*

Kode Program untuk mencari nilai *hamming distance* seperti yang ditunjukkan pada Gambar 3.27 :

```
function perbandingan($nilaimetode, $nilaimanual, $arraynilaimanual) {
    $tdksama = 0;
    $temptdksama = true;
    if($nilaimetode==$nilaimanual){
        $temptdksama = false;
        echo "<td><b>[".$nilaimetode."</b></td>";
    }else{
        echo "<td>";
        foreach($arraynilaimanual as $x=>$x_value) {
            //echo $x." ".$x_value."</br>";
            if($arraynilaimanual[$nilaimanual] == $x_value){
                if($arraynilaimanual[$nilaimetode] == $x_value){
                    echo "<div class='text-xs'>";
                    if($nilaimetode==$x){
                        $temptdksama = false;
                        echo "<b>[".$nilaimetode." ".$x." ".$x_value."</b> | ";
                    }else{
                        echo $nilaimetode." ".$x." ".$x_value." | ";
                    }
                    echo "</div>";
                }
            }
        }

        if($temptdksama == true){
            $tdksama++;
            echo $nilaimetode;
        }else{
            echo "<b>[".$nilaimetode."</b>";
        }
        echo "</td>";
    }
    return $tdksama;
}
```

Gambar 3.27 Kode Program *hamming distance*

Untuk mencari nilai *hamming distance* dengan cara yaitu :

- 1) Mengambil data guru yang berada pada rangking yang dicari pada hasil perangkingan manual.
- 2) Membandingkan dengan hasil pengakingan manual dengan hasil perangkingan metode AHP pada rangking yang dicari. Dengan ketentuan:
 - Jika sama, lanjut ke langkah 3 dan tidak dihitung sebagai ketidaksesuaian

4	9	7	7	7	7	5	5	5	5	3	3	3	3	3
5	3	3	3	3	3	5	5	5	5	7	7	7	7	9
6	5	5	3	7	7	5	9	3	7	5	5	3	3	3

Keterangan tabel adalah

C01 = Mengenal karakteristik peserta didik,

C02 = Menguasai teori belajar dan prinsip-prinsip pembelajaran yang mendidik,

C03 = Pengembangan kurikulum,

C04 = Kegiatan pembelajaran yang mendidik,

C05 = Mengembangkan potensi peserta didik,

C06 = Komunikasi dengan peserta didik,

C07 = Penilaian dan evaluasi,

C08= Bertindak sesuai dengan norma agama, hukum, sosial dan kebudayaan nasional,

C09 = Menunjukkan pribadi yang dewasa dan teladan,

C10 = Etos kerja, tanggung jawab yang tinggi, rasa bangga menjadi guru,

C11 = Bersikap inklusif, bertindak obyektif, serta tidak diskriminatif,

C12 = Komunikasi dengan sesama guru, tenaga kependidikan, orang tua, peserta didik, dan masyarakat,

C13 = Penguasaan materi, struktur, konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu, dan

C14 = dan mengembangkan keprofesionalan melalui tindakan yang reflektif.

Hasil perangkingan dari pengujian kemudian dihitung kevalidasinya berdasarkan nilai ketidaksesuaian perangkingan dengan hasil perhitungan manual dengan algoritma *hamming distance*. Nilai algoritma *hamming*

distance yang semakin besar menandakan semakin besar berbeda dengan hasil perhitungan manual atau semakin banyak yang perankingannya tidak sesuai.

Proses pengujian adalah :

1. Pengujian 1

Pada pengujian 1 ini menggunakan bobot kriteria terendah untuk semua kriteria. Bobot-bobot kriteria tersebut seperti pada Tabel 3.2 dibawah ini:

Tabel 3.2 Bobot Kriteria Pengujian 1 AHP

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 3.3.

Tabel 3.3 Hasil Pengujian 1 AHP

Uji Coba	Hasil Nilai Hamming Distance untuk Metode AHP
1	21
2	16
3	10
4	12
5	14
Rata-rata	14.6

Hasil pengujian 1 untuk metode AHP ini menghasilkan nilai ketidaksesuaian terendah adalah 10 kesalahan, nilai ketidaksesuaian tertinggi adalah 21

kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 14.6 kesalahan.

2. Pengujian 2

Pada pengujian 2 ini menggunakan bobot kriteria tertinggi untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 3.4.

Tabel 3.4 Bobot Kriteria Pengujian 2 AHP

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	9	9	9	9	9	9	9	9	9	9	9	9	9	9

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 3.5.

Tabel 3.5 Hasil Pengujian 2 AHP

Uji Coba	Hasil Nilai Hamming Distance untuk Metode AHP
1	10
2	12
3	8
4	9
5	11
Rata-rata	10

Hasil pengujian 2 untuk metode AHP ini menghasilkan nilai ketidaksesuaian terendah adalah 8 kesalahan, nilai ketidaksesuaian tertinggi adalah 12 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 10 kesalahan.

3. Pengujian 3

Pada pengujian 3 ini menggunakan bobot kriteria nilai 5 (pertengahan) untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 3.6.

Tabel 3.6 Bobot Kriteria Pengujian 3 AHP

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 3.7.

Tabel 3.7 Hasil Pengujian 3 AHP

Uji Coba	Hasil Nilai Hamming Distance untuk Metode AHP
1	13
2	18
3	12
4	16
5	15
Rata-rata	14,8

Hasil pengujian 3 untuk metode AHP ini menghasilkan nilai ketidaksesuaian terendah adalah 12 kesalahan, nilai ketidaksesuaian tertinggi adalah 18 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 14,8 kesalahan.

4. Pengujian 4

Pada pengujian 4 ini menggunakan bobot kriteria dari tinggi ke rendah.

Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 3.8.

Tabel 3.8 Bobot Kriteria Pengujian 4 AHP

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	9	7	7	7	7	5	5	5	5	3	3	3	3	3

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 3.9.

Tabel 3.9 Hasil Pengujian 4 AHP

Uji Coba	Hasil Nilai Hamming Distance untuk Metode AHP
1	19
2	15
3	12
4	16
5	20
Rata-rata	16,4

Hasil pengujian 4 untuk metode AHP ini menghasilkan nilai ketidaksesuaian terendah adalah 12 kesalahan, nilai ketidaksesuaian tertinggi adalah 20 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 16,4 kesalahan.

5. Pengujian 5

Pada pengujian 5 ini menggunakan bobot kriteria dari nilai rendah ke tinggi. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 3.10.

Tabel 3.10 Bobot Kriteria Pengujian 5 AHP

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	3	3	3	3	3	5	5	5	5	7	7	7	7	9

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 3.11.

Tabel 3.11 Hasil Pengujian 5 AHP

Uji Coba	Hasil Nilai Hamming Distance untuk Metode AHP
1	17
2	17
3	17
4	21
5	19
Rata-rata	18,2

Hasil pengujian 5 untuk metode AHP ini menghasilkan nilai ketidaksesuaian terendah adalah 17 kesalahan, nilai ketidaksesuaian tertinggi adalah 21 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 18,2 kesalahan.

6. Pengujian 6

Pada pengujian 6 ini menggunakan bobot kriteria yang acak. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 3.12.

Tabel 3.12 Bobot Kriteria Pengujian 6 AHP

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	5	5	3	7	7	5	9	3	7	5	5	3	3	3

Pengujian dilakukan dengan jumlah alternatif/peserta/guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 3.13.

Tabel 3.13 Hasil Pengujian 6 AHP

Uji Coba	Hasil Nilai Hamming Distance untuk Metode AHP
1	15
2	17
3	18
4	19
5	15
Rata-rata	16,8

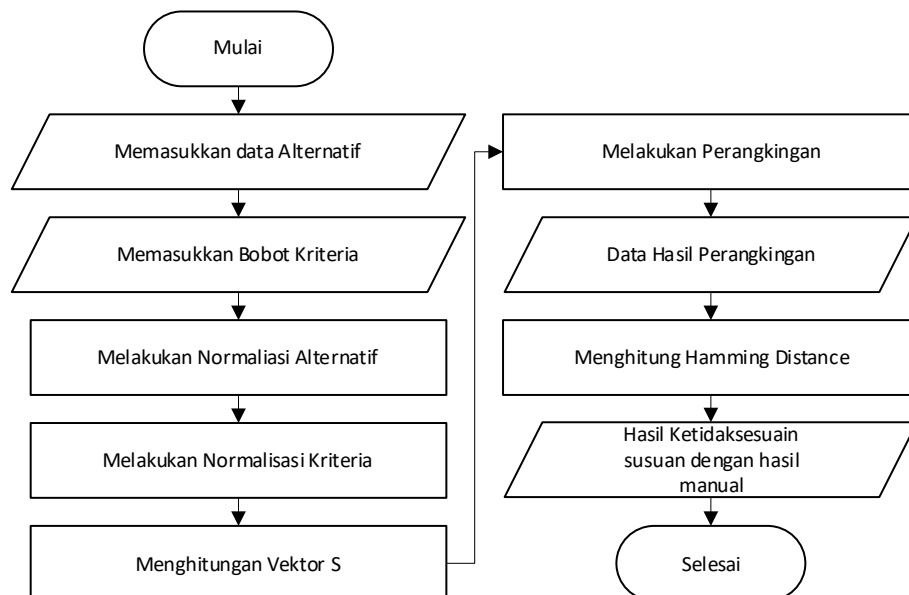
Hasil pengujian 6 untuk metode AHP ini menghasilkan nilai ketidaksesuaian terendah adalah 15 kesalahan, nilai ketidaksesuaian tertinggi adalah 19 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 16,8 kesalahan

BAB IV

METODE WEIGHTED SUM MODEL (WSM)

4.1 Desain Metode Weighted Sum Model (WSM)

Desain metode *weighted sum model* (WSM) terapat di Gambar 4.1. Desain itu metode WSM tersebut terdiri dari alur proses perhitungan WSM, perangkingan hasil WSM dan hasil perbandingan ketidaksesuaian hasil *hamming distance*.



Gambar 4.1 Desain Metode Weighted Sum Model (WSM)

4.2 Implementasi Metode Weighted Sum Model (WSM)

Metode WSM akan diimplentasikan dengan memakai bahasa pemrograman PHP dan dan *database management sistem* yang digunakan adalah MySQL. Desain database seperti pada Gambar 3.2.

Berikut ini adalah tahapan dalam pengimplementasian metode WSM:

1. Memasukkan data alternatif

Data alternatif terdiri dari data peserta dan nilai peserta. Nilai peserta terdiri dari 14 kompetensi. Nilai 14 kompetensi tersebut adalah hasil perhitungan dari poin-poin penilaian dimasing-masing kompetensi. Pada poin penilaian tersebut dinilai dengan 3 kategori yaitu

- Nilai 0 untuk tidak terpenuhi
- Nilai 1 untuk terpenuhi Sebagian
- Nilai 2 untuk seleuruhnya terpenuhi

Dibawah ini adalah tampilan form untuk memasukkan data alternatif

Gambar 4.2 Form Input Nilai Peserta

Dari form tersebut kemudian poin-poin penilaian dihitung menggunakan rumus:

- Skor maksimum = jumlah indikator x 2
- Persentase = $(\text{total skor} / \text{skor maksimum}) \times 100 \%$

Kode program untuk melakukan perhitungan diatas adalah

```

$temp = 0;
for($f=$simpantiti2[$ttkawal]; $f < $simpantiti2[$ttkakhir]; $f++){
    $temp=$temp+$simpannilaipoinkomp[$f];
}
$simpanpersenkomp[$i] = round(($temp/($simpantiti2[$ttkakhir]-$simpantiti2[$ttkawal])*2)*100, 0);
$simpanakhirkomp[$i] = viewpersenkomp($simpanpersenkomp[$i]);
$total = $total + $simpanakhirkomp[$i];

```

Gambar 4.3 Kode Program Persentase Nilai Peserta

Setelah persentase didapatkan maka dilakukan pengkategorian dengan kondisi:

```

function viewpersenkomp($persen) {
    $hasil = 0;
    if($persen >= 0 && $persen <= 25){
        $hasil = 1;
    }else if($persen > 25 && $persen <= 50){
        $hasil = 2;
    }else if($persen > 50 && $persen <= 75){
        $hasil = 3;
    }else if($persen > 75 && $persen <= 100){
        $hasil = 4;
    }
    return $hasil;
}

```

Gambar 4.4 Kode Program Pengkategorian Nilai Peserta

Sehingga didapatkan nilai untuk 14 kompetensi tersebut dengan rentang nilai antara 1 – 4. Nilai inilah yang akan menjadi nilai dari 14 kompetensi yang dimiliki setiap alternatif. Data alternatif disimpan pada tabel “nilai_peserta”.

Berikut adalah kode program untuk melakukan penyimpanan nilai tersebut

```

addnilai($detail_peserta, 'C01', $simpanakhirkomp[0], $detail_periode);
addnilai($detail_peserta, 'C02', $simpanakhirkomp[1], $detail_periode);
addnilai($detail_peserta, 'C03', $simpanakhirkomp[2], $detail_periode);
addnilai($detail_peserta, 'C04', $simpanakhirkomp[3], $detail_periode);
addnilai($detail_peserta, 'C05', $simpanakhirkomp[4], $detail_periode);
addnilai($detail_peserta, 'C06', $simpanakhirkomp[5], $detail_periode);
addnilai($detail_peserta, 'C07', $simpanakhirkomp[6], $detail_periode);
addnilai($detail_peserta, 'C08', $simpanakhirkomp[7], $detail_periode);
addnilai($detail_peserta, 'C09', $simpanakhirkomp[8], $detail_periode);
addnilai($detail_peserta, 'C10', $simpanakhirkomp[9], $detail_periode);
addnilai($detail_peserta, 'C11', $simpanakhirkomp[10], $detail_periode);
addnilai($detail_peserta, 'C12', $simpanakhirkomp[11], $detail_periode);
addnilai($detail_peserta, 'C13', $simpanakhirkomp[12], $detail_periode);
addnilai($detail_peserta, 'C14', $simpanakhirkomp[13], $detail_periode);

```

Gambar 4.5 Kode Program Penyimpan Nilai Peserta

Kode program function “addnilai” yang berisi query untuk menyimpan ke database

```

function addnilai($aidpeserta, $aidbobot, $anilai, $aperiode) {
    include('config.php');
    $query = "INSERT INTO nilai_peserta (id_peserta, id_bobot, nilai, id_periode) "
        . "VALUES('$aidpeserta', '$aidbobot', '$anilai', '$aperiode)";
    mysqli_query($db, $query);
    return mysqli_affected_rows($db);
}

```

Gambar 4.6 Kode Program Function Query Nilai Peserta

2. Memasukkan bobot kriteria

Dalam menentukan bobot pada setiap kriteria tidak ada aturan baku untuk menentukan kriteria mana yang menempati bobot tertinggi atau bobot terendah. Pada penelitian ini penulis mengasumsikan memberikan pembobotan pada kriteria berdasarkan acuan atau asumsi pribadi sebagai seorang DM (Decision Maker) / (PK) Pengambil Keputusan dengan melihat indikator pada masing-masing kriteria bahwa semakin tinggi nilai kompetensi yang diberikan oleh DM maka semakin tinggi bobot kepentingannya, sehingga didapatkan hasil sebagai berikut;

Nilai/Bobot	Tingkat Kepentingan	Kriteria/Kompetensi
1	Sangat Rendah	-
2	Rendah	C08, C09, C11, C13, C14
3	Cukup	C03, C04, C10
4	Tinggi	C02, C07, C12
5	Sangat Tinggi	C01, C05, C06

Dikarenakannya tidak adanya aturan baku pada penentuan bobot kriteria maka penulis memberikan form isian untuk menentukan bobot masing-masing kriteria berdasarkan pada asumsi dari DM (Decision Maker) / PK (Pengambil Keputusan).

Berikut adalah kode program untuk mengatur bobot dari kriteria.

```

$book = ambileditbobot($_GET['kode']);
$edit = mysqli_fetch_assoc($book);
$dafkom = viewdafkom();
?>
<div class="row">
  <div class="col-auto mr-auto"><h4>Silahkan isi untuk mengubah kriteria dengan ID : <?php echo $edit['bobot_id']; ?></h4></div>
  <div class="col-auto"><a href="bobot_view.php" class="btn btn-info">Lihat Data</a></div>
</div>
<form method="post">
  <div class="form-group">
    <label>ID Kriteria</label>
    <input type="text" class="form-control" name="id" value="<?php echo $edit['bobot_id']; ?>" disabled/>
    <input type="hidden" name="id" value="<?php echo $edit['bobot_id']; ?>" />
  </div>
  <div class="form-group">
    <label>Kriteria</label>
    <input type="text" class="form-control" name="nama" value="<?php echo $dafkom[$_GET['komp']] ?>" disabled/>
  </div>
  <div class="form-group">
    <label>Bobot</label>
    <select class="form-control" name="nilai">
      <option value="1" <?php if($edit['bobot_nilai'] == '1'){ echo 'selected'; } ?>>Sangat Rendah</option>
      <option value="2" <?php if($edit['bobot_nilai'] == '2'){ echo 'selected'; } ?>>Rendah</option>
      <option value="3" <?php if($edit['bobot_nilai'] == '3'){ echo 'selected'; } ?>>Cukup</option>
      <option value="4" <?php if($edit['bobot_nilai'] == '4'){ echo 'selected'; } ?>>Tinggi</option>
      <option value="5" <?php if($edit['bobot_nilai'] == '5'){ echo 'selected'; } ?>>Sangat Tinggi</option>
    </select>
  </div>
  <button type="submit" class="btn btn-primary" name="kirim">
    Ubah
  </button>
</form>

```

Gambar 4.7 Kode Program Form Bobot Kriteria

Berikut ini adalah tampilan dari form untuk mengatur bobot kriteria

Gambar 4.8 Tampilan Form Bobot Kriteria

Form ini akan mengirim data melalui kode program dibawah ini:

```

if (isset($_POST['kirim'])) {
    $id = $_POST['id'];
    $nilai = $_POST['nilai'];
    $nilai2 = 0;
    if($nilai == 1){
        $nilai2 = 1;
    }else if($nilai == 2){
        $nilai2 = 3;
    }else if($nilai == 3){
        $nilai2 = 5;
    }else if($nilai == 4){
        $nilai2 = 7;
    }else if($nilai == 5){
        $nilai2 = 9;
    }
    $success = updatebobot($id, $nilai, $nilai2);
    if ($success > 0) {
        header('Location: bobot_view.php');
    } else {
        echo' <div class="alert alert-danger" role="alert">
            <strong>Maaf</strong>, data gagal diinputkan, silahkan coba inputkan kembali.
        </div>';
    }
}

```

Gambar 4.9 Kode Program Bobot Kriteria

Nilai pembobotan dari kriteria tersebut akan disimpan ke database dengan memanggil function “updatebobot” melalui kode program dibawah ini

“`$success = updatebobot($id, $nilai, $nilai2);`”

Gambar 4.10 Kode Program Memanggil Function Ganti Bobot Kriteria

Dibawah ini adalah isi function “updatebobot” yang berfungsi untuk menyimpan data ke database yaitu table “bobot”

```
function updatebobot($aid, $anilai, $anilai2) {
    include('config.php');
    $query = "UPDATE bobot "
            . "SET bobot_nilai='$anilai', bobot_ahp='$anilai2' "
            . "WHERE bobot_id='$aid'";
    mysqli_query($db,$query);
    return mysqli_affected_rows($db);
}
```

Gambar 4.11 Kode Program Function Query Ganti Bobot Kriteria

3. Melakukan normalisasi alternatif

Langkah ini diawali dengan mengambil data yang dibutuhkan dari database dengan kode program seperti pada Gambar 4.12

```
$stampil = viewpenilaian();
```

Gambar 4.12 Kode Program Memanggil Penilaian WSM

Kode diatas akan menjalankan kode program Gambar 4.13 yang berisi query database.

```
function viewpenilaian(){
    include('config.php');
    $query = "SELECT b.peserta_id, b.peserta_nama, c.bobot_id, a.nilai, c.bobot_nilai
            FROM nilai_peserta a
            LEFT JOIN peserta b ON a.id_peserta = b.peserta_id
            LEFT JOIN bobot c ON a.id_bobot = c.bobot_id";
    $result = mysqli_query($db,$query);
    return $result;
}
```

Gambar 4.14 Kode Program Memanggil Penilaian WSM

Data hasil query tersebut kemudian diolah untuk memisahkan data kriteria dan data bobot dengan kode program pada Gambar 4.15


```

$data      =array();
$kriterias =array();
$bobot     =array();
$setnama  =array();
if ($tampil) {
    while($row=$tampil->fetch_object()){
        if(!isset($data[$row->peserta_id])){
            $data[$row->peserta_id]=array();
        }
        if(!isset($data[$row->peserta_id][$row->bobot_id])){
            $data[$row->peserta_id][$row->bobot_id]=array();
        }
        $bobot[$row->bobot_id]=$row->bobot_nilai;
        $data[$row->peserta_id][$row->bobot_id]=$row->nilai;
        $kriterias[]=$row->bobot_id;
        $setnama[$row->peserta_id]=$row->peserta_nama;
    }
}
$kriteria  =array_unique($kriterias);
$jml_kriteria =count($kriteria);

```

Gambar 4.15 Kode Program Pengolahan Penilaian WSM

Data-data yang dihasilkan dari pengolahan diatas adalah:

- \$data digunakan untuk menyimpan data-data alternatif
- \$bobot digunakan untuk menyimpan data bobot kriteria
- \$kriteria digunakan untuk menyimpan data nama kriteria (berupa id)
- \$setnama digunakan untuk menyimpan nama alternatif

Setelah semua data didapatkan, abrulah dapat dilakukan perhitungan normalisasi kriteria.

Semua data kriteria adalah benefit sehingga normalisasi dilakukan dengan membagi nilai peserta setiap kriteria dengan nilai max dari kriteria tersebut.

Untuk itu terlebih dahulu perlu hitung nilai max (maksimal), berikut ini adalah kode programnya:

```

$maxkriteria = array();
$tempmax = array();
for ($n = 1; $n <= $jml_kriteria; $n++){
    $nol = "";
    if($n < 10){
        $nol = "0";
    }
    unset($tempmax);
    foreach ($data as $nama => $krit) {
        $tempmax[]=$krit["C".$nol.$n];
    }
    $maxkriteria["C".$nol.$n]=max($tempmax);
}

```

Gambar 4.16 Kode Program Mencari Nilai Max Untuk Setiap Kriteria

Kemudian setelah diketahui nilai max maka lakukan normalisasi untuk semua nilai alternatif. Kode program dapat dilihat pada Gambar 4.17.

```

$normalisasidata = array();
foreach ($data as $nama => $krit) {
    echo '<tr><td>'.$nama.'</td>';
    for ($n = 1; $n <= $jml_kriteria; $n++){
        $nol = "";
        if($n < 10){
            $nol = "0";
        }
        $temps=round(($krit["C".$nol.$n]/$maxkriteria["C".$nol.$n]),4);
        $normalisasidata[$nama]["C".$nol.$n]=$temps;
        echo '<td><div class="text-xs">C'.$nol.$n.' / Max[C'.$nol.$n.'] =</div>'.$temps.'</td>';
    }
    echo '</tr>';
}

```

Gambar 4.17 Kode Program Normalisasi Alternatif WSM

4. Melakukan normalisasi kriteria

Untuk melakukan normalisasi data kriteria, terlebih dahulu perlu menghitung jumlah bobot kriteria, berikut adalah kode programnya :

```

$jumlahbobot = 0;
foreach ($bobot as $k){
    echo "<td>$k</td>";
    $jumlahbobot = $jumlahbobot + $k;
}

```

Gambar 4.18 Kode Program Menghitung Jumlah Total Bobot

Menghitung normalisasi bobot kriteria dilakukan dengan cara bobot kriteria dibagi dengan jumlah bobot berikut adalah kode programnya

```

$bobotbaru = array();
$i = 0;
foreach ($bobot as $k) {
    (++$i);
    $nol = "";
    if($i < 10){
        $nol = "0";
    }
    $temps = $k / $jumlahbobot;
    $bobotbaru["C".$nol.$i] = $temps;
    echo "<td><div class='text-xs'>C".$nol.$i." / jml =</div>".round(($temps), 4)."</td>";
}

```

Gambar 4.19 Kode Program Normalisasi Bobot Kriteria WSM

5. Menghitung vektor S

Untuk menghitung vector s atau nilai WSM dilakukan dengan mengalikan hasil normalisasi nilai alternatif dengan nilai normalisasi bobot. Hasilnya kemudian dihitung total jumlahnya, nilai ini yang kemudian menjadi nilai WSM. Berikut ini adalah code programnya

```

$matrikws = array();
$hasilws = array();
$temp = 0;
$i = 0;
foreach ($normalisasi as $nama => $krit) {
    echo "<tr><td>".$nama."</td>";
    $temp = 0;
    foreach ($kriteria as $k) {
        $temps = $krit[$k] * $bobotbaru[$k];
        $hitung = $temps;
        $matrikws[$nama][$k] = $hitung;
        $temp = $temp + $hitung;
        echo "<td><div class='text-xs'> alt ".$nama." baru * bobot ".$k." baru =</div>".round(($temps), 4)."</td>";
    }
    $hasilws[$nama] = $temp;
    echo "<td><b>".round(($temp), 4)."</b></td></tr>";
}

```

Gambar 4.20 Kode Program Perhitungan Nilai WSM

6. Melakukan Perangkingan

```

$i = 0;
arsort($hasilws);
foreach($hasilws as $x=>$x_value) {
    echo "<tr>
        <td>" . (++$i) . "</td>
        <td>{$x}</td>
        <td>${setnama[$x]}</td>
        <td>".round(($x_value), 4)."</td>";
}

```

Gambar 4.21 Kode Program Perangkingan

7. Menghitung *hamming distance*

Untuk menghitung *hamming distance* yang digunakan untuk mengetahui tingkat ketidaksesuaian perangkingan dengan hasil perhitungan manual, maka langkah yang harus dilakukan adalah :

1. Merangking hasil perhitungan manual dan mencari guru yang memiliki nilai yang sama untuk ditetapkan sebagai rangking bersama.

Perangkingan

No	ID Peserta	Nama	Nilai
1	PS0021 (1)	Moch. Yusroni, S.Pd.	37
2	PS0014 (1)	Diah Anita Ludfiana, S.Pd.	37
3	PS0017 (1)	Yeni Yusella, S.Pd.	37
4	PS0022 (4)	Nihayatul Hasanah, S.Pd	36

Gambar 4.22 Contoh Perangkingan Bersama

Gambar diatas adalah contoh hasil perangkingan manual. Dalam hasil perangkingan atas terdapat rangking bersama yaitu no. 1, 2 dan 3 atau Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. merupakan rangking bersama karena memiliki nilai yang sama yaitu 37. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd jika dirangkingkan, maka mereka ada dirangking 1, 2 dan 3. Sedangkan Nihayatul Hasanah, S.Pd ada pada rangking 4.

Kode program untuk mengambil hasil perangkingan perhitungan manual seperti yang terdapat pada gambar 4.22 Pada kode program tersebut hasil perhitungan manual akan disimpan pada variabel `stmt4`.

```
$stmt4 = viewnilaidetailperangkingan();
```

Gambar 4.22 Kode Program Untuk Mengambil Hasil Perhitungan Manual

Setelah dilakukan pengambilan perangkingan manual, pada data perangkingan manual tersebut akan dicari guru yang memiliki nilai yang sama untuk ditetapkan sebagai rangking bersama. Kode program dapat dilihat pada gambar 4.23

```
$i = 0;
$ranking = 1 ;
$perban = -1;
while ($data2 = mysqli_fetch_assoc($stmt4)) {
    if($perban == -1){
        $perban = $data2['total'];
    }
    if($perban != $data2['total']){
        $ranking = $i+1;
        $perban = $data2['total'];
    }
    $simpan_hsl_manual[$i] = $data2['detail_peserta'];
    $simpan_n_manual[$data2['detail_peserta']] = $ranking;
    $i++;
};
```

Gambar 4.23 Kode Program Untuk Mencari Rangking Bersama

2. Mengambil hasil perangkingan dengan metode WSM

Hasil perangkingan ini yang akan dibandingkan dengan hasil prangkingan manual. Kode program untuk Mengambil hasil perangkingan dengan metode WSM adalah :

```
$stmt1 = viewsimpanhasil("wsm");
$i = 0;
while ($data2 = mysqli_fetch_assoc($stmt1)) {
    $simpan_hsl_wsm[$i] = $data2['id_peserta'];
    $i++;
};
```

Gambar 4.24 Kode Program Mengambil hasil perangkingan dengan metode

WSM

3. Mencari nilai *hamming distance*

Kode Program untuk mencari nilai *hamming distance* seperti yang ditunjukkan pada Gambar 4.25 dibawah ini :

```
function perbandingan($nilaimetode, $nilaimanual, $arraynilaimanual) {
    $tdksama = 0;
    $temptdksama = true;
    if($nilaimetode==$nilaimanual){
        $temptdksama = false;
        echo "<td><b>[\".$nilaimetode.\"]</b></td>";
    }else{
        echo "<td>";
        foreach($arraynilaimanual as $x=>$x_value) {
            //echo $x." ".$x_value."</br>";
            if($arraynilaimanual[$nilaimanual] == $x_value){
                if($arraynilaimanual[$nilaimetode] == $x_value){
                    echo "<div class='text-xs'>";
                    if($nilaimetode==$x){
                        $temptdksama = false;
                        echo "<b>[\".$nilaimetode.\" \"$x.\" \"$x_value.\"] </b> | ";
                    }else{
                        echo $nilaimetode." \"$x.\" \"$x_value.\" | ";
                    }
                    echo "</div>";
                }
            }
        }

        if($temptdksama == true){
            $tdksama++;
            echo $nilaimetode;
        }else{
            echo "<b>[\".$nilaimetode.\"]</b>";
        }
        echo "</td>";
    }
    return $tdksama;
}
```

Gambar 4.25 Kode Program *hamming distance*

Untuk mencari nilai *hamming distance* langkah-langkah yang dilakukan adalah :

- 1) Mengambil data guru yang berada pada rangking yang dicari pada hasil perangkingan manual.
- 2) Membandingkan dengan hasil pengakingan manual dengan hasil perangkingan metode WSM pada rangking yang dicari. Dengan ketentuan :
 - Jika sama, lanjut ke langkah 3 dan tidak dihitung sebagai ketidaksesuaian

- Jika tidak sama, maka dicek apakah pada hasil perangkingan manual pada rangking yang dicari merupakan rangking bersama.
 - Jika bukan merupakan rangking bersama, maka lanjut kelangkah 3 dan dihitung sebagai ketidaksesuaian
 - Jika merupakan rangking bersama, maka dicari pada rangking bersama apakah ada guru hasil rangking metode WSM yang sesuai.
 - Jika ada, maka tidak dihitung sebagai ketidaksesuaian dan lanjut ke langkah 3
 - Jika tidak ada, maka dihitung sebagai ketidaksesuaian dan lanjut ke langkah 3

3) Mencari untuk rangking selanjutnya

4. Selesai

4.3 Uji Coba Metode Weighted Sum Model (WSM)

Pengujian dilakukan banyak 6 kali pengujian. Pengujian tersebut dilakukan dengan perbedaan nilai bobot kriteria yang berbeda. Adapun pengujian dengan perbedaan nilai bobot kriteria seperti ditampilkan pada tabel 4.1 berikut:

Tabel 4.1 Pengujian WSM

Penguji	Kriteria													
	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	5	4	4	4	4	3	3	3	3	2	2	2	2	2
5	2	2	2	2	2	3	3	3	3	4	4	4	4	5
6	3	3	2	4	4	3	5	2	4	3	3	2	2	2

Keterangan tabel adalah

C01 = Mengetahui karakteristik peserta didik,

C02 = Mengetahui teori belajar dan prinsip-prinsip pembelajaran yang mendidik,

C03 = Pengembangan kurikulum,

C04 = Kegiatan pembelajaran yang mendidik,

C05 = Mengembangkan potensi peserta didik,

C06 = Komunikasi dengan peserta didik,

C07 = Penilaian dan evaluasi,

C08 = Bertindak sesuai dengan norma agama, hukum, sosial dan kebudayaan nasional,

C09 = Menunjukkan pribadi yang dewasa dan teladan,

C10 = Etos kerja, tanggung jawab yang tinggi, rasa bangga menjadi guru,

C11 = Bersikap inklusif, bertindak obyektif, serta tidak diskriminatif,

C12 = Komunikasi dengan sesama guru, tenaga kependidikan, orang tua, peserta didik, dan masyarakat,

C13 = Penguasaan materi, struktur, konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu, dan

C14 = dan mengembangkan keprofesionalan melalui tindakan yang reflektif.

Hasil perbandingan dari pengujian kemudian dihitung kevalidasinya berdasarkan nilai ketidaksesuaian perbandingan dengan hasil perhitungan manual dengan algoritma *hamming distance*. Nilai algoritma *hamming distance* yang semakin besar menandakan semakin besar berbeda dengan hasil perhitungan manual atau semakin banyak yang perbandingannya tidak sesuai.

Proses pengujian adalah sebagai berikut:

1. Pengujian 1

Pada pengujian 1 ini menggunakan bobot kriteria terendah untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 4.2 dibawah ini:

Tabel 4.2 Bobot Kriteria Pengujian 1 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel

4.3.

Tabel 4.3 Hasil Pengujian 1 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	2
2	0
3	0
4	0
5	0
Rata-rata	0,4

Hasil pengujian 1 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 0 kesalahan, nilai ketidaksesuaian tertinggi adalah 2 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 0,4 kesalahan.

2. Pengujian 2

Pada pengujian 2 ini menggunakan bobot kriteria tertinggi untuk semua kriteria.

Bobot-bobot kriteria tersebut seperti yang ditunjukkan pada Tabel 4.4.

Tabel 4.4 Bobot Kriteria Pengujian 2 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Hasil Pengujian 2 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	0
2	0
3	0
4	2
5	2
Rata-rata	0,8

Hasil pengujian 2 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 0 kesalahan, nilai ketidaksesuaian tertinggi adalah 2 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 0,8 kesalahan.

3. Pengujian 3

Pada pengujian 3 ini menggunakan bobot kriteria nilai 5 (pertengahan) untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 4.6.

Tabel 4.6 Bobot Kriteria Pengujian 3 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 4.7.

Tabel 4.7 Hasil Pengujian 3 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	0
2	2
3	2
4	2
5	0
Rata-rata	1,2

Hasil pengujian 3 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 0 kesalahan, nilai ketidaksesuaian tertinggi adalah 2 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 1,2 kesalahan.

4. Pengujian 4

Pada pengujian 4 ini menggunakan bobot kriteria dari tinggi ke rendah. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 4.8.

Tabel 4.8 Bobot Kriteria Pengujian 4 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	5	4	4	4	4	3	3	3	3	2	2	2	2	2

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 4.9.

Tabel 4.9 Hasil Pengujian 4 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	7
2	6
3	11
4	7
5	9
Rata-rata	8

Hasil pengujian 4 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 6 kesalahan, nilai ketidaksesuaian tertinggi adalah 11 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 8 kesalahan.

5. Pengujian 5

Pada pengujian 5 ini menggunakan bobot kriteria dari nilai rendah ke tinggi Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 4.10.

Tabel 4.10 Bobot Kriteria Pengujian 5 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	2	2	2	2	2	3	3	3	3	4	4	4	4	5

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil Pengujian 5 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	9
2	9
3	7
4	10
5	8
Rata-rata	8,6

Hasil pengujian 5 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 7 kesalahan, nilai ketidaksesuaian tertinggi adalah 10 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 8.6 kesalahan.

6. Pengujian 6

Pada pengujian 6 ini menggunakan bobot kriteria yang acak. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 4.12.

Tabel 4.12 Bobot Kriteria Pengujian 6 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	3	3	2	4	4	3	5	2	4	3	3	2	2	2

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 4.13.

Tabel 4.13 Hasil Pengujian 6 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	12
2	8
3	11
4	9
5	11
Rata-rata	10,2

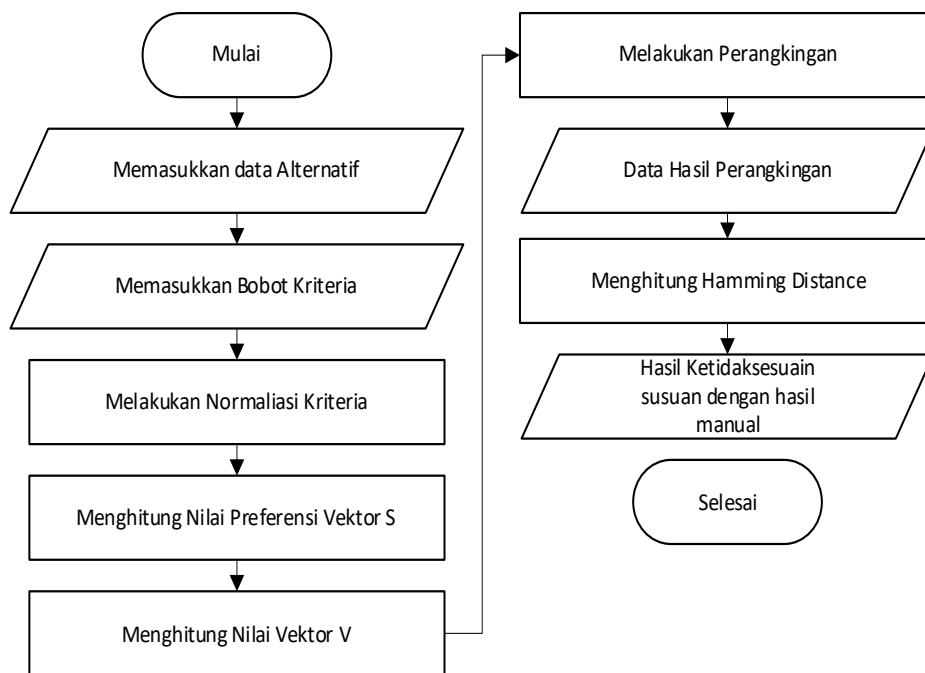
Hasil pengujian 6 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 8 kesalahan, nilai ketidaksesuaian tertinggi adalah 12 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 10.2 kesalahan.

BAB V

METODE WEIGHTED PRODUCT MODEL (WPM)

5.1 Desain Metode Weighted Product Model (WPM)

Desain metode *weighted product model* (WPM) terdapat di Gambar 5.1. Desain WPM tersebut terdiri dari alur proses perhitungan WPM, perangkingan hasil WPM dan hasil perbandingan ketidakesesuaian hasil Hamming Distance.



Gambar 5.1 Desain Metode Weighted Product Model (WPM)

5.2 Implementasi Metode Weighted Product Model (WPM)

Metode WPM akan diimplementasikan dengan memakai bahasa pemrograman PHP dan dan *database management sistem* yang digunakan adalah MySQL. Desain database seperti pada Gambar 3.2. Berikut ini adalah tahapan dalam pengimplementasian metode WPM:

1. Memasukkan data alternatif

Data alternatif terdiri dari data peserta dan nilai peserta. Nilai peserta terdiri dari 14 kompetensi. Nilai 14 kompetensi tersebut adalah hasil perhitungan dari poin-poin penilaian dimasing-masing kompetensi. Pada poin penilaian tersebut dinilai dengan 3 kategori yaitu

- Nilai 0 untuk tidak terpenuhi
- Nilai 1 untuk terpenuhi Sebagian
- Nilai 2 untuk seleuruhnya terpenuhi

Dibawah ini adalah tampilan form untuk memasukkan data alternatif

The screenshot shows a web browser window with the URL `localhost/topsis-master/mdcmfinal/penilaian_add.php`. The page title is "Silahkan Isi Untuk Melakukan Penilaian :". The form contains the following fields:

- ID Penilaian:** A text input field with a placeholder "Silahkan isi dengan ID Penilaian".
- Penguji:** A dropdown menu with the option "Pilih Penguji".
- Peserta:** A dropdown menu with the option "Pilih Peserta".
- Periode Penilaian:** A dropdown menu with the option "Pilih Periode".

Below these fields is a section titled "Penilaian (Berdasarkan Kompetensi)" with a navigation bar for 14 competencies. The first competency, "Kompetensi 1 : Mengenal Karakteristik Peserta Didik", is expanded to show 6 indicators, each with a dropdown menu for rating (0 = Tidak Terpenuhi, 1 = Sebagian, 2 = Terpenuhi).

Gambar 5.2 Form Input Nilai Peserta

Dari form tersebut kemudian poin-poin penilaian dihitung menggunakan rumus:

- Skor maksimum = jumlah indikator x 2
- Persentase = $(\text{total skor} / \text{skor maksimum}) \times 100 \%$

Kode program untuk melakukan perhitungan diatas adalah

```

$temp = 0;
for($f=$simpantiti2[$ttkawal]; $f < $simpantiti2[$ttkakhir]; $f++){
    $temp=$temp+$simpannilaipoinkomp[$f];
}
$simpanpersenkomp[$i] = round(($temp/($simpantiti2[$ttkakhir]-$simpantiti2[$ttkawal])*2)*100, 0);
$simpanakhirkomp[$i] = viewpersenkomp($simpanpersenkomp[$i]);
$total = $total + $simpanakhirkomp[$i];

```

Gambar 5.3 Kode Program Persentase Nilai Peserta

Setelah persentase didapatkan maka dilakukan pengkategorian dengan kondisi:

```

function viewpersenkomp($persen) {
    $hasil = 0;
    if($persen >= 0 && $persen <= 25){
        $hasil = 1;
    }else if($persen > 25 && $persen <= 50){
        $hasil = 2;
    }else if($persen > 50 && $persen <= 75){
        $hasil = 3;
    }else if($persen > 75 && $persen <= 100){
        $hasil = 4;
    }
    return $hasil;
}

```

Gambar 5.4 Kode Program Pengkategorian Nilai Peserta

Sehingga didapatkan nilai untuk 14 kompetensi tersebut dengan rentang nilai antara 1 – 4. Nilai inilah yang akan menjadi nilai dari 14 kompetensi yang dimiliki setiap alternatif. Data alternatif disimpan pada tabel “nilai_peserta”.

Berikut adalah kode program untuk melakukan penyimpanan nilai tersebut


```

addnilai($detail_peserta, 'C01', $simpanakhirkomp[0], $detail_periode);
addnilai($detail_peserta, 'C02', $simpanakhirkomp[1], $detail_periode);
addnilai($detail_peserta, 'C03', $simpanakhirkomp[2], $detail_periode);
addnilai($detail_peserta, 'C04', $simpanakhirkomp[3], $detail_periode);
addnilai($detail_peserta, 'C05', $simpanakhirkomp[4], $detail_periode);
addnilai($detail_peserta, 'C06', $simpanakhirkomp[5], $detail_periode);
addnilai($detail_peserta, 'C07', $simpanakhirkomp[6], $detail_periode);
addnilai($detail_peserta, 'C08', $simpanakhirkomp[7], $detail_periode);
addnilai($detail_peserta, 'C09', $simpanakhirkomp[8], $detail_periode);
addnilai($detail_peserta, 'C10', $simpanakhirkomp[9], $detail_periode);
addnilai($detail_peserta, 'C11', $simpanakhirkomp[10], $detail_periode);
addnilai($detail_peserta, 'C12', $simpanakhirkomp[11], $detail_periode);
addnilai($detail_peserta, 'C13', $simpanakhirkomp[12], $detail_periode);
addnilai($detail_peserta, 'C14', $simpanakhirkomp[13], $detail_periode);

```

Gambar 5.5 Kode Program Penyimpan Nilai Peserta

Kode program function “addnilai” yang berisi query untuk menyimpan ke database

```

function addnilai($aidpeserta, $aidbobot, $anilai, $aperiode) {
    include('config.php');
    $query = "INSERT INTO nilai_peserta (id_peserta, id_bobot, nilai, id_periode) "
        . "VALUES('$aidpeserta', '$aidbobot', '$anilai', '$aperiode)";
    mysqli_query($db, $query);
    return mysqli_affected_rows($db);
}

```

Gambar 5.6 Kode Program Function Query Nilai Peserta

2. Memasukkan bobot kriteria

Dalam menentukan bobot pada setiap kriteria tidak ada aturan baku untuk menentukan kriteria mana yang menempati bobot tertinggi atau bobot terendah. Pada penelitian ini penulis mengasumsikan memberikan pembobotan pada kriteria berdasarkan acuan atau asumsi pribadi sebagai seorang DM (Decision Maker) / (PK) Pengambil Keputusan dengan melihat indikator pada masing-masing kriteria bahwa semakin tinggi nilai kompetensi yang diberikan oleh DM maka semakin tinggi bobot kepentingannya, sehingga didapatkan hasil sebagai berikut;

Nilai/Bobot	Tingkat Kepentingan	Kriteria/Kompetensi
1	Sangat Rendah	-
2	Rendah	C08, C09, C11, C13, C14
3	Cukup	C03, C04, C10
4	Tinggi	C02, C07, C12
5	Sangat Tinggi	C01, C05, C06

Dikarenakannya tidak adanya aturan baku pada penentuan bobot kriteria maka penulis memberikan form isian untuk menentukan bobot masing-masing kriteria berdasarkan pada asumsi dari DM (Decision Maker) / PK (Pengambil Keputusan).

Berikut adalah kode program untuk mengatur bobot dari kriteria.

```

$book = ambileditbobot($_GET['kode']);
$edit = mysqli_fetch_assoc($book);
$dafkom = viewdafkom();
?>
<div class="row">
  <div class="col-auto mr-auto"><h4>Silahkan isi untuk mengubah kriteria dengan ID :<?php echo $edit['bobot_id'];?></h4></div>
  <div class="col-auto"><a href="bobot_view.php" class="btn btn-info">Lihat Data</a></div>
</div>
<form method="post">
  <div class="form-group">
    <label>ID Kriteria</label>
    <input type="text" class="form-control" name="id" value="<?php echo $edit['bobot_id'];?>" disabled/>
    <input type="hidden" name="id" value="<?php echo $edit['bobot_id'];?>" />
  </div>
  <div class="form-group">
    <label>Kriteria</label>
    <input type="text" class="form-control" name="nama" value="<?php echo $dafkom[$_GET['komp']];?>" disabled/>
  </div>
  <div class="form-group">
    <label>Bobot</label>
    <select class="form-control" name="nilai">
      <option value="1" <?php if($edit['bobot_nilai'] == '1'){ echo 'selected'; }?>>Sangat Rendah</option>
      <option value="2" <?php if($edit['bobot_nilai'] == '2'){ echo 'selected'; }?>>Rendah</option>
      <option value="3" <?php if($edit['bobot_nilai'] == '3'){ echo 'selected'; }?>>Cukup</option>
      <option value="4" <?php if($edit['bobot_nilai'] == '4'){ echo 'selected'; }?>>Tinggi</option>
      <option value="5" <?php if($edit['bobot_nilai'] == '5'){ echo 'selected'; }?>>Sangat Tinggi</option>
    </select>
  </div>
  <button type="submit" class="btn btn-primary" name="kirim">
    Ubah
  </button>
</form>

```

Gambar 5.7 Kode Program Form Bobot Kriteria

Berikut ini adalah tampilan dari form untuk mengatur bobot kriteria

The screenshot shows a web application interface for setting criteria weights. The interface is titled "SD ANAK SALEH" and shows a form for editing criteria with ID "C01". The criteria is "Kompetensi 1 : Mengenal Karakteristik Peserta Didik". The weight is set to "Sangat Tinggi".

Gambar 5.8 Tampilan Form Bobot Kriteria

Form ini akan mengirim data melalui kode program dibawah ini:

```

if (isset($_POST['kirim'])) {
    $id = $_POST['id'];
    $nilai = $_POST['nilai'];
    $nilai2 = 0;
    if($nilai == 1){
        $nilai2 = 1;
    }else if($nilai == 2){
        $nilai2 = 3;
    }else if($nilai == 3){
        $nilai2 = 5;
    }else if($nilai == 4){
        $nilai2 = 7;
    }else if($nilai == 5){
        $nilai2 = 9;
    }
    $success = updatebobot($id, $nilai, $nilai2);
    if ($success > 0) {
        header('Location: bobot_view.php');
    } else {
        echo ' <div class="alert alert-danger" role="alert">
            <strong>Maaf</strong>, data gagal diinputkan, silahkan coba inputkan kembali.
            </div>';
    }
}

```

Gambar 5.9 Kode Program Bobot Kriteria

Nilai pembobotan dari kriteria tersebut akan disimpan ke database dengan memanggil function “updatebobot” melalui kode program dibawah ini

```
“$success = updatebobot($id, $nilai, $nilai2);”
```

Gambar 5.10 Kode Program Memanggil Function Ganti Bobot Kriteria

Dibawah ini adalah isi function “updatebobot” yang berfungsi untuk menyimpan data ke database yaitu table “bobot”

```
function updatebobot($aid, $anilai, $anilai2) {
    include('config.php');
    $query = "UPDATE bobot "
            . "SET bobot_nilai='$anilai', bobot_ahp='$anilai2' "
            . "WHERE bobot_id='$aid'";
    mysqli_query($db,$query);
    return mysqli_affected_rows($db);
}
```

Gambar 5.11 Kode Program Function Query Ganti Bobot Kriteria

3. Melakukan normalisasi kriteria

Pertama dilakukan pengambilan data yang dibutuhkan dari database dengan kode program seperti pada Gambar 5.12

```
$stampil = viewpenilaian();
```

Gambar 5.12 Kode Program Memanggil Penilaian WPM

Kode diatas akan menjalankan kode program Gambar 5.13 yang berisi query database.

```
function viewpenilaian(){
    include('config.php');
    $query = "SELECT b.peserta_id, b.peserta_nama, c.bobot_id, a.nilai, c.bobot_nilai
            FROM nilai_peserta a
            LEFT JOIN peserta b ON a.id_peserta = b.peserta_id
            LEFT JOIN bobot c ON a.id_bobot = c.bobot_id";
    $result = mysqli_query($db,$query);
    return $result;
}
```

Gambar 5.14 Kode Program Memanggil Penilaian WPM

Data hasil query tersebut kemudian diolah untuk memisahkan data kriteria dan data bobot dengan kode program pada Gambar 5.15

```

$data      =array();
$kriterias =array();
$bobot     =array();
$setnama   =array();
if ($tampil) {
    while($row=$tampil->fetch_object()){
        if(!isset($data[$row->peserta_id])){
            $data[$row->peserta_id]=array();
        }
        if(!isset($data[$row->peserta_id][$row->bobot_id])){
            $data[$row->peserta_id][$row->bobot_id]=array();
        }
        $bobot[$row->bobot_id]=$row->bobot_nilai;
        $data[$row->peserta_id][$row->bobot_id]=$row->nilai;
        $kriterias[]=$row->bobot_id;
        $setnama[$row->peserta_id]=$row->peserta_nama;
    }
}
$kriteria   =array_unique($kriterias);
$jml_kriteria =count($kriteria);

```

Gambar 5.15 Kode Program Pengolahan Penilaian WPM

Data-data yang dihasilkan dari pengolahan diatas adalah:

- \$data digunakan untuk menyimpan data-data alternatif
- \$bobot digunakan untuk menyimpan data bobot kriteria
- \$kriteria digunakan untuk menyimpan data nama kriteria (berupa id)
- \$setnama digunakan untuk menyimpan nama alternatif

Setelah semua data didapatkan, abruilah dapat dilakukan perhitungan normalisasi kriteria.

Untuk mlakukan normalisasi data kriteria, terlebih dahulu perlu menghitung jumlah bobot kriteria, berikut adalah kode programnya:

```

$jumlahbobot = 0;
foreach ($bobot as $k){
    echo "<td>$k</td>";
    $jumlahbobot = $jumlahbobot + $k;
}

```

Gambar 5.16 Kode Program Penjumlahan Bobot KriteriaWPM

Menghitung normalisasi dengan cara bobot kriteria dibagi dengan jumlah bobot berikut adalah kode programnya

```

$bobotbaru = array();
$i = 0;
foreach ($bobot as $k) {
    (++$i);
    $nol = "";
    if($i < 10){
        $nol = "0";
    }
    $temps = $k / $jumlahbobot;
    $bobotbaru[] = $temps;
    echo "<td><div class='text-xs'>C".$nol.$i."/ jml =</div>".round(($temps), 4)."</td>";
}

```

Gambar 5.17 Kode Program Normalisasi Bobot Kriteria WPM

4. Menghitung nilai vektor S

Untuk tahap ini, pertama kali harus melakukan pemangkat setiap nilai alternatif dengan hasil normalisasi bobot, berikut ini adalah kode programnya

```

$vektor_s = Array();
foreach ($data as $nama => $krit) {
    $i = 0;
    foreach ($kriteria as $k) {
        $vektor_s[$nama][$k] = pow($krit[$k], $bobotbaru[$i]);
        ++$i;
    }
}

```

Gambar 5.18 Kode Program Pemangkat Alternatif WPM

Kemudian menghitung jumlah untuk setiap alternatif dan dilakukan perhitungan nilai preferensi vektor S.

```

$jml_vektor_s = Array();
$total_vs = 0;

foreach ($data as $nama => $krit) {
    echo "<tr><td>".$nama."</td>";
    $tempvs = 1;
    foreach ($kriteria as $k) {
        echo "<td><div class='text-xs'>".$nama."^".$k." =</div>".round($vektor_s[$nama][$k], 4)."</td>";
        $tempvs = ($tempvs * $vektor_s[$nama][$k]);
    }
    echo "<td><b>".round($tempvs, 4)."</b></td></tr>";
    $jml_vektor_s[$nama] = $tempvs;
    $total_vs = $total_vs + $tempvs;
}

```

Gambar 5.19 Kode Program Nilai Vektor S WPM

5. Menghitung nilai vektor V

Nilai vektor v atau nilai WPM didapat dari nilai s untuk setiap alternatif dibagi jumlah nilai s.

```
$vektor_v = Array();
foreach ($data as $nama => $krit) {
echo '<tr><td>'. $nama. '</td>';
$temp_s = ($jml_vektor_s[$nama]/$total_vs);
$vektor_v[$nama] = $temp_s;
echo "<td><div class='text-xs'> s ". $nama. "/ jumlah s =</div><b>".round($temp_s, 4). "</b></td></tr>";
}
```

Gambar 5.20 Kode Program Nilai WPM

6. Melakukan perangkian

```
$i = 0;
arsort($vektor_v);
foreach($vektor_v as $x=>$x_value) {
echo "<tr>
    <td>" . (++$i) . "</td>
    <td>{$x}</td>
    <td>${setnama[$x]}</td>
    <td>".round($x_value, 4). "</td>";
}
?>
```

Gambar 5.21 Kode Program Perangkian

7. Menghitung *hamming distance*

Untuk menghitung *hamming distance* yang digunakan untuk mengetahui tingkat ketidaksesuaian perangkian dengan hasil perhitungan manual, maka langkah yang harus dilakukan adalah:

1. Merangking hasil perhitungan manual dan mencari guru yang memiliki nilai yang sama untuk ditetapkan sebagai rangking bersama.

Perangkingan

No	ID Peserta	Nama	Nilai
1	PS0021 (1)	Moch. Yusroni, S.Pd.	37
2	PS0014 (1)	Diah Anita Ludfiana, S.Pd.	37
3	PS0017 (1)	Yeni Yusella, S.Pd.	37
4	PS0022 (4)	Nihayatul Hasanah, S.Pd	36

Gambar 5.22 Contoh Perangkingan Bersama

Gambar diatas adalah contoh hasil perangkingan manual. Dalam hasil perangkingan atas terdapat rangking bersama yaitu no. 1, 2 dan 3 atau Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. merupakan rangking bersama karena memiliki nilai yang sama yaitu 37. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd jika dirangkingkan, maka mereka ada dirangking 1, 2 dan 3. Sedangkan Nihayatul Hasanah, S.Pd ada pada rangking 4.

Kode program untuk mengambil hasil perangkingan perhitungan manual seperti yang terdapat pada gambar 5.23 Pada kode program tersebut hasil perhitungan manual akan disimpan pada variabel stmt4.

```
stmt4 = viewnilaidetailperangkingan();
```

Gambar 5.23 Kode Program Untuk Mengambil Hasil Perhitungan Manual

Setelah dilakukan pengambilan perangkingan manual, pada data perangkingan manual tersebut akan dicari guru yang memiliki nilai yang sama untuk ditetapkan sebagai rangking bersama. Kode program dapat dilihat pada gambar 5.24


```

$i = 0;
$ranking = 1 ;
$perban = -1;
while ($data2 = mysqli_fetch_assoc($stmt4)) {
    if($perban == -1){
        $perban = $data2['total'];
    }
    if($perban != $data2['total']){
        $ranking = $i+1;
        $perban = $data2['total'];
    }
    $simpan_hsl_manual[$i] = $data2['detail_peserta'];
    $simpan_n_manual[$data2['detail_peserta']] = $ranking;
    $i++;
};

```

Gambar 5.24 Kode Program Untuk Mencari Rangkaing Bersama

2. Mengambil hasil perangkingan dengan metode WPM

Hasil perangkingan ini yang akan dibandingkan dengan hasil prangkingan manual. Kode program untuk Mengambil hasil perangkingan dengan metode WPM adalah :

```

$stmt2 = viewsimpanhasil("wpm");
while ($data2 = mysqli_fetch_assoc($stmt2)) {
    $simpan_hsl_wpm[$i] = $data2['id_peserta'];
    $i++;
};

```

Gambar 5.25 Kode Program Mengambil hasil perangkingan dengan metode

WPM

3. Mencari nilai *hamming distance*

Kode Program untuk mencari nilai *hamming distance* seperti yang ditunjukkan pada Gambar 5.26 dibawah ini:

```

function perbandingan($nilaimetode, $nilaimanual, $arraynilaimanual) {
    $tdksama = 0;
    $temptdksama = true;
    if($nilaimetode==$nilaimanual){
        $temptdksama = false;
        echo "<td><b>[\".$nilaimetode.\"]</b></td>";
    }else{
        echo "<td>";
        foreach($arraynilaimanual as $x=>$x_value) {
            //echo $x." ".$x_value."</br>";
            if($arraynilaimanual[$nilaimanual] == $x_value){
                if($arraynilaimanual[$nilaimetode] == $x_value){
                    echo "<div class='text-xs'>";
                    if($nilaimetode==$x){
                        $temptdksama = false;
                        echo "<b>[\".$nilaimetode.\" ".$x.\" ".$x_value.\"] </b> | ";
                    }else{
                        echo $nilaimetode." ".$x." ".$x_value." | ";
                    }
                    echo "</div>";
                }
            }
        }

        if($temptdksama == true){
            $tdksama++;
            echo $nilaimetode;
        }else{
            echo "<b>[\".$nilaimetode.\"]</b>";
        }
        echo "</td>";
    }
    return $tdksama;
}

```

Gambar 5.26 Kode Program *hamming distance*

Untuk mencari nilai *hamming distance* langkah-langkah yang dilakukan adalah :

- 1) Mengambil data guru yang berada pada rangking yang dicari pada hasil perangkingan manual.
- 2) Membandingkan dengan hasil pengakingan manual dengan hasil perangkingan metode WPM pada rangking yang dicari. Dengan ketentuan :
 - Jika sama, lanjut ke langkah 3 dan tidak dihitung sebagai ketidaksesuaian
 - Jika tidak sama, maka dicek apakah pada hasil perangkingan manual pada rangking yang dicari merupakan rangking bersama.

- Jika bukan merupakan rangking bersama, maka lanjut kelangkah 3 dan dihitung sebagai ketidaksesuaian
- Jika merupakan rangking bersama, maka dicari pada rangking bersama apakah ada guru hasil rangking metode WPM yang sesuai.
 - Jika ada, maka tidak dihitung sebagai ketidaksesuaian dan lanjut ke langkah 3
 - Jika tidak ada, maka dihitung sebagai ketidaksesuaian dan lanjut ke langkah 3

3) Mencari untuk rangking selanjutnya

4. Selesai.

5.3 Uji Coba Metode Weighted Product Model (WPM)

Pengujian dilakukan banyak 6 kali pengujian. Pengujian tersebut dilakukan dengan perbedaan nilai bobot kriteria yang berbeda seperti ditampilkan pada tabel 5.1.

Tabel 5.1 Pengujian WPM

Pengujia	Kriteria													
	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	5	4	4	4	4	3	3	3	3	2	2	2	2	2
5	2	2	2	2	2	3	3	3	3	4	4	4	4	5
6	3	3	2	4	4	3	5	2	4	3	3	2	2	2

Keterangan tabel adalah

C01 = Mengenal karakteristik peserta didik,

C02 = Menguasai teori belajar dan prinsip-prinsip pembelajaran yang mendidik,

C03 = Pengembangan kurikulum,

C04 = Kegiatan pembelajaran yang mendidik,

C05 = Mengembangkan potensi peserta didik,

C06 = Komunikasi dengan peserta didik,

C07 = Penilaian dan evaluasi,

C08 = Bertindak sesuai dengan norma agama, hukum, sosial dan kebudayaan nasional,

C09 = Menunjukkan pribadi yang dewasa dan teladan,

C10 = Etos kerja, tanggung jawab yang tinggi, rasa bangga menjadi guru,

C11 = Bersikap inklusif, bertindak obyektif, serta tidak diskriminatif,

C12 = Komunikasi dengan sesama guru, tenaga kependidikan, orang tua, peserta didik, dan masyarakat,

C13 = Penguasaan materi, struktur, konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu, dan

C14 = dan mengembangkan keprofesionalan melalui tindakan yang reflektif.

Hasil perangkingan dari pengujian kemudian dihitung kevalidasinya berdasarkan nilai ketidaksesuaian perangkingan dengan hasil perhitungan manual dengan algoritma *hamming distance*. Nilai algoritma *hamming distance* yang semakin besar menandakan semakin besar berbeda dengan hasil perhitungan manual atau semakin banyak yang perangkingannya tidak sesuai.

Proses pengujian adalah sebagai berikut:

1. Pengujian 1

Pada pengujian 1 ini menggunakan bobot kriteria terendah untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 5.2 dibawah ini:

Tabel 5.2 Bobot Kriteria Pengujian 1 WPM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 5.3.

Tabel 5.3 Hasil Pengujian 1 WPM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WPM
1	10
2	8
3	6
4	10
5	12
Rata-rata	9,2

Hasil pengujian 1 untuk metode WPM ini menghasilkan nilai ketidaksesuaian terendah adalah 8 kesalahan, nilai ketidaksesuaian tertinggi adalah 6 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 9.2 kesalahan.

2. Pengujian 2

Pada pengujian 2 ini menggunakan bobot kriteria tertinggi untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan pada Tabel 5.4.

Tabel 5.4 Bobot Kriteria Pengujian 2 WPM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 5.5.

Tabel 5.5 Hasil Pengujian 2 WPM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WPM
1	6
2	11
3	2
4	12
5	8
Rata-rata	7,8

Hasil pengujian 2 untuk metode WPM ini menghasilkan nilai ketidaksesuaian terendah adalah 2 kesalahan, nilai ketidaksesuaian tertinggi adalah 12 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 7,8 kesalahan.

3. Pengujian 3

Pada pengujian 3 ini menggunakan bobot kriteria nilai 5 (pertengahan) untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan pada Tabel 5.6.

Tabel 5.6 Bobot Kriteria Pengujian 3 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 5.7.

Tabel 5.7 Hasil Pengujian 3 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	9
2	11
3	4
4	10
5	9
Rata-rata	8,6

Hasil pengujian 3 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 4 kesalahan, nilai ketidaksesuaian tertinggi adalah 11 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 8,6 kesalahan.

4. Pengujian 4

Pada pengujian 4 ini menggunakan bobot kriteria dari tinggi ke rendah. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 5.8.

Tabel 5.8 Bobot Kriteria Pengujian 4 WSM

	C	C	C	C	C	C	C	C	C	C	C	C	C	C
	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Bobot	5	4	4	4	4	3	3	3	3	2	2	2	2	2

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 5.9.

Tabel 5.9 Hasil Pengujian 4 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	11
2	13
3	16
4	12
5	12
Rata-rata	12,8

Hasil pengujian 4 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 11 kesalahan, nilai ketidaksesuaian tertinggi adalah 16 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 12,8 kesalahan.

5. Pengujian 5

Pada pengujian 5 ini menggunakan bobot kriteria dari nilai rendah ke tinggi Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 5.10.

Tabel 5.10 Bobot Kriteria Pengujian 5 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	2	2	2	2	2	3	3	3	3	4	4	4	4	5

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 5.11.

Tabel 5.11 Hasil Pengujian 5 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	11
2	12
3	13
4	15
5	14
Rata-rata	13

Hasil pengujian 5 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 11 kesalahan, nilai ketidaksesuaian tertinggi adalah 15 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 13 kesalahan.

6. Pengujian 6

Pada pengujian 6 ini menggunakan bobot kriteria yang acak. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 5.12.

Tabel 5.12 Bobot Kriteria Pengujian 6 WSM

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	3	3	2	4	4	3	5	2	4	3	3	2	2	2

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 5.13.

Tabel 5.13 Hasil Pengujian 6 WSM

Uji Coba	Hasil Nilai Hamming Distance untuk Metode WSM
1	15
2	12
3	14
4	13
5	13
Rata-rata	13,4

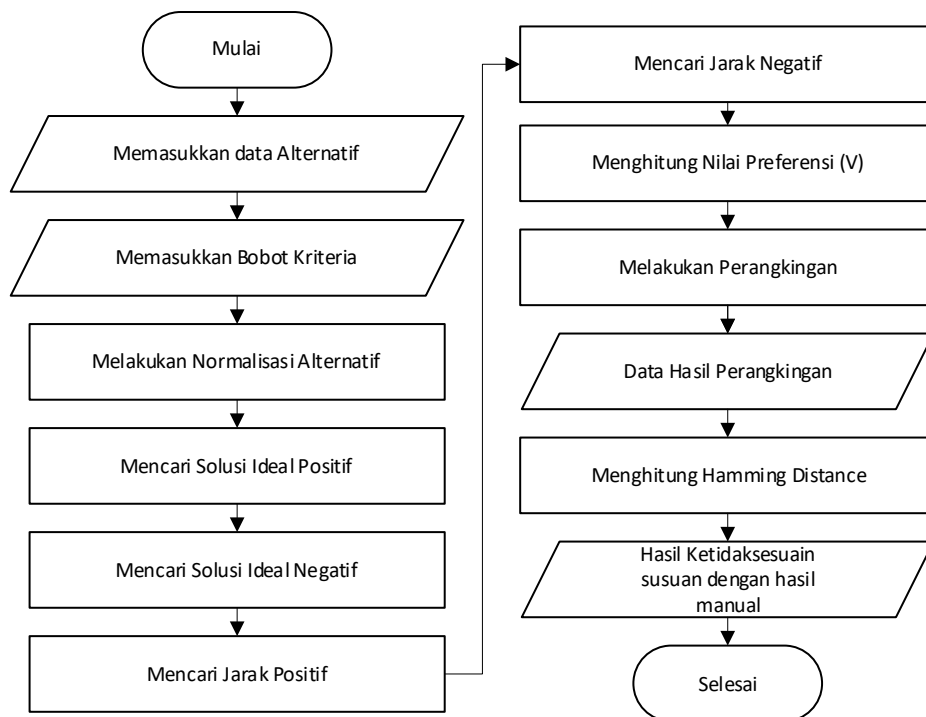
Hasil pengujian 6 untuk metode WSM ini menghasilkan nilai ketidaksesuaian terendah adalah 12 kesalahan, nilai ketidaksesuaian tertinggi adalah 15 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 13.4 kesalahan.

BAB VI

METODE TECHNIQUE FOR OTHERS PREFERENCE BY SIMILARITY TO IDEAL SOLUTION (TOPSIS)

6.1 Desain Metode Technique For Others Preference by Similarity to Ideal Solution (TOPSIS)

Desain metode Metode *technique for others preference by similarity to ideal solution* (TOPSIS) terdapat di Gambar 6.1. Desain TOPSIS tersebut terdiri dari alur proses perhitungan TOPSIS, perangkingan hasil TOPSIS dan hasil perbandingan ketidaksesuaian hasil *hamming distance*.



Gambar 6.1 Desain Technique For Others Preference by Similarity to Ideal
Solution (TOPSIS)

6.2 Implementasi Metode Technique For Others Preference by Similarity to Ideal Solution (TOPSIS)

Metode WSM akan diimplementasikan dengan memakai bahasa pemrograman PHP dan *database management sistem* yang digunakan adalah MySQL. Desain database seperti pada Gambar 3.2. Berikut ini adalah tahapan dalam pengimplementasian metode TOPSIS:

1. Memasukkan data alternatif

Data alternatif terdiri dari data peserta dan nilai peserta. Nilai peserta terdiri dari 14 kompetensi. Nilai 14 kompetensi tersebut adalah hasil perhitungan dari poin-poin penilaian dimasing-masing kompetensi. Pada poin penilaian tersebut dinilai dengan 3 kategori yaitu

- Nilai 0 untuk tidak terpenuhi
- Nilai 1 untuk terpenuhi Sebagian
- Nilai 2 untuk seleuruhnya terpenuhi

Dibawah ini adalah tampilan form untuk memasukkan data alternatif

The screenshot shows a web browser window with a URL of localhost:topsis-master/midcomfinal/penilaian_add.php. The page title is 'Silahkan Isi Untuk Melakukan Penilaian :'. The form contains the following fields:

- ID Penilaian:** A text input field with a note 'Silahkan isi dengan ID Penilaian'.
- Waktu:** A text input field with a dropdown menu for 'Pilih Waktu'.
- Peserta:** A dropdown menu for 'Pilih Peserta'.
- Periode Penilaian:** A dropdown menu for 'Pilih Periode'.
- Penilaian (Berdasarkan Kompetensi):** A table with 14 columns. The first column is 'Kompetensi 1 : Mengenal Karakteristik Peserta Didik'. The table contains 6 rows of assessment items, each with a dropdown menu for the score (0 = Tidak Terpenuhi).

At the bottom of the form, there are 'Simpan' and 'Batal' buttons.

Gambar 6.2 Form Input Nilai Peserta

Dari form tersebut kemudian poin-poin penilaian dihitung menggunakan rumus :

- Skor maksimum = jumlah indikator x 2
- Persentase = (total skor / skor maksimum) x 100 %

Kode program untuk melakukan perhitungan diatas adalah

```

$temp = 0;
for($f=$simpantiti2[$ttkawal]; $f < $simpantiti2[$ttkakhir]; $f++){
    $temp=$temp+$simpannilaipoinkomp[$f];
}
$simpanpersenkomp[$i] = round(($temp/($simpantiti2[$ttkakhir]-$simpantiti2[$ttkawal])*2)*100, 0);
$simpanakhirkomp[$i] = viewpersenkomp($simpanpersenkomp[$i]);
$total = $total + $simpanakhirkomp[$i];

```

Gambar 6.3 Kode Program Persentase Nilai Peserta

Setelah persentase didapatkan maka dilakukan pengkategorian dengan kondisi:

```

function viewpersenkomp($persen) {
    $hasil = 0;
    if($persen >= 0 && $persen <= 25){
        $hasil = 1;
    }else if($persen > 25 && $persen <= 50){
        $hasil = 2;
    }else if($persen > 50 && $persen <= 75){
        $hasil = 3;
    }else if($persen > 75 && $persen <= 100){
        $hasil = 4;
    }
    return $hasil;
}

```

Gambar 6.4 Kode Program Pengkategorian Nilai Peserta

Sehingga didapatkan nilai untuk 14 kompetensi tersebut dengan rentang nilai antara 1 – 4. Nilai inilah yang akan menjadi nilai dari 14 kompetensi

yang dimiliki setiap alternatif. Data alternatif disimpan pada tabel “nilai_peserta”. Berikut adalah kode program untuk melakukan penyimpanan nilai tersebut

```
addnilai($detail_peserta, 'C01', $simpanakhirkomp[0], $detail_periode);
addnilai($detail_peserta, 'C02', $simpanakhirkomp[1], $detail_periode);
addnilai($detail_peserta, 'C03', $simpanakhirkomp[2], $detail_periode);
addnilai($detail_peserta, 'C04', $simpanakhirkomp[3], $detail_periode);
addnilai($detail_peserta, 'C05', $simpanakhirkomp[4], $detail_periode);
addnilai($detail_peserta, 'C06', $simpanakhirkomp[5], $detail_periode);
addnilai($detail_peserta, 'C07', $simpanakhirkomp[6], $detail_periode);
addnilai($detail_peserta, 'C08', $simpanakhirkomp[7], $detail_periode);
addnilai($detail_peserta, 'C09', $simpanakhirkomp[8], $detail_periode);
addnilai($detail_peserta, 'C10', $simpanakhirkomp[9], $detail_periode);
addnilai($detail_peserta, 'C11', $simpanakhirkomp[10], $detail_periode);
addnilai($detail_peserta, 'C12', $simpanakhirkomp[11], $detail_periode);
addnilai($detail_peserta, 'C13', $simpanakhirkomp[12], $detail_periode);
addnilai($detail_peserta, 'C14', $simpanakhirkomp[13], $detail_periode);
```

Gambar 6.5 Kode Program Penyimpan Nilai Peserta

Kode program function “addnilai” yang berisi query untuk menyimpan kedatabase

```
function addnilai($aidpeserta, $aidbobot, $anilai, $aperiode) {
    include('config.php');
    $query = "INSERT INTO nilai_peserta (id_peserta, id_bobot, nilai, id_periode) "
        . "VALUES('$aidpeserta', '$aidbobot', '$anilai', '$aperiode)";
    mysqli_query($db, $query);
    return mysqli_affected_rows($db);
}
```

Gambar 6.6 Kode Program Function Query Nilai Peserta

2. Memasukkan bobot kriteria

Dalam menentukan bobot pada setiap kriteria tidak ada aturan baku untuk menentukan kriteria mana yang menempati bobot tertinggi atau bobot terendah. Pada penelitian ini penulis mengasumsikan memberikan pembobotan pada kriteria berdasarkan acuan atau asumsi pribadi sebagai seorang DM (Decision Maker) / (PK) Pengambil Keputusan dengan

melihat indikator pada masing-masing kriteria bahwa semakin tinggi nilai kompetensi yang diberikan oleh DM maka semakin tinggi bobot kepentingannya, sehingga didapatkan hasil sebagai berikut;

Nilai/Bobot	Tingkat Kepentingan	Kriteria/Kompetensi
1	Sangat Rendah	-
2	Rendah	C08, C09, C11, C13, C14
3	Cukup	C03, C04, C10
4	Tinggi	C02, C07, C12
5	Sangat Tinggi	C01, C05, C06

Dikarenakannya tidak adanya aturan baku pada penentuan bobot kriteria maka penulis memberikan form isian untuk menentukan bobot masing-masing kriteria berdasarkan pada asumsi dari DM (Decision Maker) / PK (Pengambil Keputusan).

Berikut adalah kode program untuk mengatur bobot dari kriteria.

```

$book = ambileditbobot($_GET['kode']);
$edit = mysqli_fetch_assoc($book);
$dafkom = viewdafkomp();
?>
<div class="row">
  <div class="col-auto mr-auto"><h4>Silahkan isi untuk mengubah kriteria dengan ID :<?php echo $edit['bobot_id'];?></h4></div>
  <div class="col-auto"><a href="bobot_view.php" class="btn btn-info">Lihat Data</i></a></div>
</div>
<form method="post">
<div class="form-group">
  <label>ID Kriteria</label>
  <input type="text" class="form-control" name="id" value="<?php echo $edit['bobot_id'];?>" disabled/>
  <input type="hidden" name="id" value="<?php echo $edit['bobot_id'];?>" />
</div>
<div class="form-group">
  <label>Kriteria</label>
  <input type="text" class="form-control" name="nama" value="<?php echo $dafkom[$_GET['komp']];?>" disabled/>
</div>
<div class="form-group">
  <label>Bobot</label>
  <select class="form-control" name="nilai">
    <option value="1" <?php if($edit['bobot_nilai'] == '1'){ echo 'selected'; }?>>Sangat Rendah</option>
    <option value="2" <?php if($edit['bobot_nilai'] == '2'){ echo 'selected'; }?>>Rendah</option>
    <option value="3" <?php if($edit['bobot_nilai'] == '3'){ echo 'selected'; }?>>Cukup</option>
    <option value="4" <?php if($edit['bobot_nilai'] == '4'){ echo 'selected'; }?>>Tinggi</option>
    <option value="5" <?php if($edit['bobot_nilai'] == '5'){ echo 'selected'; }?>>Sangat Tinggi</option>
  </select>
</div>
<button type="submit" class="btn btn-primary" name="kirim">
  Ubah
</button>
</form>

```

Gambar 6.7 Kode Program Form Bobot Kriteria

Berikut ini adalah tampilan dari form untuk mengatur bobot kriteria

Gambar 6.8 Tampilan Form Bobot Kriteria

Form ini akan mengirim data melalui kode program dibawah ini:

```

if (isset($_POST['kirim'])) {
    $id = $_POST['id'];
    $nilai = $_POST['nilai'];
    $nilai2 = 0;
    if($nilai == 1){
        $nilai2 = 1;
    }else if($nilai == 2){
        $nilai2 = 3;
    }else if($nilai == 3){
        $nilai2 = 5;
    }else if($nilai == 4){
        $nilai2 = 7;
    }else if($nilai == 5){
        $nilai2 = 9;
    }
    $success = updatebobot($id, $nilai, $nilai2);
    if ($success > 0) {
        header('Location: bobot_view.php');
    } else {
        echo' <div class="alert alert-danger" role="alert">
            <strong>Maaf</strong>, data gagal diinputkan, silahkan coba inputkan kembali.
        </div>';
    }
}

```

Gambar 6.9 Kode Program Bobot Kriteria

Nilai pembobotan dari kriteria tersebut akan disimpan ke database dengan memanggil function “updatebobot” melalui kode program dibawah ini

“`$success = updatebobot($id, $nilai, $nilai2);`”

Gambar 6.10 Kode Program Memanggil Function Ganti Bobot Kriteria

Dibawah ini adalah isi function “updatebobot” yang berfungsi untuk menyimpan data ke database yaitu tabel “bobot”

```
function updatebobot($aid, $anilai, $anilai2) {
    include('config.php');
    $query = "UPDATE bobot "
        . "SET bobot_nilai='$anilai', bobot_ahp='$anilai2' "
        . "WHERE bobot_id='$aid'";
    mysqli_query($db, $query);
    return mysqli_affected_rows($db);
}
```

Gambar 6.11 Kode Program Function Query Ganti Bobot Kriteria

3. Melakukan normaliasi alternatif

Pertama dilakukan pengambilan data yang dibutuhkan dari database dengan kode program seperti pada Gambar 6.12

```
$tampil = viewpenilaian();
```

Gambar 6.12 Kode Program Memanggil Penilaian TOPSIS

Kode diatas akan menjalankan kode program Gambar 6.13 yang berisi query database.

```
function viewpenilaian(){
    include('config.php');
    $query = "SELECT b.peserta_id, b.peserta_nama, c.bobot_id, a.nilai, c.bobot_nilai
        FROM nilai_peserta a
        LEFT JOIN peserta b ON a.id_peserta = b.peserta_id
        LEFT JOIN bobot c ON a.id_bobot = c.bobot_id";
    $result = mysqli_query($db, $query);
    return $result;
}
```

Gambar 6.14 Kode Program Memanggil Query database

Data hasil query tersebut kemudian diolah untuk memisahkan data kriteria dan data bobot dengan kode program pada Gambar 6.15


```

$data      =array();
$kriterias =array();
$bobot     =array();
$nilai_kuadrat =array();
$setnama   =array();
if ($tampil) {
    while($row=$tampil->fetch_object()){
        if(!isset($data[$row->peserta_id])){
            $data[$row->peserta_id]=array();
        }
        if(!isset($data[$row->peserta_id][$row->bobot_id])){
            $data[$row->peserta_id][$row->bobot_id]=array();
        }
        if(!isset($nilai_kuadrat[$row->bobot_id])){
            $nilai_kuadrat[$row->bobot_id]=0;
        }
        $bobot[$row->bobot_id]=$row->bobot_nilai;
        $data[$row->peserta_id][$row->bobot_id]=$row->nilai;
        $nilai_kuadrat[$row->bobot_id]+=pow($row->nilai,2);
        $kriterias[]=$row->bobot_id;
        $setnama[$row->peserta_id]=$row->peserta_nama;
    }
}
$kriteria   =array_unique($kriterias);
$jml_kriteria =count($kriteria);

```

Gambar 6.15 Kode Program Pengolahan Penilaian TOPSIS

Data-data yang dihasilkan dari pengolahan diatas adalah:

- \$data digunakan untuk menyimpan data-data alternatif
- \$bobot digunakan untuk menyimpan data bobot kriteria
- \$kriteria digunakan untuk menyimpan data nama kriteria (berupa id)
- \$setnama digunakan untuk menyimpan nama alternatif
- \$nilai_kuadrat digunakan untuk menyimpan hasil penjumlahan dari data nilai alternatif yang telah dipangkat 2. Nilai_kuadrat akan digunakan untuk menormalisasi alternatif.

Setelah semua data didapatkan, abrulah dapat dilakukan perhitungan normalisasi kriteria.

Normalisasi dilakukan dengan cara membagi nilai alternatif dengan nilai rata-rata hasil penjumlahan dari data nilai alternatif yang telah dipangkat

2.

```

$i = 0;
$y = array();
foreach ($data as $nama => $krit) {
    (++)$i;
    echo "<tr>
    <td>{$nama}</td>";
    foreach ($kriteria as $k) {
        $y[$k][$i - 1] = round(($krit[$k] / sqrt($nilai_kuadrat[$k])), 4) * $bobot[$k];
        echo "<td align='center'>" . $y[$k][$i - 1] . "</td>";
    }
    echo
    "</tr>\n";
}
?>

```

Gambar 6.16 Kode Program Normalisasi Alternatif TOPSIS

4. Mencari solusi ideal positif

Semua data kriteria adalah benefit sehingga solusi ideal positif adalah nilai max dari tiap kriteria hasil normalisasi alternatif TOPSIS. Kode Programnya seperti Gambar 6.17.

```

$yplus = array();
foreach ($kriteria as $k) {
    $yplus[$k] = ([ $k ] ? max($y[$k]) : min($y[$k]));
    echo "<td>{$yplus[$k]}</td>";
}

```

Gambar 6.17 Kode Program Solusi Ideal Positif TOPSIS

5. Mencari solusi ideal negative

Semua data kriteria adalah benefit sehingga solusi ideal negative adalah nilai min dari tiap kriteria hasil normalisasi alternatif TOPSIS. Kode Programnya seperti Gambar 6.18.

```

$ymin = array();
foreach ($kriteria as $k) {
    $ymin[$k] = [$k] ? min($y[$k]) : max($y[$k]);
    echo "<td>{$ymin[$k]}</td>";
}

```

Gambar 6.18 Kode Program Solusi Ideal Negative TOPSIS

Mencari jarak positif Nilai jarak positif didapatkan dari hasil perhitungan yaitu hasil akar hasil pengurangan nilai max kriteria dikurangi nilai kriteria normalisasi dipangkat dua. Kode programnya seperti Gambar 6.19.

```

$i = 0;
$dplus = array();
foreach ($data as $nama => $krit) {
    ++$i;
    echo "<tr>
    <td>{$nama}</td>";
    foreach ($kriteria as $k) {
        if (!isset($dplus[$i - 1]))
            $dplus[$i - 1] = 0;
        $dplus[$i - 1] += pow($yplus[$k] - $y[$k][$i - 1], 2);
    }
    echo "<td>" . round(sqrt($dplus[$i - 1]), 6) . "</td>
    </tr>\n";
}

```

Gambar 6.19 Kode Program Jarak Positif TOPSIS

6. Mencari jarak negative

Nilai jarak negative didapatkan dari hasil perhitungan yaitu hasil akar hasil pengurangan nilai kriteria normalisasi dikurangi nilai min kriteria dipangkat dua. Kode programnya seperti Gambar 6.20

```

$i = 0;
$dmin = array();
foreach ($data as $nama => $krit) {
    (++$i);
    echo "<tr>
    <td>{$nama}</td>";
        foreach ($kriteria as $k) {
            if (!isset($dmin[$i - 1]))
                $dmin[$i - 1] = 0;
            $dmin[$i - 1] += pow($ymin[$k] - $y[$k][$i - 1], 2);
        }
        echo "<td>" . round(sqrt($dmin[$i - 1]), 6) . "</td>
    </tr>\n";
}

```

Gambar 6.20 Kode Program Jarak Negative TOPSIS

7. Menghitung nilai preferensi (V)

Dilakukanlah nilai preferensi (V) atau nilai TOPSIS yang dilakukan dengan kode program Gambar 6.20 dibawah ini

```

$i = 0;
$V = array();
$hasiltopsis = array();
foreach ($data as $nama => $krit) {
    (++$i);
    echo "<tr>
    <td>{$nama}</td>";
        foreach ($kriteria as $k) {
            $V[$i - 1] = round(sqrt($dmin[$i - 1]) / (sqrt($dmin[$i - 1]) + sqrt($dplus[$i - 1])), 4);
        }
        echo "<td>{$V[$i - 1]}</td></tr>\n";
        $hasiltopsis[$nama]=$V[$i - 1];
}

```

Gambar 6.20 Kode Program Nilai Preferensi (V) Nilai TOPSIS

8. Melakukan Perangkingan

```

$i = 0;
arsort($hasiltopsis);
foreach($hasiltopsis as $x=>$x_value) {
    echo "<tr>
        <td>" . (++$i) . "</td>
        <td>{$x}</td>
        <td>${setnama[$x]}</td>
        <td>{$x_value}</td>";
}

```

Gambar 6.21 Kode Program Perangkingan

9. Menghitung *hamming distance*

Untuk menghitung *hamming distance* yang digunakan untuk mengetahui tingkat ketidaksesuian perangkingan dengan hasil perhitungan manual, maka langkah yang harus dilakukan adalah:

1. Merangking hasil perhitungan manual dan mencari guru yang memiliki nilai yang sama untuk ditetapkan sebagai rangking bersama.

Perangkingan

No	ID Peserta	Nama	Nilai
1	PS0021 (1)	Moch. Yusroni, S.Pd.	37
2	PS0014 (1)	Diah Anita Ludfiana, S.Pd.	37
3	PS0017 (1)	Yeni Yusella, S.Pd.	37
4	PS0022 (4)	Nihayatul Hasanah, S.Pd	36

Gambar 6.22 Contoh Perangkingan Bersama

Gambar diatas adalah contoh hasil perangkingan manual. Dalam hasil perangkingan atas terdapat rangking bersama yaitu no. 1, 2 dan 3 atau Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd. merupakan rangking bersama karena memiliki nilai yang sama yaitu 37. Moch. Yusroni, S.Pd, Diah Anita Ludiana, S.Pd dan Yeni Yusella, S.Pd jika dirangkingkan, maka mereka ada dirangking 1, 2 dan 3. Sedangkan Nihayatul Hasanah, S.Pd ada pada rangking 4.

Kode program untuk mengambil hasil perangkingan perhitungan manual seperti yang terdapat pada gambar 6.23 Pada kode program tersebut hasil perhitungan manual akan disimpan pada variabel `stmt4`.

```
$stmt4 = viewnilaidetailperangkingan();
```

Gambar 6.23 Kode Program Untuk Mengambil Hasil Perhitungan Manual

Setelah dilakukan pengambilan perangkingan manual, pada data perangkingan manual tersebut akan dicari guru yang memiliki nilai yang sama untuk untuk ditetapkan sebagai rangking bersama. Kode program dapat dilihat pada gambar 6.24

```

$i = 0;
$ranking = 1 ;
$perban = -1;
while ($data2 = mysqli_fetch_assoc($stmt4)) {
    if($perban == -1){
        $perban = $data2['total'];
    }
    if($perban != $data2['total']){
        $ranking = $i+1;
        $perban = $data2['total'];
    }
    $simpan_hsl_manual[$i] = $data2['detail_peserta'];
    $simpan_n_manual[$data2['detail_peserta']] = $ranking;
    $i++;
};

```

Gambar 6.24 Kode Program Untuk Mencari Rangking Bersama

2. Mengambil hasil perangkingan dengan metode TOPSIS

Hasil perangkingan ini yang akan dibandingkan dengan hasil prangkingan manual. Kode program untuk Mengambil hasil perangkingan dengan metode TOPSIS adalah :

```

$stmt3 = viewsimpanhasil("topsis");
$i = 0;
while ($data2 = mysqli_fetch_assoc($stmt3)) {
    $simpan_hsl_topsis[$i] = $data2['id_peserta'];
    $i++;
};

```

Gambar 6.25 Kode Program Mengambil hasil perangkingan dengan metode TOPSIS

3. Mencari nilai *hamming distance*

Kode Program untuk mencari nilai *hamming distance* seperti yang ditunjukkan pada Gambar 6.26 dibawah ini :

```

function perbandingan($nilaimetode, $nilaimanual, $arraynilaimanual) {
    $tdksama = 0;
    $temptdksama = true;
    if($nilaimetode==$nilaimanual){
        $temptdksama = false;
        echo "<td><b>[". $nilaimetode. "]/</b></td>";
    }else{
        echo "<td>";
        foreach($arraynilaimanual as $x=>$x_value) {
            //echo $x. " ". $x_value. "</br>";
            if($arraynilaimanual[$nilaimanual] == $x_value){
                if($arraynilaimanual[$nilaimetode] == $x_value){
                    echo "<div class='text-xs'>";
                    if($nilaimetode==$x){
                        $temptdksama = false;
                        echo "<b>[". $nilaimetode. " ". $x. " ". $x_value. " ] </b> | ";
                    }else{
                        echo $nilaimetode. " ". $x. " ". $x_value. " | ";
                    }
                    echo "</div>";
                }
            }
        }

        if($temptdksama == true){
            $tdksama++;
            echo $nilaimetode;
        }else{
            echo "<b>[". $nilaimetode. "]/</b>";
        }
        echo "</td>";
    }
    return $tdksama;
}

```

Gambar 6.26 Kode Program *hamming distance*

Untuk mencari nilai *hamming distance* langkah-langkah yang dilakukan adalah :

- 1) Mengambil data guru yang berada pada rangking yang dicari pada hasil perangkingan manual.
- 2) Membandingkan dengan hasil pengakingan manual dengan hasil perangkingan metode TOPSIS pada rangking yang dicari. Dengan ketentuan :
 - Jika sama, lanjut ke langkah 3 dan tidak dihitung sebagai ketidaksesuaian

- Jika tidak sama, maka dicek apakah pada hasil perangkingan manual pada rangking yang dicari merupakan rangking bersama.
 - Jika bukan merupakan rangking bersama, maka lanjut kelangkah 3 dan dihitung sebagai ketidaksesuaian
 - Jika merupakan rangking bersama, maka dicari pada rangking bersama apakah ada guru hasil rangking metode TOPSIS yang sesuai.
 - Jika ada, maka tidak dihitung sebagai ketidaksesuaian dan lanjut ke langkah 3
 - Jika tidak ada, maka dihitung sebagai ketidaksesuaian dan lanjut ke langkah 3

3) Mencari untuk rangking selanjutnya

4. Selesai

6.3 Uji Coba Metode Technique For Others Preference by Similarity to Ideal Solution (TOPSIS)

Pengujian dilakukan banyak 6 kali pengujian. Pengujian tersebut dilakukan dengan perbedaan nilai bobot kriteria yang berbeda. Adapun pengujian dengan perbedaan nilai bobot kriteria seperti ditampilkan pada tabel 6.1 berikut:

Tabel 6.1 Pengujian TOPSIS

Pengujian	Kriteria													
	C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11	C12	C13	C14
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	5	4	4	4	4	3	3	3	3	2	2	2	2	2
5	2	2	2	2	2	3	3	3	3	4	4	4	4	5
6	3	3	2	4	4	3	5	2	4	3	3	2	2	2

Keterangan tabel adalah

C01 = Mengetahui karakteristik peserta didik,

C02 = Mengetahui teori belajar dan prinsip-prinsip pembelajaran yang mendidik,

C03 = Pengembangan kurikulum,

C04 = Kegiatan pembelajaran yang mendidik,

C05 = Mengembangkan potensi peserta didik,

C06 = Komunikasi dengan peserta didik,

C07 = Penilaian dan evaluasi,

C08 = Bertindak sesuai dengan norma agama, hukum, sosial dan kebudayaan nasional,

C09 = Menunjukkan pribadi yang dewasa dan teladan,

C10 = Etos kerja, tanggung jawab yang tinggi, rasa bangga menjadi guru,

C11 = Bersikap inklusif, bertindak obyektif, serta tidak diskriminatif,

C12 = Komunikasi dengan sesama guru, tenaga kependidikan, orang tua, peserta didik, dan masyarakat,

C13 = Penguasaan materi, struktur, konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu, dan

C14 = dan mengembangkan keprofesionalan melalui tindakan yang reflektif.

Hasil perangkingan dari pengujian kemudian dihitung kevalidasinya berdasarkan nilai ketidaksesuaian perangkingan dengan hasil perhitungan manual dengan algoritma *hamming distance*. Nilai algoritma *hamming distance* yang semakin besar menandakan semakin besar berbeda dengan hasil perhitungan manual atau semakin banyak yang perangkingannya tidak sesuai.

Proses pengujian adalah sebagai berikut:

1. Pengujian 1

Pada pengujian 1 ini menggunakan bobot kriteria terendah untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 6.2 dibawah ini:

Tabel 6.2 Bobot Kriteria Pengujian 1 TOPSIS

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian 1 TOPSIS

Uji Coba	Hasil Nilai Hamming Distance untuk Metode TOPSIS
1	8
2	6
3	2
4	4
5	4
Rata-rata	4,8

Hasil pengujian 1 untuk metode TOPSIS ini menghasilkan nilai ketidaksesuaian terendah adalah 2 kesalahan, nilai ketidaksesuaian tertinggi adalah 8 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 4.8 kesalahan.

2. Pengujian 2

Pada pengujian 2 ini menggunakan bobot kriteria tertinggi untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 6.4.

Tabel 6.4 Bobot Kriteria Pengujian 2 TOPSIS

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 6.5.

Tabel 6.5 Hasil Pengujian 2 TOPSIS

Uji Coba	Hasil Nilai Hamming Distance untuk Metode TOPSIS
1	8
2	6
3	10
4	6
5	7
Rata-rata	7,4

Hasil pengujian 2 untuk metode TOPSIS ini menghasilkan nilai ketidaksesuaian terendah adalah 6 kesalahan, nilai ketidaksesuaian tertinggi adalah 10 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 7,4 kesalahan.

3. Pengujian 3

Pada pengujian 3 ini menggunakan bobot kriteria nilai 5 (pertengahan) untuk semua kriteria. Bobot-bobot kriteria tersebut seperti yang ditunjukkan pada Tabel 6.6.

Tabel 6.6 Bobot Kriteria Pengujian 3 TOPSIS

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 6.7.

Tabel 6.7 Hasil Pengujian 3 TOPSIS

Uji Coba	Hasil Nilai Hamming Distance untuk Metode TOPSIS
1	2
2	4
3	4
4	6
5	2
Rata-rata	3,6

Hasil pengujian 3 untuk metode TOPSIS ini menghasilkan nilai ketidaksesuaian terendah adalah 2 kesalahan, nilai ketidaksesuaian tertinggi adalah 6 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 3,6 kesalahan.

4. Pengujian 4

Pada pengujian 4 ini menggunakan bobot kriteria dari tinggi ke rendah. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 6.8.

Tabel 6.8 Bobot Kriteria Pengujian 4 TOPSIS

	C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11	C12	C13	C14
Bobot	5	4	4	4	4	3	3	3	3	2	2	2	2	2

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 6.9.

Tabel 6.9 Hasil Pengujian 4 TOPSIS

Uji Coba	Hasil Nilai Hamming Distance untuk Metode TOPSIS
1	15
2	11
3	14
4	14
5	17
Rata-rata	14,2

Hasil pengujian 4 untuk metode TOPSIS ini menghasilkan nilai ketidaksesuaian terendah adalah 11 kesalahan, nilai ketidaksesuaian tertinggi adalah 17 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 14,2 kesalahan.

5. Pengujian 5

Pada pengujian 5 ini menggunakan bobot kriteria dari nilai rendah ke tinggi Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 6.10.

Tabel 6.10 Bobot Kriteria Pengujian 5 TOPSIS

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	2	2	2	2	2	3	3	3	3	4	4	4	4	5

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 6.11.

Tabel 6.11 Hasil Pengujian 5 TOPSIS

Uji Coba	Hasil Nilai Hamming Distance untuk Metode TOPSIS
1	13
2	13
3	14
4	18
5	16
Rata-rata	14,8

Hasil pengujian 5 untuk metode TOPSIS ini menghasilkan nilai ketidaksesuaian terendah adalah 13 kesalahan, nilai ketidaksesuaian tertinggi adalah 18 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 14,8 kesalahan.

6. Pengujian 6

Pada pengujian 6 ini menggunakan bobot kriteria yang acak. Bobot-bobot kriteria tersebut seperti yang ditunjukkan seperti pada Tabel 6.12.

Tabel 6.12 Bobot Kriteria Pengujian 6 TOPSIS

	C 01	C 02	C 03	C 04	C 05	C 06	C 07	C 08	C 09	C 10	C 11	C 12	C 13	C 14
Bobot	3	3	2	4	4	3	5	2	4	3	3	2	2	2

Pengujian dilakukan dengan jumlah alternatif / peserta / guru sebanyak 25 peserta.

Hasil Pengujian dengan nilai guru yang berbeda (Berdasarkan nilai *hamming distance* / nilai ketidaksesuaian / nilai error) seperti yang ditunjukkan pada Tabel 6.13.

Tabel 6.13 Hasil Pengujian 6 TOPSIS

Uji Coba	Hasil Nilai Hamming Distance untuk Metode TOPSIS
1	14
2	13
3	13
4	17
5	15
Rata-rata	14,4

Hasil pengujian 6 untuk metode TOPSIS ini menghasilkan nilai ketidaksesuaian terendah adalah 13 kesalahan, nilai ketidaksesuaian tertinggi adalah 17 kesalahan dan rata-rata nilai ketidaksesuaian dari 5 kali uji coba adalah 14.4 kesalahan.

BAB VII

PEMBAHASAN

Di bab ini akan berisi hasil dari perbandingan ketidaksesuaian metode AHP, WSM, WPM dan TOPSIS melalui data-data pengujian yang telah dilakukan pada bab-bab sebelumnya. Data ketidaksesuaian tersebut didapatkan dari hasil *hamming distance* antara perangkingan dari metode-metode dengan perangkingan hasil perhitungan manual. Nilai algoritma *hamming distance* yang semakin besar menandakan semakin besar berbeda dengan hasil perhitungan manual atau semakin banyak yang perangkingannya tidak sesuai. Pembahasan perbandingan dalam bab ini akan dibagi menjadi 3 bagian yaitu pertama: perbandingan antar metode untuk mencari tingkat ketidaksesuaian dalam semua jenis pembobotan, kedua: perbandingan antar metode ketika bobot kriteria sama dan ketiga: perbandingan antar metode untuk mencari tingkat ketidaksesuaian ketika bobot tidak sama antar kriteria.

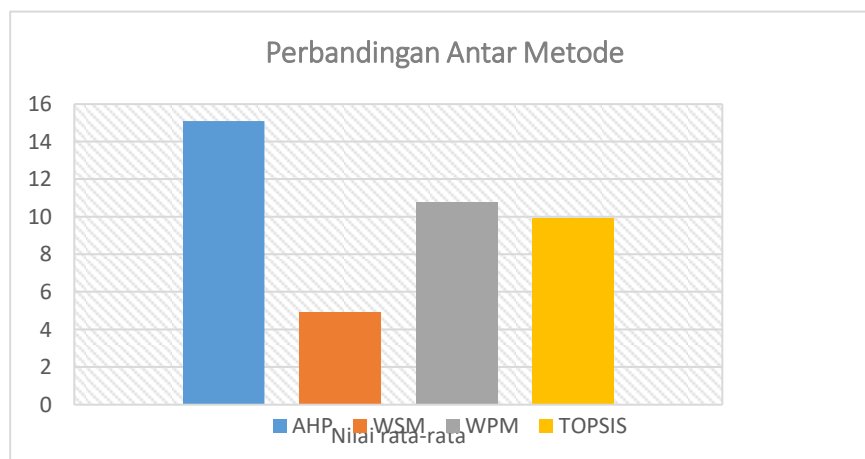
7.1 Perbandingan metode AHP, WSM, WPM, dan TOPSIS

Tabel 7.1 menunjukkan hasil perbandingan metode AHP, WSM, WPM, dan TOPSIS untuk mencari tingkat ketidaksesuaian perangkingan dengan perhitungan manual.

Tabel 7.1 Perbandingan Metode AHP, WSM, WPM dan Topsis

Pembobotan Kriteria	AHP	WSM	WPM	TOPSIS
Semua bobot sangat rendah	14,6	0,4	9,2	4,8
Semua bobot sangat tinggi	10	0,8	7,8	7,4
Semua bobot cukup	14,8	1,2	8,6	3,6
Bobot tinggi kerendah	16,4	8	12,8	14,2
Bobot rendah ketinggi	18,2	8,6	13	14,8
Bobot diacak	16,8	10,2	13,4	14,4
Rata-rata	15,1	4,9	10,8	9,9

Berdasarkan Tabel 7.1 dapat disimpulkan bahwa WSM memiliki tingkat ketidaksesuaian terkecil dengan rata-rata 4,9 kesalahan perangkingan sedangkan untuk tingkat ketidaksesuaian tertinggi adalah metode AHP dengan 15,1 tingkat kesalahan. Urutan kedua untuk ketidaksesuaian terkecil adalah TOPSIS dengan rata-rata 9,9 kesalahan dan ketiga WPM dengan rata-rata 10,8 kesalahan. Data diatas dihitung dari 25 alternatif guru. Gambar 7.1 menunjukkan grafik perbandingan metode AHP, WSM, WPM, dan TOPSIS.



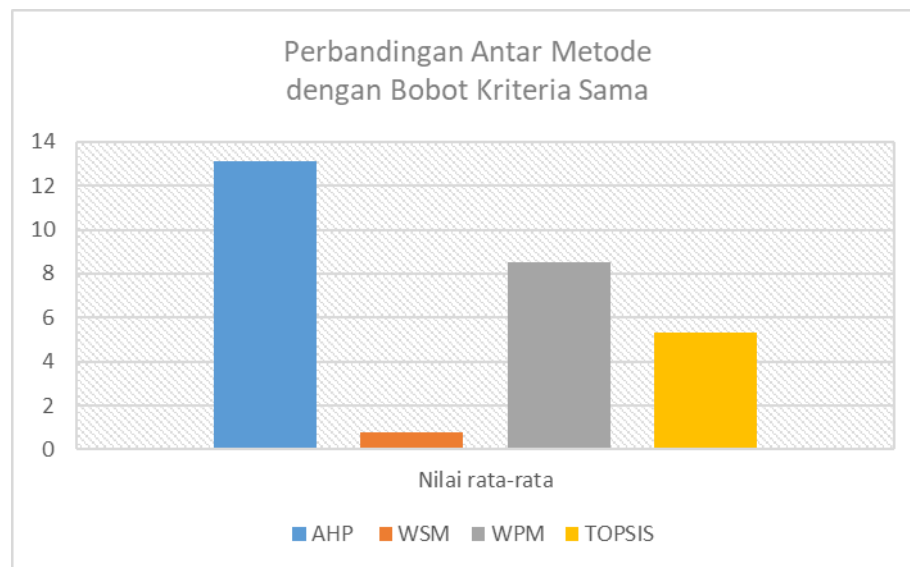
Gambar 7.1 Perbandingan Metode AHP, WSM, WPM, dan TOPSIS

7.2 Perbandingan antar metode (dengan nilai kriteria yang sama)

Hasil perbandingan metode AHP, WSM, WPM dan Topsis dengan bobot kriteria sama hasilnya seperti yang ditunjukkan Tabel 7.2.

Tabel 7.2 Perbandingan Metode AHP, WSM, WPM dan Topsis dengan Bobot Kriteria Sama

Pembobotan Kriteria	AHP	WSM	WPM	TOPSIS
Semua bobot sangat rendah	14,6	0,4	9,2	4,8
Semua bobot sangat tinggi	10	0,8	7,8	7,4
Semua bobot cukup	14,8	1,2	8,6	3,6
Rata-rata	13,1	0,8	8,5	5,3



Gambar 7.2 Perbandingan Metode AHP, WSM, WPM, dan TOPSIS dengan Bobot Kriteria Sama

Berdasarkan Tabel 7.2 dan Gambar 7.2 menunjukkan bahwa urutan hasil rata-rata ketidaksesuaian dari yang terkecil yaitu WSM = 0,8, TOPSIS = 5,3, WPM = 8,5 dan AHP = 13,1.

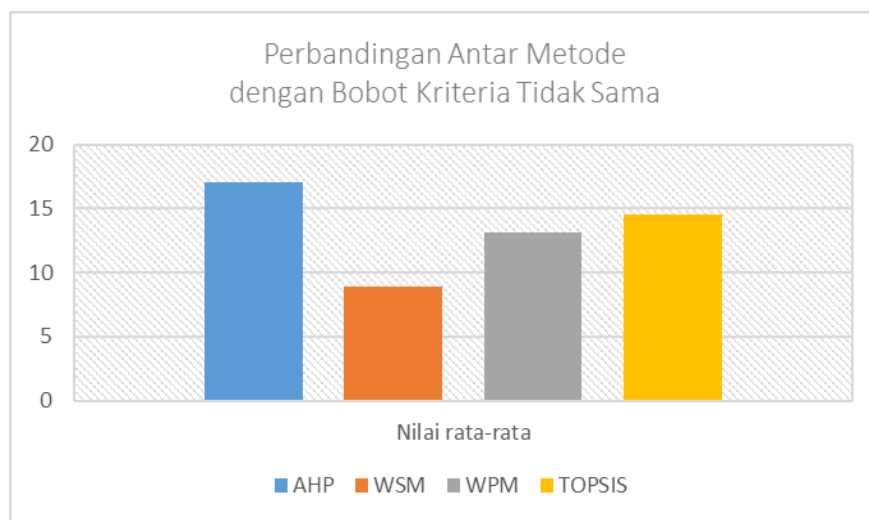
7.3 Perbandingan antar metode (dengan bobot kriteria yang tidak sama)

Hasil perbandingan metode AHP, WSM, WPM dan Topsis dengan bobot kriteria tidak sama hasilnya seperti yang ditunjukkan Tabel 7.3.

Tabel 7.3 Perbandingan Metode AHP, WSM, WPM dan Topsis dengan Bobot Kriteria

Tidak sama

Pembobotan Kriteria	AHP	WSM	WPM	TOPSIS
Bobot tinggi kerendah	16,4	8	12,8	14,2
Bobot rendah ketinggian	18,2	8,6	13	14,8
Bobot diacak	16,8	10,2	13,4	14,4
Rata-rata	17,1	8,9	13,1	14,5



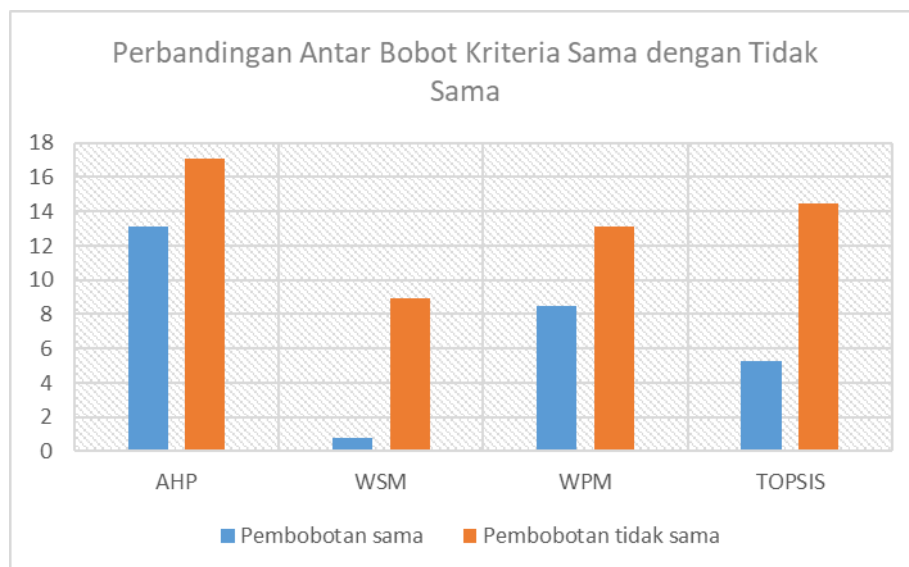
Gambar 7.3 Perbandingan Metode AHP, WSM, WPM, dan TOPSIS dengan Bobot Kriteria Sama

Urutan hasil perbandingan dengan bobot kriteria yang tidak sama berbeda dengan hasil sebelumnya. Berdasarkan Tabel 7.2, WSM masih memiliki tingkat ketidaksesuaian terkecil dengan rata-rata 8,9 kesalahan perangkingan sedangkan untuk tingkat ketidaksesuaian tertinggi adalah masih metode AHP dengan 17,1 tingkat kesalahan. Urutan kedua untuk ketidaksesuaian terkecil pada perbandingan ini adalah WPM dengan rata-rata 13,1 kesalahan dan ketiga TOPSIS dengan rata-rata 13,1 kesalahan.

Dari data-data perbandingan diatas metode yang memiliki tingkat ketidaksesuaian terkecil adalah WSM. Bahkan dibeberapa pengujian metode WSM ini memiliki tingkat ketidaksesuaian adalah 0 kesalahan, dimana metode-metode lainnya tidak bisa menghasilkan ketidaksesuaian seperti metode WSM ini.

Metode AHP merupakan metode yang memiliki tingkat ketidaksesuaian tertinggi. Berdasarkan pengujian-pengujian dibab sebelumnya tingkat ketidaksesuaian terbesar adalah 21 kesalahan perangkingan yang artinya hanya memilki 4 perangkingan yang sesuai.

Untuk metode TOPSIS dan WPM, keduanya berada di urutan ketidaksesuaian 3 dan 4.



Gambar 7.4 Perbandingan Bobot Kriteria Sama dan Tidak Sama

Dari perbandingan hasil pengujian diatas menunjukkan bahwa pengujian pada pembobotan yang sama menghasilkan tingkat ketidaksesuain lebih kecil dari pada pembobotan yang tidak sama.

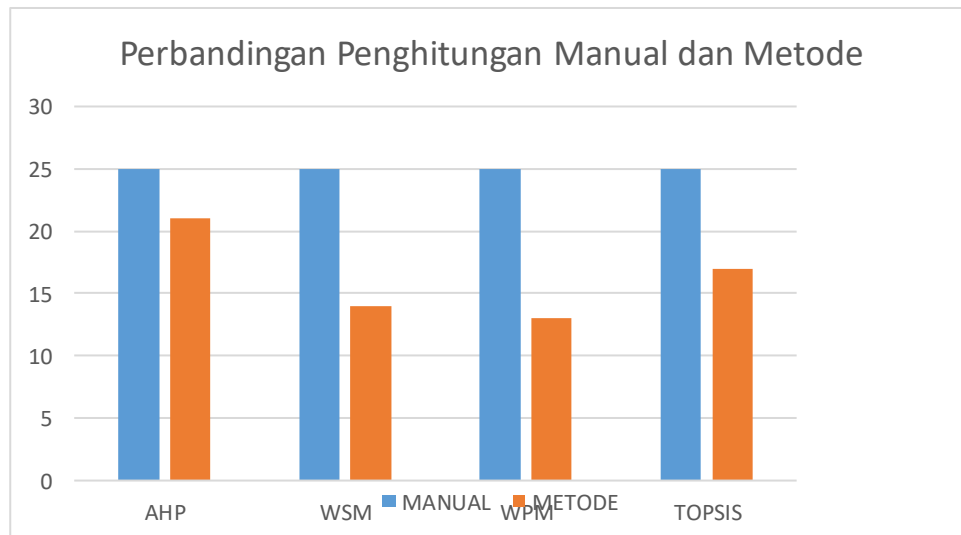
7.4 Perbandingan Hasil Rangkings Manual dengan Metode AHP, WSM, WPM dan TOPSIS

Hasil perbandingan urutan rangking Manual dengan metode AHP, WSM, WPM dan TOPSIS ditunjukkan pada tabel 7.4

Tabel. 7.4 Perbandingan Hasil

Rangking	Manual	AHP	WSM	WPM	TOPSIS
1.	Amiruddin Nahdliyan, S.Hum	Hadi Susanto, S.Hum	Amiruddin Nahdliyan, S.Hum.	Amiruddin Nahdliyan, S.Hum.	Amiruddin Nahdliyan, S.Hum.
2.	Noor Jehhan, S.Ag., M.Pd.I	Mohamad Nurahman, S.Pd.I	Hadi Susanto, S.Hum	Hadi Susanto, S.Hum	Hadi Susanto, S.Hum
3.	Hadi Susanto, S.Hum	Amiruddin Nahdliyan, S.Hum.	Noor Jehhan, S.Ag., M.Pd.I	Noor Jehhan, S.Ag., M.Pd.I	Arif Nur Rahman, S.S.
4.	Moch. Yusroni, S.Pd	Masruhan, S.Pd.I	Moch. Yusroni, S.Pd.	Moch. Yusroni, S.Pd.	Moch. Yusroni, S.Pd.
5.	Paramita Puri Anggraini, S.Pd.	Lailatul Munawaroh, S.Pd.	Arif Nur Rahman, S.S.	Ika Wijayanti, S.AB	Mohamad Nurahman, S.Pd.I
6.	Sri Subakti, S.Pd.SD.	Arif Nur Rahman, S.S.	Sri Subakti, S.Pd.SD.	Sri Subakti, S.Pd.SD.	Muhammad Firmansyah, S.S.
7.	Arif Nur Rahman, S.S.	Sri Subakti, S.Pd.SD.	Paramita Puri Anggraini, S.Pd.	Arif Nur Rahman, S.S.	Sri Subakti, S.Pd.SD.
8.	Ika Wijayanti, S.AB	Muhammad Firmansyah, S.S.	Ika Wijayanti, S.AB	Muhammad Firmansyah, S.S.	Ika Wijayanti, S.AB
9.	Imarotul Muhibbah, S.Si., M.Pd.	Moch. Yusroni, S.Pd.	Muhammad Firmansyah, S.S.	Imarotul Muhibbah, S.Si., M.Pd.	Noor Jehhan, S.Ag., M.Pd.I
10.	Aning Masyrufatin F., S.Pd.I	Noor Jehhan, S.Ag., M.Pd.I	Mohamad Nurahman, S.Pd.I	Aning Masyrufatin F., S.Pd.I	Aning Masyrufatin F., S.Pd.I
11.	Muhammad Firmansyah, S.S.	Ika Wijayanti, S.AB	Imarotul Muhibbah, S.Si.	Paramita Puri Anggraini, S.Pd.	Masruhan, S.Pd.I
12.	Mutaahirotul Hidayati, S.Pd.	Paramita Puri Anggraini, S.Pd.	Aning Masyrufatin F., S.Pd.I	Mohamad Nurahman, S.Pd.I	Lailatul Munawaroh, S.Pd.
13.	Mohamad Nurahman, S.Pd.I	Imarotul Muhibbah, S.Si.	Mutaahirotul Hidayati, S.Pd.	Mutaahirotul Hidayati, S.Pd.	Imarotul Muhibbah, S.Si.
14.	Anis Amilia, S.Pd.I	Anis Amilia, S.Pd.I	Anis Amilia, S.Pd.I	Anis Amilia, S.Pd.I	Paramita Puri Anggraini, S.Pd.
15.	Desty Ariani Mutiara, S.Pd	Mutaahirotul Hidayati, S.Pd.	Desty Ariani Mutiara, S.Pd	Desty Ariani Mutiara, S.Pd	Desty Ariani Mutiara, S.Pd
16.	Lailatul Munawaroh, S.Pd.	Desty Ariani Mutiara, S.Pd	Lailatul Munawaroh, S.Pd.	Lailatul Munawaroh, S.Pd.	Mutaahirotul Hidayati, S.Pd.
17.	Jaronah, S.Pd.I	Herlina Tri Pambudiati, S.Pd.	Masruhan, S.Pd.I	Diah Anita Ludfiana, S.Pd.	Anis Amilia, S.Pd.I
18.	Herlina Tri Pambudiati, S.Pd	Aning Masyrufatin F., S.Pd.I	Herlina Tri Pambudiati, S.Pd.	Jaronah, S.Pd.I	Herlina Tri Pambudiati, S.Pd.
19.	Nihayatul Hasanah, S.Pd	Nihayatul Hasanah, S.Pd	Nihayatul Hasanah, S.Pd	Nihayatul Hasanah, S.Pd	Nihayatul Hasanah, S.Pd
20.	Kusnia, S.Pd.I	Diah Anita Ludfiana, S.Pd.	Jaronah, S.Pd.I	Masruhan, S.Pd.I	Diah Anita Ludfiana, S.Pd.
21.	Diah Anita Ludfiana, S.Pd.	Achmad Efendi, S.Hum	Diah Anita Ludfiana, S.Pd.	Herlina Tri Pambudiati, S.Pd.	Jaronah, S.Pd.I
22.	Nur Rizky Marsa R., S.Pd	Nur Rizky Marsa R., S.Pd.	Nur Rizky Marsa R., S.Pd.	Nur Rizky Marsa R., S.Pd.	Nur Rizky Marsa R., S.Pd.
23.	Masruhan, S.Pd.I	Jaronah, S.Pd.I	Kusnia, S.Pd.I	Achmad Efendi, S.Hum	Achmad Efendi, S.Hum
24.	Gita Noviria, S.Pd.	Gita Noviria, S.Pd.	Achmad Efendi, S.Hum	Gita Noviria, S.Pd.	Kusnia, S.Pd.I
25.	Achmad Efendi, S.Hum	Kusnia, S.Pd.I	Gita Noviria, S.Pd.	Kusnia, S.Pd.I	Gita Noviria, S.Pd.
Tingkat Ketidaksesuaian		21 84%	14 56%	13 52%	17 68%

Berdasarkan tabel 7.4 dapat di simpulkan bahwa tingkat ketidaksesuaian urutan rangking ditunjukkan melalui grafik pada gambar 7.5



Gambar 7.5 Perbandingan penghitungan manuan dan metode

Urutan hasil perangkingan Metode AHP menepati ketidaksesuai tertinggi, sedangkan metode WPM menempati ketidaksesuaian terendah, dengan begitu untuk penentuan metode yang direkomendasikan untuk memilih guru terbaik berdasarkan hasil perbandingan diatas adalah metode AHP yang memiliki ketidaksesuai tertinggi dengan tingkat kesalahan sebesar 21 atau sekitar 84%

BAB VIII

KESIMPULAN

8.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dihasilkan beberapa kesimpulan dari perbandingan hasil ketidaksesuaian metode AHP, WSM, WPM dan TOPSIS melalui data-data pengujian yang telah dilakukan yaitu:

1. Metode dengan tingkat ketidaksesuain terkecil adalah WPM
2. Metode dengan tingkat ketidaksesuian terbesar adalah AHP
3. Pembobotan kriteria yang tidak sama akan menghasilkan tidak ketidaksesuaian yang semakin tinggi.

8.2 Saran

Saran yang dapat dilakukan untuk penelitian selanjutnya adalah:

1. Pada penelitian selanjutnya diharapkan dapat dilakukan perbandingan dengan MCDM yang lainnya.
2. Dapat dilakukan perbandingan MCDM untuk kasus perangkaan yang lain

DAFTAR PUSTAKA

- Alireza Afshari, M. M. (2010). Simple Additive Weighting approach to Personnel Selection problem. *International Journal of Innovation, Management and Technology*, Vol. 1, No. 5.
- Chen Yiming, Z. Y. (2011). SaaS Vendor Selection Basing on Analitical Hierarchy Process. IEEE.
- Fishburn, P. C. 1967. A Problem-based Selection of Multi-Attribute Decision Making Methods. Blackwell Publishing. New Jersey.
- Gungor, Z. (2009). A fuzzy AHP approach to personnel selection problem. *Applied Soft Computing* 9, 641–646.
- Javeed Kittur, P. C. (2015). Evaluating Optimal Generation using different Multi Criteria Decision Making Methods. IEEE.
- Kusumadewi, Sri. 2006. Fuzzy Multi-Attribute Decision Making (Fuzzy MADM). Graha Ilmu. Yogyakarta.
- Lazim Abdullah, C. R. (2014). Simple Additive Weighting Methods of Multi criteria Decision Making and Applications: A Decade Review. *International Journal of Information Processing and Management (IJIPM)*, 39-49.
- Mcleod, Raymond, dan Schell, 2007, Sistem Informasi Manajemen. Ed.9, PT Index, Jakarta.
- Pawito. (2008). Penelitian Komunikasi Kualitatif. Yogyakarta: PT LKiS Pelangi Aksara.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*, Seventh Edition. New York: The McGraw-Hill Companies, Inc.
- Primantari, F. L. (2008). Tesis Aplikasi Analitical Hierarchy Process (AHP) Pada Pemberdayaan Landas Pacu Bandara International Adi Sumarno Surakarta.
- Saaty, T. L. (2008). Decision making with the Analitical Hierarchy Process. *Services Sciences*, Vol. 1, 83-98.
- Schell, R. M. (2007). *Management Information Systems*. New Jersey: Prentice Hall.
- S. H. Andreas Budiman, D. G. (2016). *Implementasi Algoritma Hamming Distance dan Brute Force dalam Mendeteksi Kemiripan Source Code Bahasa Pemrograman C*. *ULTIMATICS*, vol. III, no. 2085–4552, p. 72.

- Sihaloho, S. (2014). Sistem Pendukung Keputusan Penentuan Lokasi Pembangunan Tower Base Tranceiver Station (BTS) Pada Telkomsel dengan Metode ANALITYCAL Hierarchy Process (AHP). *Jurnal Informasi dan Teknologi Ilmu (INTI) Volume III Nomor I*, 84-90.
- Silvi Agustina, A. R. (2010). Pengukuran index konsistensi dalam proses pengambilan keputusan menggunakan metode AHP.
- Silvi Agustina, A. R. (2014). Sistem Pendukung Keputusan Penentuan Prioritas Pelanggan Dealer Suzuki Soekarno-Hatta Malang Menggunakan Metode AHP dan SAW.
- Sommerville, I. (2007). *Software Engeneering Eighth Edition*. England.: Pearson Education Limited.
- Subakti, I. (2002). IF1524 Sistem Pendukung Keputusan (Decision Support). Surabaya: Jurusan Teknik Informatika, Fakultas Teknologi Informasi, ITS.
- Tim Direktorat Profesi Pendidik Dirjen PMPTK, 2010. *Pedoman Pelaksanaan Penilaian Kinerja Guru (PK Guru)*. Kementrian Pendidikan Nasional Direktorat Jenderal Peningkatan Mutu Pendidik dan Tenaga Kependidikan.
- Turban, E. A. (2005). *Decision Support System and Intelligent Systems*. New Delhi: Asoke K. Ghosh, Prentice-Hall of India Private Limited.
- TOPSIS dan AHP-TOPSIS dalam Studi Kasus Sistem Pendukung Keputusan Penerimaan Siswa Program Akselerasi*. *Jurnal ITSMART Vol 2. No. 01. Juni 2013*.
- Valeriy Bezruk, Y. S. (2015). Multi-Criteria Choice of the Preferred Type of Mobile Phone by the ANALITYCAL Hierarchy Process. *Second International Scientific-Practical Conference*, 108-109.