

IMPLEMENTASI SISTEM
PEMILIHAN KARYAWAN
PADA PENJADWALAN
AKADEMIK)

Dalam melakukan pembangunan pada bidang ekonomi di Indonesia, Badan Usaha Milik Negara (BUMN) diharuskan memiliki komitmen dalam mengangkat perekonomian rakyat dalam belenggu kemiskinan, maka terdapat 3 (tiga) bentuk badan hukum perusahaan negara, yaitu: Perusahaan jawatan (Perjan), Perusahaan umum (Perum) dan Perseroan .

Pada setiap perusahaan dituntut dalam kurun waktu pertahun untuk memiliki kenaikan target maupun keseimbangan target dalam hal pasar marketingnya, sehingga perusahaan dapat terus berkembang dan maju.

Dalam hal untuk melakukan pencapaian tersebut, diperlukan sumber daya manusia (SDM) yang mumpuni baik itu pada pimpinan perusahaan, manajer, ataupun staff perusahaan.

Untuk merealisasikan hal tersebut diperlukan nya suatu bidang ilmu tambahan baik itu yang diadakan untuk internal perusahaan maupun hal luas, dalam hal ini perusahaan dapat memberikan suatu fasilitas pengetahuan seperti memberikan fasilitas seminar, training, diklat, sertifikasi, dll.

Dengan adanya kegiatan tersebut maka perusahaan tersebut diharuskan memilih karyawan atau peserta kegiatan yang tepat dan sesuai bidangnya sehingga kesesuaian tersebut dapat menjadi.

Setiap keputusan membutuhkan penyeimbangan banyak faktor, kriteria, yang dilakukan terkadang secara eksplisit,terkadang tanpa pertimbangan sadar. Keputusan mungkin merupakan pilihan sederhana antara dua atau lebih alternatif yang terdefinisi dengan baik namun, seringkali,

masalah keputusan adalah masalah yang agak rumit yang mencakup informasi yang sifatnya kompleks dan saling bertentangan yang mencerminkan berbagai perspektif. Salah satu pendekatan untuk

menyelesaikan masalah tersebut menggunakan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS).

Dengan demikian penelitian ini akan membangun sebuah sistem pemilihan karyawan dalam penjadwalan kegiatan akademik dengan menggunakan metode topsis yang merupakan bagian dari penentuan kriteria karyawan pada penjadwalan, berdasarkan kriteria yang telah ditentukan oleh perusahaan. Hasil dari penerapan metode topsis ini dapat menjadi suatu acuan nilai dengan hasil pemilihan yang sesuai dengan kriteria yang ditentukan.

Berdasarkan latar belakang masalah yang ada, maka dapat diidentifikasi menjadi beberapa masalah sebagai berikut:

1. Bagaimana membangun suatu sistem pemilihan karyawan yang sesuai

dengan kriteria dalam penjadwalan kegiatan akademik menggunakan metode TOPSIS?

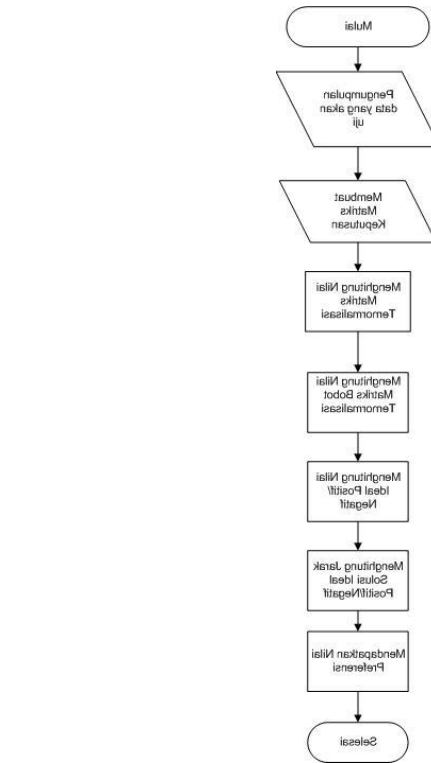
Dari data-data yang didapatkan dari proses rumusan masalah ini maka batasan masalah dari penelitian ini adalah sebagai berikut :

1. Pada Penelitian ini hanya membahas pembuatan pemilihan karyawan pada penjadwalan kegiatan akademik berdasarkan kriteria yang telah ditentukan oleh perusahaan.
2. Dalam penelitian ini menggunakan metode TOPSIS untuk memberikan acuan nilai dengan hasil kriteria yang diinginkan perusahaan.
3. Sistem yang akan dibangun menggunakan Bahasa Pemrograman PHP dengan *database* MySQL.

Berdasarkan latar belakang dan identifikasi masalah yang ada, maka akan memberikan tujuan sebagai berikut:

Membangun sistem pemilihan karyawan dalam penjadwalan kegiatan akademik menggunakan metode TOPSIS.

Pada Penelitian ini penulis menerapkan Algoritma TOPSIS pada proses perhitungan akurasi keakuratan pemilihan peserta kegiatan akademik di PT Pegadaian. Untuk menyelesaikan masalah tersebut perlu diterapkan nya sebuah metodologi, metodologi yang akan digunakan yaitu metodologi MADM (*Multiple Attribute Decision Making*) sebagai berikut :



Gambar 4. 1 Alur Metodologi Penelitian

MADM (*Multiple Attribute Decision Making*) merupakan salah satu metodologi dalam pengambilan keputusan, yang mana dalam menghasilkan sebuah keputusan akan memilih alternatif yang tidak hanya paling mendekati solusi ideal positif, akan tetapi juga paling jauh dari solusi ideal negatif. Maka langkah-langkah yang dilakukan adalah sebagai berikut :

1. Pengumpulan data yang akan di uji.
2. Membuat nilai matriks keputusan.
3. Membangun matriks keputusan ternormalisasi. Dalam TOPSIS kinerja dari setiap *alternative* dihitung dengan menggunakan Persamaan 1.
4. Membangun matriks bobot ternormalisasi. Solusi ideal positif A^+ dan negatif A^- dapat ditentukan

- berdasarkan pada rating bobot ternormalisasi seperti Persamaan 2.
5. Menentukan solusi ideal positif dan negatif Matriks solusi ideal positif dapat dihitung dengan Persamaan 3, sedangkan matriks solusi ideal negatif dapat dihitung berdasarkan Persamaan 4.
 6. Menghitung jarak setiap alternatif keputusan dari solusi ideal positif dan negatif. Jarak antara alternatif A_i dengan solusi ideal positif dapat dihitung dengan Persamaan 5.
 7. Menentukan nilai preferensi untuk setiap alternatif [30].

Pada metode MADM (*Multiple Attribute Decision Making*) terdapat beberapa fitur umum yang digunakan dalam MADM, yaitu :

1. Alternatif, adalah obyek-obyek yang berbeda dan memiliki kesempatan yang sama untuk dipilih oleh pengambil keputusan.
2. Atribut, merupakan karakteristik, komponen atau kriteria keputusan. Meskipun kebanyakan kriteria bersifat satu level, namun tidak menutup kemungkinan adanya sub kriteria yang berhubungan dengan kriteria yang telah diberikan.
3. Konfil antar kriteria, beberapa kriteria biasanya mempunyai konflik dengan kriteria antara satu dengan yang lainnya.
4. Bobot keputusan, bobot keputusan menunjukkan kepentingan relative dari setiap kriteria, Pada MADM akan

dicari bobot kepentingan dari setiap kriteria.

5. Matriks keputusan, suatu matriks keputusan yang berukuran $m \times n$, berisi elemen-elemen yang merepresentasikan rating dari alternative terhadap kriteria [31].

Sebagian besar pendekatan MADM (*Multiple Attribut Decision Making*) dilakukan dengan 2 langkah, yaitu :

1. Melakukan penindakan terhadap keputusan-keputusan yang tanggap terhadap semua tujuan pada setiap alternatif.

Melakukan perangkingan alternatif-alternatif keputusan tersebut berdasarkan hasil penindakan keputusan

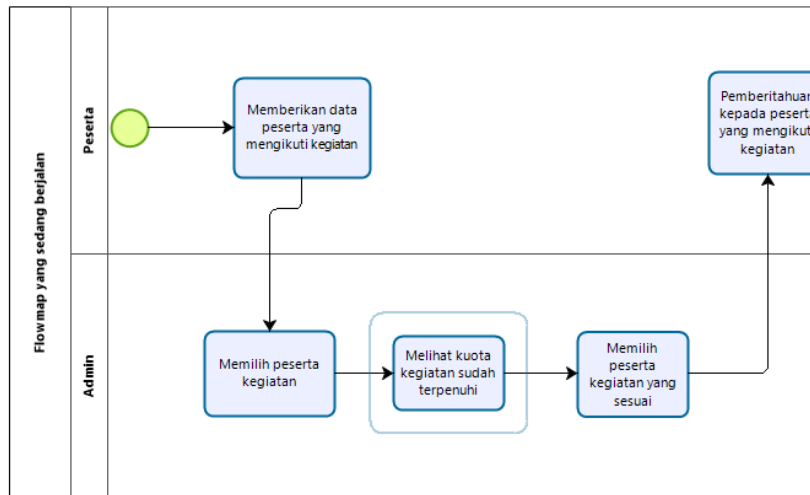
BAB V

ANALISIS DAN PERANCANGAN SISTEM

5.1 Analisis dan Perancangan Sistem

5.1.1 Analisis Sistem Berjalan

5.1.1.1 Analisis Prosedur Yang Sedang Berjalan



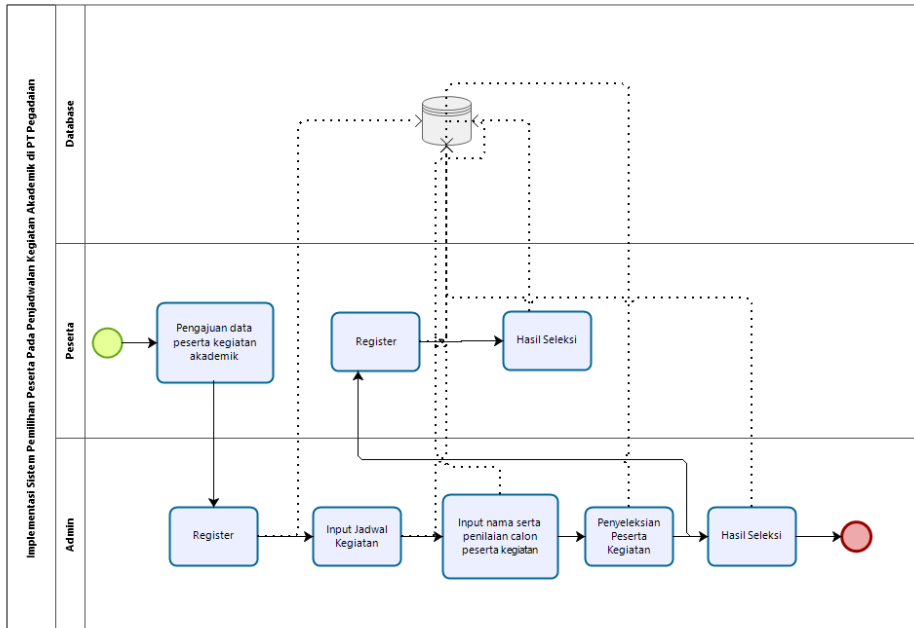
Gambar 5.1 Analisis Prosedur Yang Sedang Berjalan

Prosedur yang sedang berjalan :

1. Pada setiap kegiatan akademik yang akan diselenggarakan, peserta akan memberikan data peserta yang akan mengikuti kegiatan akademik.
2. Admin akan memilih peserta kegiatan.
3. Admin akan melihat kuota peserta kegiatan
4. Admin akan melihat apakah data peserta sudah terpenuhi sesuai dengan kuota yang diinginkan atau belum.
5. Admin akan memilih peserta sesuai dengan kegiatan yang diselenggarakan.
6. Setelah admin memilih peserta maka admin akan memberitahukan kepada peserta yang telah sesuai untuk mengikuti kegiatan.

5.1.2 Analisis Sistem Yang Akan Dibangun

5.1.2.1 Analisis Prosedur Yang Akan Dibangun



Gambar 5.2 Analisis Prosedur Yang Akan Dibangun

Prosedur yang akan dibangun :

1. Peserta melakukan pengajuan data peserta kegiatan akademik kepada admin.
2. Admin melakukan register untuk menginputkan jadwal kegiatan peserta yang akan mengikuti kegiatan akademik.

3. Setelah menginputkan jadwal admin akan menginputkan nama calon peserta kegiatan.
4. Admin melakukan penyeleksian peserta kegiatan berdasarkan nilai-nilai yang di inputkan.
5. Setelah proses penyeleksian maka akan keluar hasil berdasarkan nilai yang di inputkan.
6. Peserta akan melakukan register untuk melihat hasil pemilihan peserta kegiatan.

5.1.2.2 Analisis Dokumen Yang Akan Dibangun

Didalam sistem yang saat ini berjalan terdapat beberapa dokumen yang digunakan sebagai arsip untuk menyimpan data di pos market. Adapun dokumen yang dimaksud adalah sebagai berikut :

Tabel 5.1 Dokumen Admin

Dibuat Oleh	Admin
Dibuat Untuk	Admin
Isi	Berupa jadwal kegiatan akademik dan kuota peserta, pemilihan karyawan dalam kegiatan akademik

Frekuensi	Dibuat form pemilihan peserta akademik
Tujuan	Pendataan peserta, jumlah peserta dan nama peserta kegiatan

Tabel 5. 2 Dokumen *User*

Dibuat oleh	Admin
Dibuat untuk	User
Isi	Berupa form hasil penilaian peserta kegiatan akademik
Frekuensi	Dibuat form untuk melihat hasil penilaian
Tujuan	Melihat penilaian peserta kegiatan akademik

5.1.2.3 Analisis Metode TOPSIS

Metode ini merupakan metode kompromi yang banyak digunakan untuk menangani berbagai kriteria masalah pengambilan keputusan. Dasar pemikiran metode TOPSIS adalah bahwa alternatif yang dipilih harus memiliki jarak terdekat solusi positif ideal dan jarak

terjauh dari solusi *negative* ideal. Bobot kriteria dalam metode TOPSIS diberikan apriori. Selain itu, solusi positif-ideal dan solusi *negative* ideal langsung dihasilkan dari matriks keputusan.

Langkah-langkah metode TOPSIS dapat dilakukan seperti yang sudah dijelaskan pada BAB II 2.1.8. Berikut merupakan tahap peneerapan metode TOPSIS :

1. Data Kriteria dan Bobot

No	Kriteria	Atribut	Bobot
1	Grade	Benefit	20
2	Kesesuaian Proker Corpu	Benefit	20
3	Keikutsertaan	Cost	15
4	Tema (Jobdesk)	Benefit	25
5	Sharing	Benefit	20

Deskripsi Bobot :

- a. Grade (20) = Untuk meningkatkan *performance* mengenai kesesuaian dengan tema

- b. Kesesuaian Proker Corpu (20) = Untuk mendukung program kerja dari Coporate University
- c. Keikutsertaan (15) = Mengenai keadilan keikutsertaan karyawan dalam kegiatan pertahun
- d. Tema (Jobdesk) (25) = Untuk meningkatkan kompetensi karyawan sehingga diharapkan dapat meningkatkan *Performance* karyawan dalam bekerja.
- e. Sharing (20) = Materi yang telah diseminarkan harus di bagikan untuk menghidupkan sistem KMS (*Knowledge Management System*) PT Pegadaian (Persero).

2. Data Matriks Berdasarkan Kriteria

Nama	<i>Gra</i> <i>de</i>	Kesesua ian Proker	Keikutsert aan	Te ma	<i>Shari</i> <i>ng</i>
Eka Pebrians yah	5	4	1	5	5

Supriyanto	5	5	3	5	5
Endang Pertiwi	5	4	1	5	5
Muh Ariyadi	5	4	1	5	5
Hertin Maulida	5	4	1	5	5
Elvi Rofiqatul Hidayah	5	5	4	5	5
Teguh Budi Ismanto	5	4	2	5	5
Eri Mardianto	5	4	2	5	5
Sriyani	5	4	1	5	5

Deskripsi Penilaian :

1. *Grade*

- a. Tenaga Pembantu (1)
- b. Tenaga Keamanan (2)
- c. Karyawan Grade 4-7 (3)
- d. Karyawan Grade 8-11 (4)
- e. Karyawan Grade 12-15 (5)

2. Kesesuaian Proker Corpu

Nilai rata-rata dari penilaian grade,tema,*sharing*, dan keikutsertaan

3. Keikutsertaan

- a. (1-5) Berdasarkan banyaknya keikutsertaan peserta dalam jangka waktu setahun

4. Tema

- a. Keterkaitan dengan bagian tersebut Grade 4-7 (1)
- b. Keterkaitan dengan bagian tersebut Grade 8-11 (2)
- c. Keterkaitan dengan jobdesk dan seminar Grade 4-7 (3)
- d. Keterkaitan dengan jobdesk dan seminar Grade 8-11 (4)
- e. Keterkaitan dengan jobdesk dan seminar Grade 12-15 (5)

5. *Sharing* (Membagikan Materi Seminar di KMS)

- a. Grade 1 (1)
- b. Grade 2-3 (2)
- c. Grade 4-7 (3)
- d. Grade 8-11 (4)
- e. Grade 12-15 (5)

3. Pemberian bobot pada masing-masing peserta, kemudian pencarian nilai dengan rumus sebagai berikut :

$$|X_n| = \sqrt{\sum_{i=1}^m x_{ij}^2}$$

$$|X_1| = \sqrt{(5)^2 + (5)^2 + (5)^2 + (5)^2 +$$

$$(5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 = 15,0000$$

$$|X_2| = \sqrt{(4)^2 + (5)^2 + (4)^2 + (4)^2 + (4)^2 + (5)^2 + (4)^2 + (4)^2 + (4)^2 = 12,7279$$

$$|X_3| = \sqrt{(1)^2 + (3)^2 + (1)^2 + (1)^2 + (1)^2 + (4)^2 + (2)^2 + (2)^2 + (1)^2 = 6,1644$$

$$|X_4| = \sqrt{(5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 = 15,0000$$

$$|X_5| = \sqrt{(5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 + (5)^2 = 15,0000$$

1. Setelah rumus X_n maka selanjutnya menggunakan rumus rij :

$$r_{ij} = \frac{x^{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

$R_{10} = 5/15,0000 = 0,333$	$R_{20} = 4/12,7279 = 0,314$	$R_{30} = 1/6,$
$R_{11} = 5/15,0000 = 0,333$	$R_{21} = 5/12,7279 = 0,393$	$R_{31} = 3/6,$
$R_{12} = 5/15,0000 = 0,333$	$R_{22} = 4/12,7279 = 0,314$	$R_{32} = 1/6,$
$R_{13} = 5/15,0000 = 0,333$	$R_{23} = 4/12,7279 = 0,314$	$R_{33} = 1/6,$
$R_{14} = 5/15,0000 = 0,333$	$R_{24} = 4/12,7279 = 0,314$	$R_{34} = 1/6,$
$R_{15} = 5/15,0000 = 0,333$	$R_{25} = 5/12,7279 = 0,393$	$R_{35} = 4/6,$
$R_{16} = 5/15,0000 = 0,333$	$R_{26} = 4/12,7279 = 0,314$	$R_{36} = 2/6,$
$R_{17} = 5/15,0000 = 0,333$	$R_{27} = 4/12,7279 = 0,314$	$R_{37} = 2/6,$
$R_{18} = 5/15,0000 = 0,333$	$R_{28} = 4/12,7279 = 0,314$	$R_{38} = 1/6,$
$R_{40} = 5/15,0000 = 0,333$	$R_{50} = 5/15,0000 = 0,333$	
$R_{41} = 5/15,0000 = 0,333$	$R_{51} = 5/15,0000 = 0,333$	
$R_{42} = 5/15,0000 = 0,333$	$R_{52} = 5/15,0000 = 0,333$	
$R_{43} = 5/15,0000 = 0,333$	$R_{53} = 5/15,0000 = 0,333$	
$R_{44} = 5/15,0000 = 0,333$	$R_{54} = 5/15,0000 = 0,333$	
$R_{45} = 5/15,0000 = 0,333$	$R_{55} = 5/15,0000 = 0,333$	

$R_{46} = 5/15,0000 = 0,333$	$R_{56} = 5/15,0000 = 0,333$
$R_{47} = 5/15,0000 = 0,333$	$R_{57} = 5/15,0000 = 0,333$
$R_{48} = 5/15,0000 = 0,333$	$R_{58} = 5/15,0000 = 0,333$

2. Menghitung matriks ternormalisasi yang sudah terbobot dengan bobot yang sudah ditentukan (W) = $[4,5,3,5,2]$, dengan rumus :

$$y_{ij} = w_i * r_{ij}$$

$R_{10} = 20 * 0,333 = 6,667$	$R_{20} = 20 * 0,314 = 6,285$	$R_{30} = 15 *$
$R_{11} = 20 * 0,333 = 6,667$	$R_{21} = 20 * 0,393 = 7,857$	$R_{31} = 15 *$
$R_{12} = 20 * 0,333 = 6,667$	$R_{22} = 20 * 0,314 = 6,285$	$R_{32} = 15 *$
$R_{13} = 20 * 0,333 = 6,667$	$R_{23} = 20 * 0,314 = 6,285$	$R_{33} = 15 *$
$R_{14} = 20 * 0,333 = 6,667$	$R_{24} = 20 * 0,314 = 6,285$	$R_{34} = 15 *$
$R_{15} = 20 * 0,333 = 6,667$	$R_{25} = 20 * 0,393 = 7,857$	$R_{35} = 15 *$
$R_{16} = 20 * 0,333 = 6,667$	$R_{26} = 20 * 0,314 = 6,285$	$R_{36} = 15 *$
$R_{17} = 20 * 0,333 = 6,667$	$R_{27} = 20 * 0,314 = 6,285$	$R_{37} = 15 *$
$R_{18} = 20 * 0,333 = 6,667$	$R_{28} = 20 * 0,314 = 6,285$	$R_{38} = 15 *$
$R_{41} = 25 * 0,333 = 8,333$	$R_{51} = 20 * 0,333 = 6,667$	

$R_{42} = 25 * 0,333 = 8,333$	$R_{52} = 20 * 0,333 = 6,667$
$R_{43} = 25 * 0,333 = 8,333$	$R_{53} = 20 * 0,333 = 6,667$
$R_{44} = 25 * 0,333 = 8,333$	$R_{54} = 20 * 0,333 = 6,667$
$R_{45} = 25 * 0,333 = 8,333$	$R_{55} = 20 * 0,333 = 6,667$
$R_{46} = 25 * 0,333 = 8,333$	$R_{56} = 20 * 0,333 = 6,667$
$R_{47} = 25 * 0,333 = 8,333$	$R_{57} = 20 * 0,333 = 6,667$
$R_{48} = 25 * 0,333 = 8,333$	$R_{58} = 20 * 0,333 = 6,667$

3. Menentukan Solusi Ideal Matriks (A^+)

$$A^+ = \max(y_1^+, y_2^+, \dots, y_n^+)$$

$$Y_1 = \max(6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667) = 6.667$$

$$Y_2 = \max(6.285, 7.857, 6.285, 6.285, 6.285, 7.857, 6.285, 6.285, 6.285) = 7.857$$

$$Y_3 = \max(2.433, 7.300, 2.433, 2.433, 2.433, 9.733, 4.867, 4.867, 2.433) = 9.733$$

$$Y_4 = \max(8.333, 8.333, 8.333, 8.333, 8.333, 8.333, 8.333, 8.333, 8.333) = 8.333$$

$$Y_5 = \max(6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667) = 6.667$$

4. Menentukan Solusi Ideal Matriks (A^-)

$$A^- = \min(y_1^+, y_2^+, \dots, y_n^+)$$

$$Y_1 = \min(6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667) = 6.667$$

$$Y_2 = \min(6.285, 7.857, 6.285, 6.285, 6.285, 7.857, 6.285, 6.285, 6.285) = 6.285$$

$$Y_3 = \min(2.433, 7.300, 2.433, 2.433, 2.433, 9.733, 4.867, 4.867, 2.433) = 2.433$$

$$Y_4 = \min(8.333, 8.333, 8.333, 8.333, 8.333, 8.333, 8.333, 8.333, 8.333) = 8.333$$

$$Y_5 = \min(6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667, 6.667) = 6.667$$

5. Menghitung Jarak Solusi Ideal Positif (D^+)

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}$$

$$D_1^+$$

$$= \sqrt{(6.667 - 6.667)^2 + (6.285 - 7.857)^2 + (2.433 - 9.733)^2 + (8.333 - 8.333)^2 + (6.667 - 6.667)^2} = 7.467$$

$$D_2^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (7,857 - 7,857)^2 + (7,300 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 2,433$$

$$D_3^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 7,857)^2 + (2,433 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 7,467$$

$$D_4^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 7,857)^2 + (2,433 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 7,467$$

$$D_5^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 7,857)^2 + (2,433 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 7,467$$

$$D_6^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (7,857 - 7,857)^2 + (9,733 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 0,000$$

$$D_7^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 7,857)^2 + (4,867 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 5,114$$

$$D_8^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 7,857)^2 + (4,867 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 5,114$$

$$D_9^+$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 7,857)^2 + (2,433 - 9,733)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2}$$

$$= 7,467$$

6. Menghitung Jarak Solusi Ideal Negatif (D^-)

$$D_i^- = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}$$

$$D_1^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 6,285)^2 + (2,433 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 0,000$$

$$D_2^- =$$

$$\sqrt{(6,667 - 6,667)^2 + (7,857 - 6,285)^2 + (7,300 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 5,114$$

$$D_3^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 6,285)^2 + (2,433 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 0,000$$

$$D_4^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 6,285)^2 + (2,433 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 0,000$$

$$D_5^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 6,285)^2 + (2,433 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 0,000$$

$$D_6^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (7,857 - 6,285)^2 + (9,733 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 7,467$$

$$D_7^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 6,285)^2 + (4,867 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 2,433$$

$$D_8^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 6,285)^2 + (4,867 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 2,433$$

$$D_9^-$$

$$= \sqrt{(6,667 - 6,667)^2 + (6,285 - 6,285)^2 + (2,433 - 2,433)^2 + (8,333 - 8,333)^2 + (6,667 - 6,667)^2} = 0,000$$

7. Menghitung Nilai Preferensi

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

$$V_1 = \frac{0,000}{0,000} + 7,467 = 0,0000$$

$$V_2 = \frac{5,114}{5,114} + 2,433 = 0,6776$$

$$V_3 = \frac{0,000}{0,000} + 7,467 = 0,0000$$

$$V_4 = \frac{0,000}{0,000} + 7,467 = 0,0000$$

$$V_5 = \frac{0,000}{0,000} + 7,467 = 0,0000$$

$$V_6 = \frac{7,467}{7,467} + 0,000 = 1,0000$$

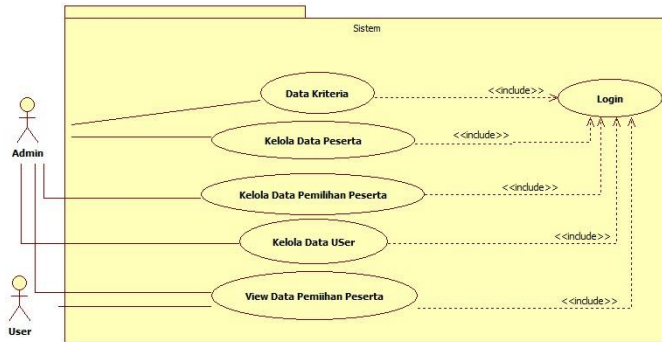
$$V_7 = \frac{2,433}{2,433} + 5,114 = 0,3224$$

$$V_8 = \frac{2,433}{2,433} + 5,114 = 0,3224$$

$$V_9 = \frac{0,000}{0,000} + 7,467 = 0,0000$$

5.1.2.4 UML (Unified Modelling Language)

5.1.2.4.1 Usecase Diagram



Gambar 5.3 Usecase Diagram

5.1.2.4.1.1 Definisi Usecase

Tabel 5.3 Definisi Usecase

No	Use Case	Deskripsi
1.	<i>Login</i>	Melakukan proses <i>login</i>
2.	Data Kriteria	Merupakan serangkaian data kriteria yang berlangsung untuk akademik karyawan
3.	Kelola Data Peserta	Merupakan serangkaian aktivitas yang untuk mengelola data peserta yang mer kegiatan akademik.
4.	Kelola Data Pemilihan Peserta	Merupakan serangkaian aktivitas yang untuk mengelola data untuk pemilihan

5.	Kelola Data User	Merupakan serangkaian aktivitas yang untuk mengelola user atau pengguna.
6.	View Data Pemilihan Peserta	Melihat hasil dari data pemilihan peserta

5.1.2.4.1.2 Usecase Diagram Skenario

Use Case Skenario mendeskripsikan urutan langkah-langkah dalam proses sistem, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berikut ini penjelasan dari masing-masing skenario tersebut:

5.1.2.4.1.2.1 Skenario *Usecase Login* Admin

Tabel 5. 4 Skenario *Usecase Login* Admin

Identifikasi	
Nomor	UC1
Nama	<i>Login admin</i>
Tujuan	Memberikan hak akses admin terhadap sistem dengan melakukan validasi terhadap <i>username</i> dan <i>password</i> yang dimasukkan oleh pengguna.
Deskripsi	
Aktor	Admin
Skenario Utama	

Kondisi Awal	<i>Form Login</i> sudah tersedia
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan password	2. <i>Form Login</i> akan menampilkan <i>username</i> , <i>password</i> dan untuk <i>password</i> ditampilkan dalam bentuk kode ‘•’ untuk jaminan keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username</i> , <i>password</i> yang telah dimasukkan dengan menekan tombol <i>Ok</i> .	4. Aplikasi melakukan validasi terhadap <i>username</i> , <i>password</i> yang telah dimasukkan oleh pengguna dengan melakukan query pada basis data.
Kondisi Akhir	Jika pada akhir interaksi <i>username</i> dan <i>password</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.
Alur Alternatif kondisi akhir	
a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal login	

5.1.2.4.1.2.2 Skenario Usecase Diagram Login User

Tabel 5.5 Skenario Use Case Login User

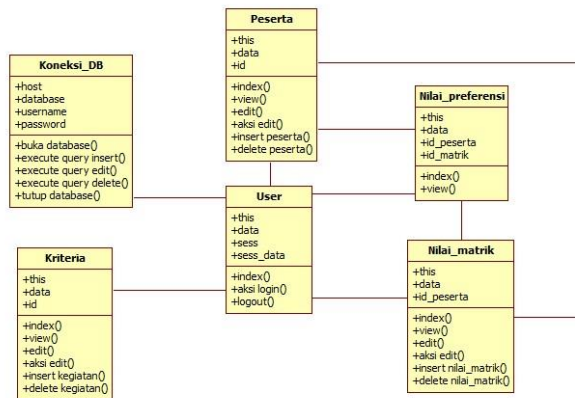
Identifikasi	
Nomor	UC2
Nama	<i>Login User</i>
Tujuan	Memberikan hak akses peserta ter dengan melakukan validasi terhadap dan <i>password</i> yang dimasukan oleh
Deskripsi	
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	<i>Form Login</i> sudah tersedia
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan <i>password</i>	2. <i>Form Login</i> akan menampilkan <i>username</i> , <i>password</i> dan untuk <i>pa</i> ditampilkan dalam bentuk kode ‘•’ untuk jaminan keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username</i> , <i>password</i> yang telah dimasukkan dengan menekan tombol <i>Ok</i> .	4. Aplikasi melakukan validasi terhadap <i>username</i> , <i>password</i> yang telah di oleh pengguna dengan melakukan pada basis data.

Kondisi Akhir	Jika pada akhir interaksi <i>username</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi dengan hak aksesnya.
Alur Alternatif kondisi akhir a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal <i>login</i> .	

5.1.2.4.2 Class Diagram

Class Diagram menggambarkan *class* dan hubungan antar *class* didalam sistem. *Class* digambarkan dengan sebuah kotak dibagi menjadi tiga bagian. Bagian paling atas diisikan nama *class*, bagian tengah diisikan variabel yang dimiliki *class*, dan bagian bawah diisikan *method-method* dari *class*.

Dalam proses pembuatan aplikasi ini *class diagram* yang akan digunakan adalah sebagai berikut :



Gambar 5.4 Class Diagram

5.1.2.4.3 Interaction Diagram

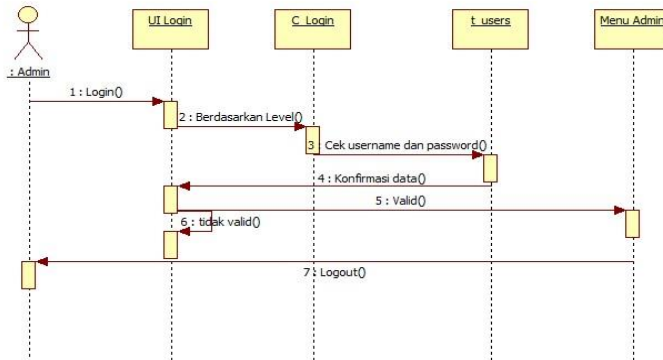
Interaction diagram adalah diagram yang menggambarkan interaksi dalam sistem, antara aplikasi, *user* dan *database*. *Interaction diagram* didefinisikan dengan dua tipe diagram yaitu *sequence diagram* dan *collaboration diagram*.

5.1.2.4.4 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari

diagram, pada diagram ini, dimensi vertikal merepresentasikan waktu.

5.1.2.4.4.1 Sequence Diagram Login Admin

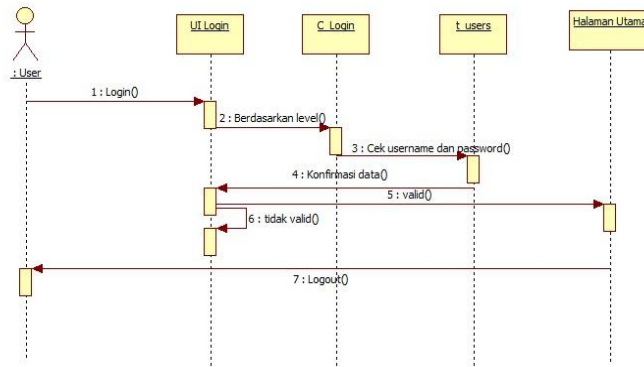


Gambar 5.5 Sequence Diagram Login Admin

Penjelasan :

Pada *sequence diagram* ini menjelaskan proses *login*. *Admin* mulai menjalankan aplikasi akan tampil *form login*. Kemudian *admin* memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di tabel *user*. Jika sesuai maka akan mendapatkan tampilan halaman utama *admin*.

5.1.2.4.4.2 Sequence Diagram Login User

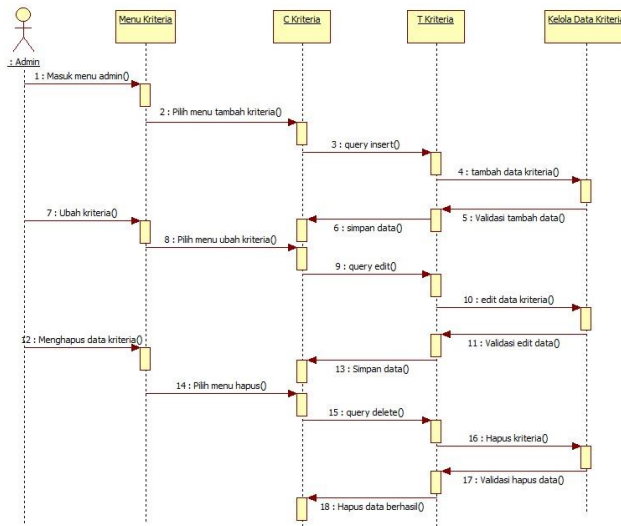


Gambar 5.6 Sequence Diagram Login User

Penjelasan :

Pada *sequence diagram* ini menjelaskan proses *login*. *User* mulai menjalankan aplikasi akan tampil *form login*. Kemudian *user* memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di tabel *user*. Jika sesuai maka *user* dapat melakukan pendaftaran kegiatan akademik.

5.1.2.4.4.3 Sequence Diagram Kelola Data Kriteria

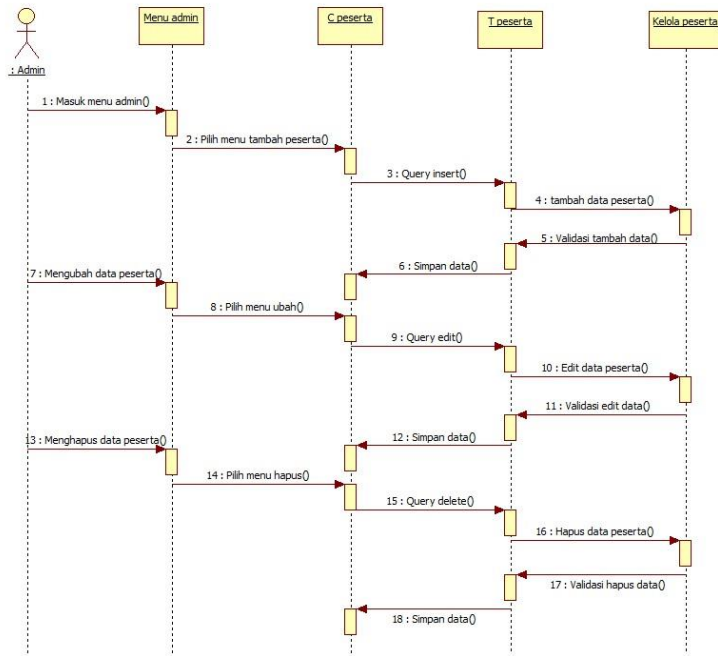


Gambar 5.7 Sequence Diagram Kelola Data Kriteria

Penjelasan :

Pada *sequence diagram* ini menjelaskan proses kelola data kriteria. Admin mulai menjalankan aplikasi akan tampil *form login*. Kemudian admin memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di tabel *database*. Jika sesuai maka admin dapat melakukan pengelolaan data kegiatan dengan *meng input*, ubah dan hapus data pada form tersebut.

5.1.2.4.4.4 Sequence Diagram Kelola Data Peserta



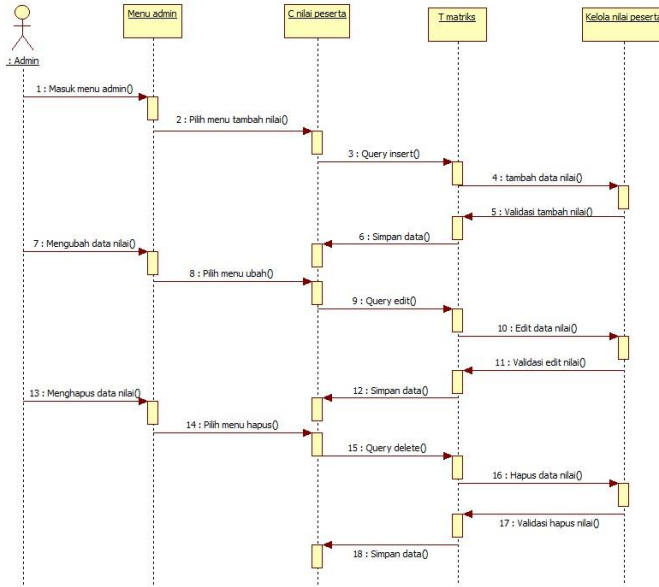
Gambar 5.8 Sequence Diagram Kelola Data Peserta

Penjelasan :

Pada *sequence diagram* ini menjelaskan proses kelola data peserta. Aktor memilih menu kelola data peserta. Namun sebelumnya aktor harus melakukan *login* terlebih dahulu. Maka sistem akan melakukan koneksi *database*. Selanjutnya sistem akan memvalidasi. Maka

sistem memuat tampilan data peserta, pada tampilan ini aktor dapat melakukan *input*, ubah dan hapus.

5.1.2.4.4.5 Sequence Diagram Kelola Pemilihan Peserta



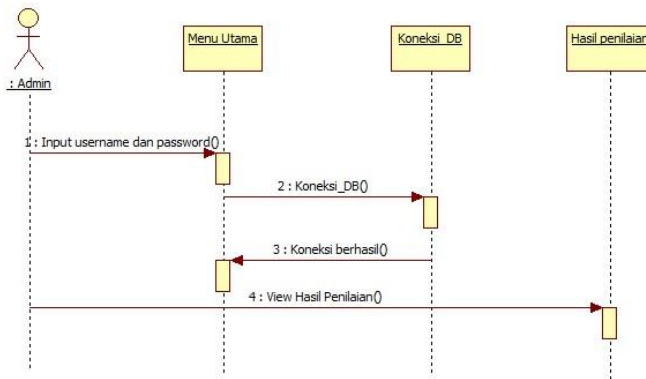
Gambar 5.9 Sequence Diagram Kelola Data Pemilihan Peserta

Penjelasan :

Pada *sequence diagram* ini menjelaskan proses kelola data pemilihan peserta. Aktor memilih menu kelola data pemilihan peserta. Namun sebelumnya aktor harus melakukan *login* terlebih dahulu. Maka sistem akan melakukan koneksi *database*. Selanjutnya sistem akan

memvalidasi. Pada saat ini sistem memuat tampilan data pemilihan peserta, pada tampilan ini aktor dapat melakukan *input* nilai peserta, ubah dan hapus.

5.1.2.4.4.6 Sequence Diagram Admin Hasil Penilaian

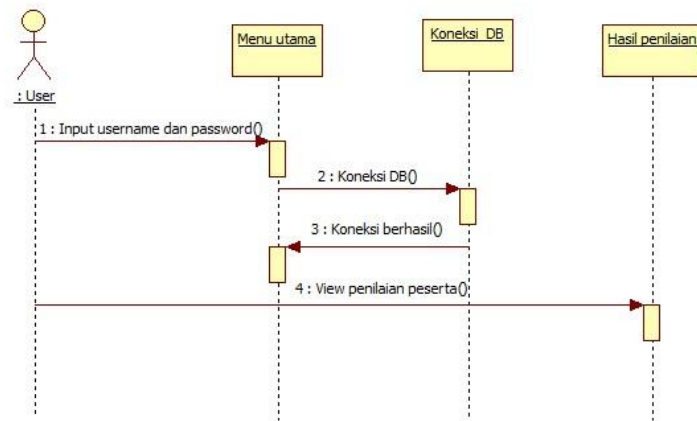


Gambar 5.10 Sequence Diagram Admin Hasil Penilaian

Penjelasan :

Pada *sequence diagram* ini menjelaskan proses hasil penilaian peserta. Aktor memilih menu hasil penilaian. Namun sebelumnya aktor harus melakukan *login* terlebih dahulu. Maka sistem akan melakukan koneksi *database*. Selanjutnya sistem akan memvalidasi. Pada saat ini sistem memuat tampilan data hasil penilaian peserta, pada tampilan ini aktor dapat melihat hasil penilaian dari perhitungan sebelumnya.

5.1.2.4.4.7 Sequence Diagram User Hasil Penilaian

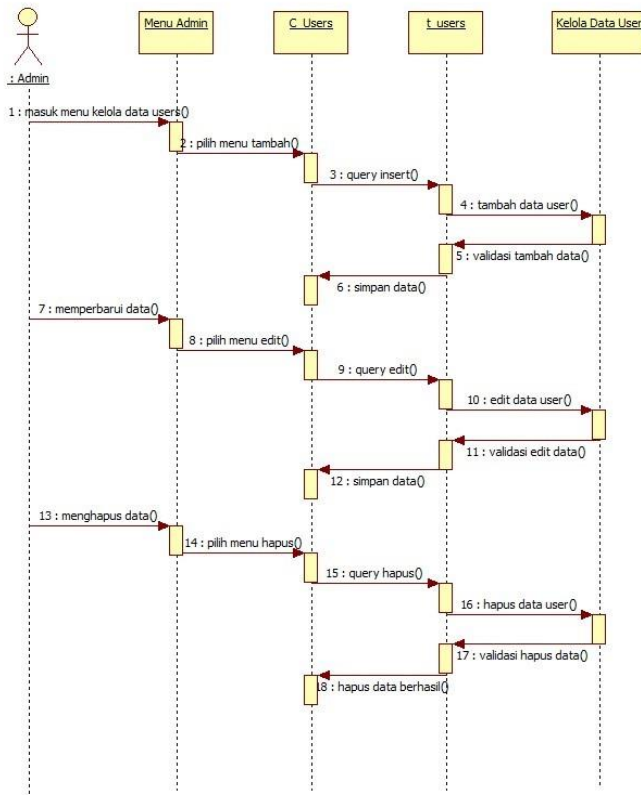


Gambar 5.11 Sequence Diagram Admin Hasil Penilaian

Penjelasan :

Pada *sequence diagram* ini menjelaskan proses hasil penilaian peserta. Aktor memilih menu hasil penilaian. Namun sebelumnya aktor harus melakukan *login* terlebih dahulu. Maka sistem akan melakukan koneksi *database*. Selanjutnya sistem akan memvalidasi. Pada saat ini sistem memuat tampilan data hasil penilaian peserta, pada tampilan ini aktor dapat melihat hasil penilaian dari perhitungan sebelumnya.

5.1.2.4.4.7 Sequence Diagram User Hasil Penilaian



Gambar 5. 12 Sequence Diagram Kelola Data User

Penjelasan :

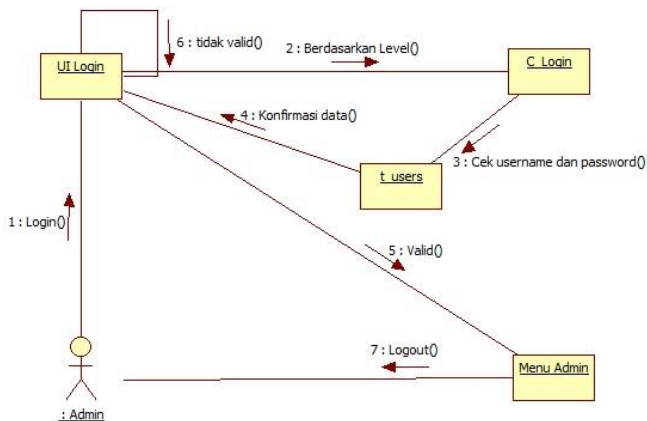
Pada *sequence diagram* ini menjelaskan proses pengelolaan *user*. Aktor memilih menu *user*. Namun sebelumnya aktor harus melakukan *login* terlebih dahulu.

Maka sistem akan melakukan koneksi *database*. Selanjutnya sistem akan memvalidasi. Pada saat ini sistem memuat tampilan data *user*, pada tampilan ini aktor dapat menambah, mengubah dan menghapus data *user*.

5.1.2.4.5 Collaboration Diagram

Collaboration Diagram adalah suatu diagram yang memperlihatkan atau menampilkan pengorganisasian interaksi yang terdapat disekitar objek (seperti halnya *sequence diagram*) dan hubungannya terhadap yang lainnya. *Collaboration Diagram* lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan/*message*.

5.1.2.4.5.1 Collaboration Diagram Login Admin

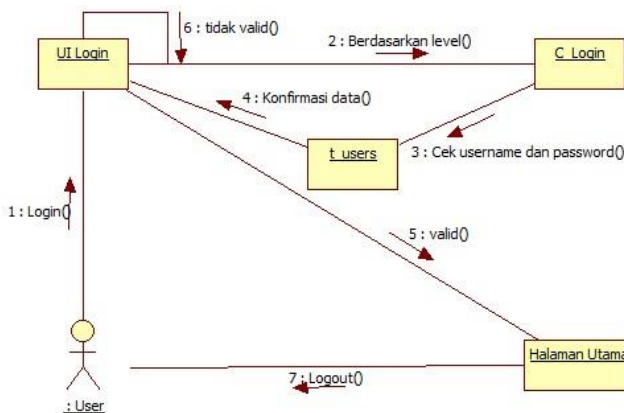


Gambar 5.13 Collaboration Diagram Login Admin

Penjelasan :

Pada *collaboration diagram* ini menjelaskan proses *login*. Admin mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian *admin* memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di tabel *user*. Jika sesuai maka akan mendapatkan tampilan *form* halaman utama *admin*.

5.1.2.4.5.2 Collaboration Diagram Login User



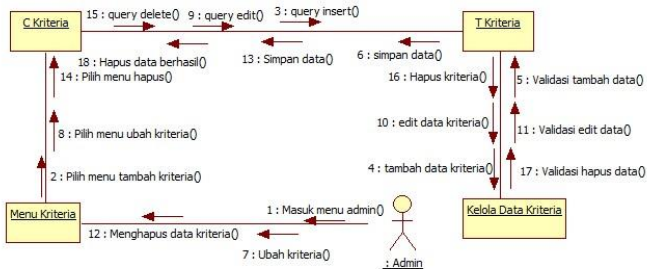
Gambar 5.14 Collaboration Diagram Login User

Penjelasan :

Pada *collaboration diagram* ini menjelaskan proses *login*. User mulai menjalankan aplikasi dan akan

tampil *form login*. Kemudian *user* memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di tabel *user*. Jika sesuai maka akan mendapatkan tampilan *form* halaman utama.

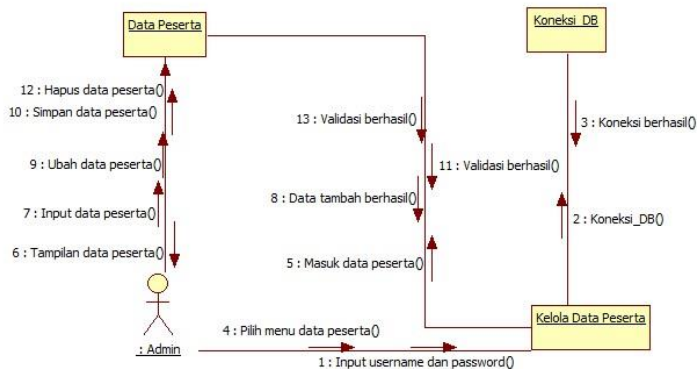
5.1.2.4.5.3 Collaboration Diagram Kelola Data Kriteria



Gambar 5.15 Collaboration Diagram Kelola Data Kegiatan

Pada *collaboration diagram* ini menjelaskan proses mengelola data kegiatan. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini maka dilakukan koneksi *database* terlebih dahulu kemudian masuk ke menu kelola data kegiatan. Setelah *form* tampil, *user* dapat melakukan operasi tambah data, ubah data dan hapus data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

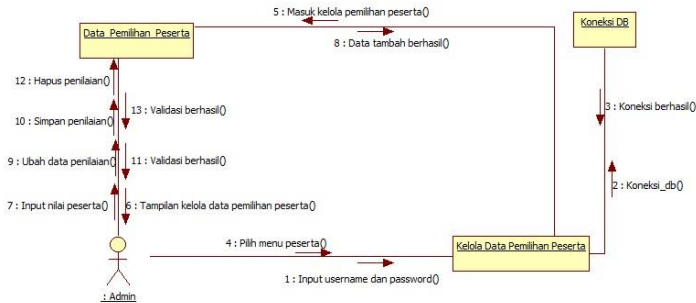
5.1.2.4.5.4 Collaboration Diagram Kelola Data Peserta



Gambar 5.16 Collaboration Diagram Kelola Data Peserta

Pada *collaboration diagram* ini menjelaskan proses mengelola data kegiatan. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini maka dilakukan koneksi *database* terlebih dahulu kemudian masuk ke menu kelola data peserta. Setelah *form* tampil, *user* dapat melakukan operasi tambah data, ubah data dan hapus data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

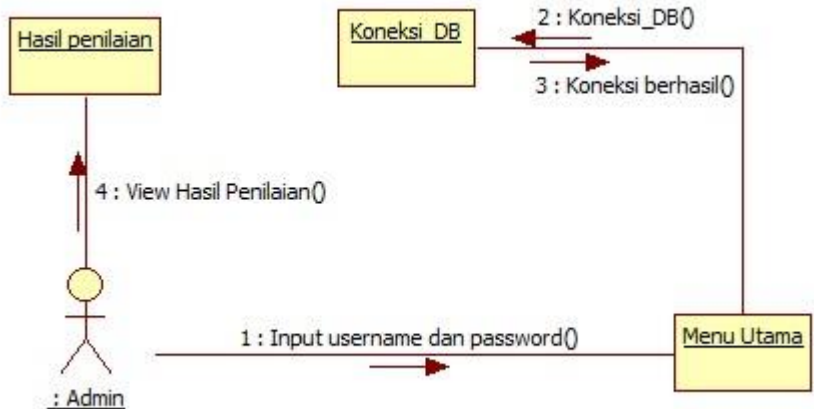
5.1.2.4.5.5 Collaboration Diagram Kelola Data Pemilihan Peserta



Gambar 5.17 Collaboration Diagram Kelola Data Pemilihan Peserta

Pada *collaboration diagram* ini menjelaskan proses mengelola data kegiatan. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini maka dilakukan koneksi *database* terlebih dahulu kemudian masuk ke menu data hasil penilaian peserta. Setelah *form* tampil, *user* dapat melakukan operasi tambah data, ubah data, hitung nilai data dan hapus data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

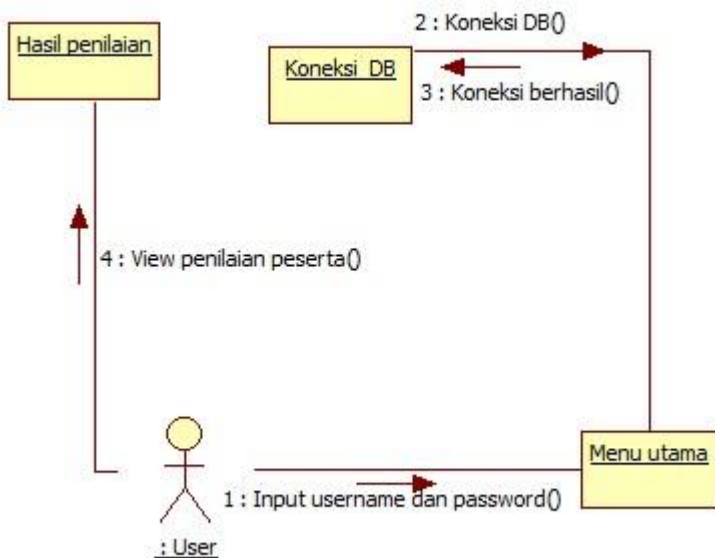
5.1.2.4.5.6 Collaboration Diagram Admin Hasil Pemilihan Peserta



Gambar 5.18 Collaboration Diagram Admin Hasil Pemilihan Peserta

Pada *collaboration diagram* ini menjelaskan proses mengelola data kegiatan. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini maka dilakukan koneksi *database* terlebih dahulu kemudian masuk ke menu data hasil pemilihan peserta. Setelah *form* tampil, *user* dapat melakukan operasi melihat data hasil pemilihan.

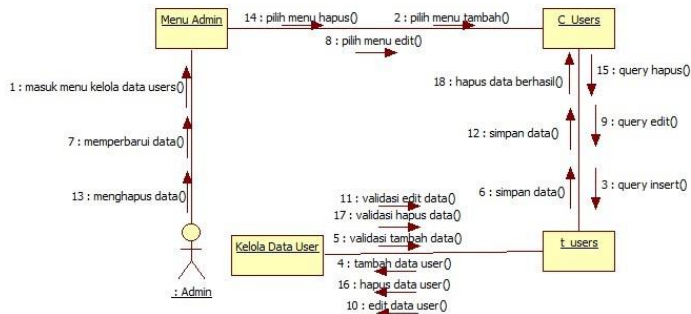
5.1.2.4.5.7 Collaboration Diagram User Hasil Pemilihan Peserta



Gambar 5.19 Collaboration Diagram User Hasil Pemilihan Peserta

Pada *collaboration diagram* ini menjelaskan proses mengelola data kegiatan. *User* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama user. Untuk memulai kelola ini maka dilakukan koneksi *database* terlebih dahulu kemudian masuk ke menu data hasil pemilihan peserta. Setelah *form* tampil, *user* dapat melakukan operasi melihat data hasil pemilihan.

5.1.2.4.5.8 Collaboration Diagram Kelola Data User



Gambar 5.20 Collaboration Diagram Kelola Data User

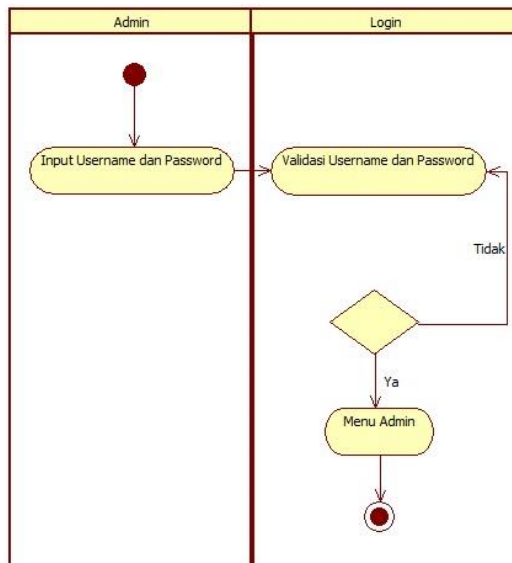
Pada *collaboration diagram* ini menjelaskan proses mengelola data *User*. Admin melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama user. Untuk memulai kelola ini maka dilakukan koneksi *database* terlebih dahulu kemudian

masuk ke menu *User*. Setelah *form* tampil, admin dapat melakukan operasi menambah, mengubah dan menghapus.

5.1.2.4.6 Activity Diagram

Activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna atau sistem, orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

5.1.2.4.6.1 Activity Diagram Login Admin

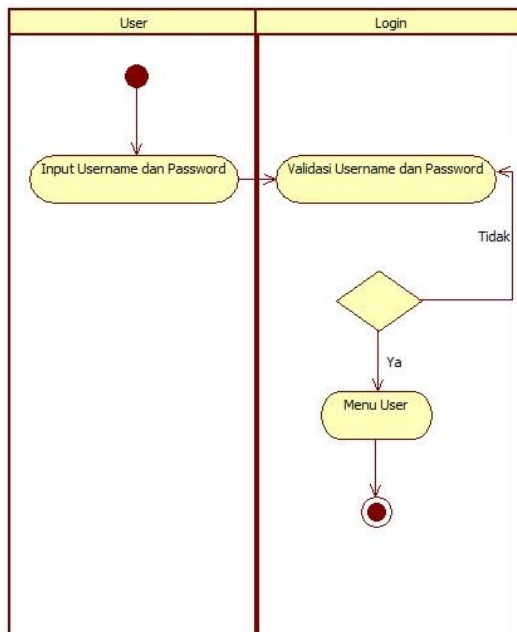


Gambar 5.21 Activity Diagram Login Admin

Penjelasan :

Aktor yang dapat *login* adalah *admin*. Admin mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama admin, tetapi jika tidak *valid* akan kembali ke *form login*.

5.1.2.4.6.2 Activity Diagram Login User

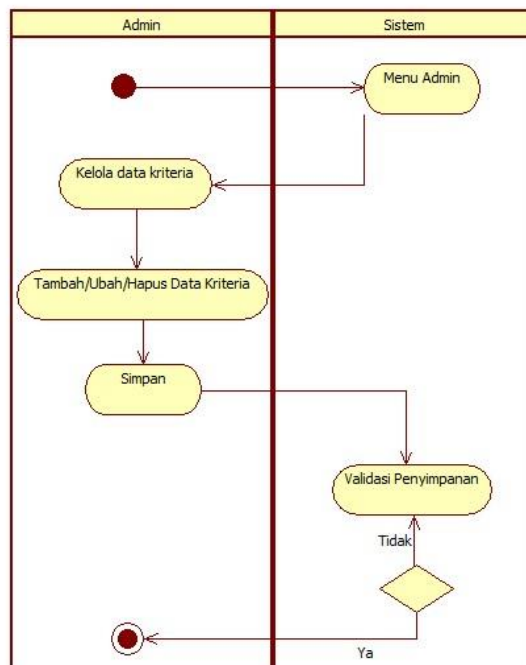


Gambar 5.22 Activity Diagram Login User

Penjelasan :

Aktor yang dapat *login* adalah *user*. *User* mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil halaman utama, tetapi jika tidak *valid* akan kembali ke *form login*.

5.1.2.4.6.3 Activity Diagram Kelola Data Kriteria

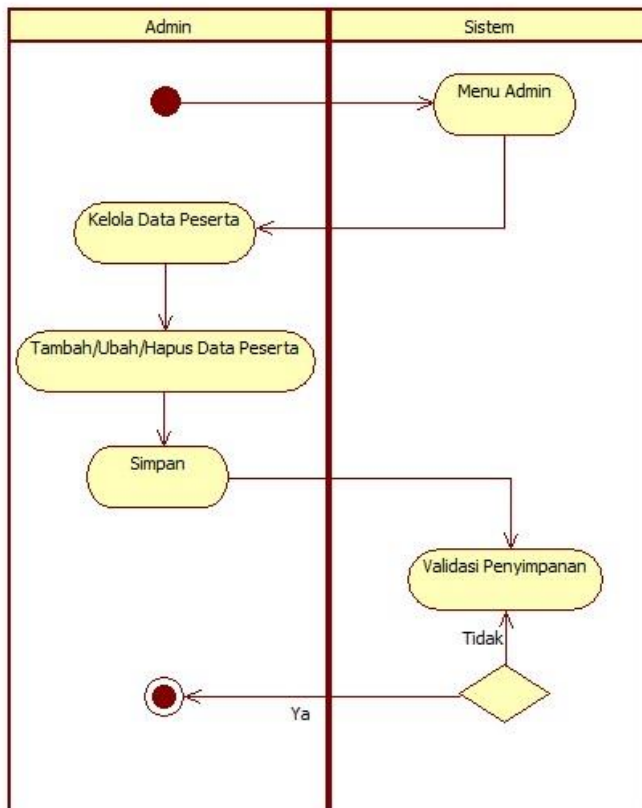


Gambar 5.23 Activity Diagram Kelola Data Kriteria

Penjelasan :

Pada *activity diagram* ini menjelaskan proses mengelola data peserta. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini, masuk ke menu kelola data peserta, data peserta ini dapat melakukan beberapa aksi yaitu tambah data, ubah data dan hapus data kegiatan, kemudian simpan jika data tidak dapat tersimpan maka sistem akan melakukan validasi kepada *database* jika ya maka data kegiatan telah selesai.

5.1.2.4.6.4 Activity Diagram Kelola Data Peserta



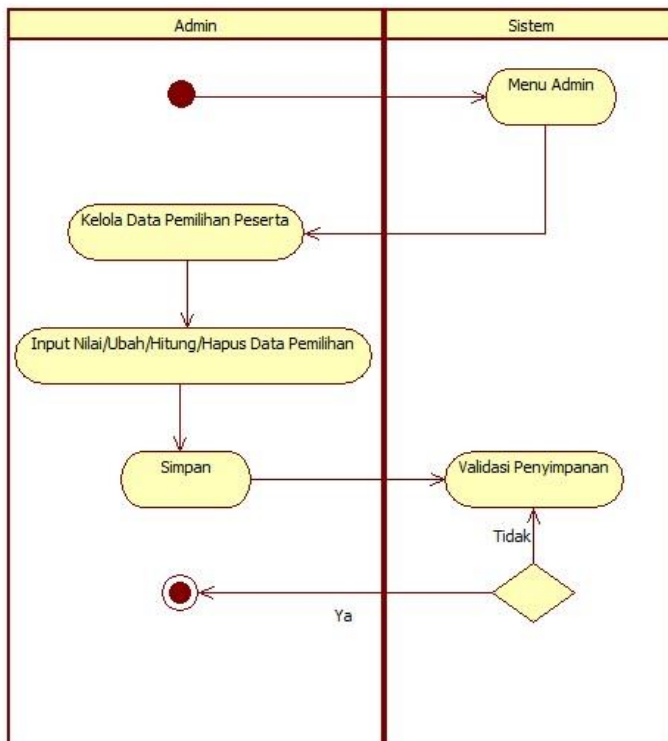
Gambar 5.24 Activity Diagram Kelola Data Peserta

Penjelasan :

Pada *activity diagram* ini menjelaskan proses mengelola data peserta. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu

utama admin. Untuk memulai kelola ini, masuk ke menu kelola data peserta, data peserta ini dapat melakukan beberapa aksi yaitu tambah data, ubah data dan hapus data kegiatan, kemudian simpan jika data tidak dapat tersimpan maka sistem akan melakukan validasi kepada *database* jika ya maka data kegiatan telah selesai.

5.1.2.4.6.5 Activity Diagram Pemilihan Peserta

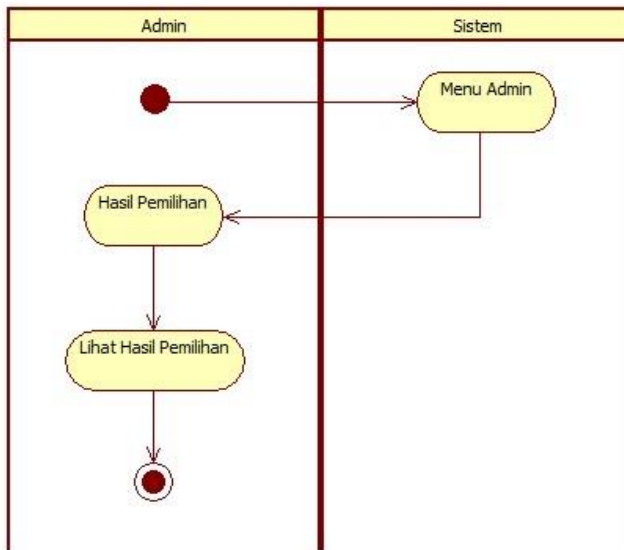


Gambar 5.25 Activity Diagram Kelola Data Pemilihan Peserta

Penjelasan :

Pada *activity diagram* ini menjelaskan proses mengelola data pemilihan peserta. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini, masuk ke menu kelola data pemilihan peserta, data pemilihan peserta ini dapat melakukan beberapa aksi yaitu tambah penilaian, ubah data, hitung data dan hapus data pemilihan peserta, kemudian simpan jika data tidak dapat tersimpan maka sistem akan melakukan validasi kepada *database* jika ya maka data kegiatan telah selesai.

5.1.2.4.6.6 Activity Diagram Admin Hasil Pemilihan Peserta

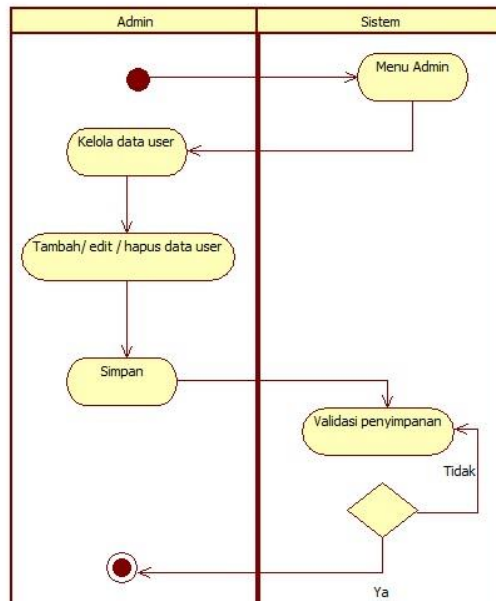


Gambar 5.26 Activity Diagram Admin Hasil Pemilihan Peserta

Penjelasan :

Pada *activity diagram* ini menjelaskan proses melihat data hasil pemilihan peserta. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini, masuk ke menu hasil pemilihan peserta, data pemilihan peserta ini dapat melakukan aksi yaitu melihat hasil data pemilihan peserta.

5.1.2.4.6.7 Activity Diagram Kelola Data User

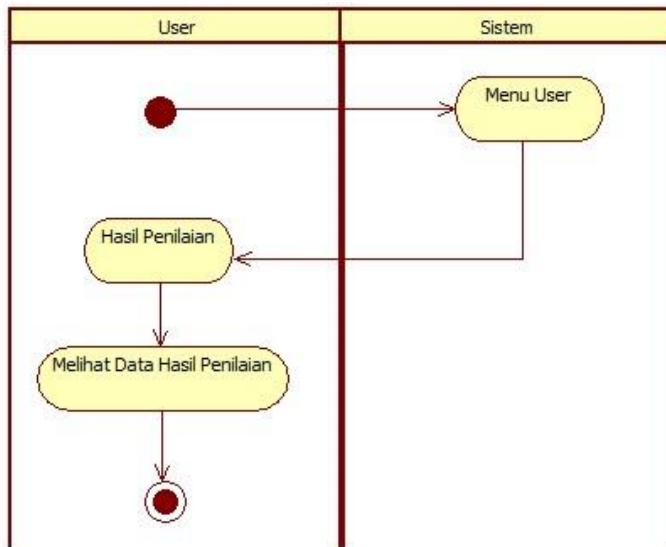


Gambar 5.27 Activity Diagram Kelola Data User

Penjelasan :

Pada *activity diagram* ini menjelaskan proses pengelolaan data *user*. *Admin* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama admin. Untuk memulai kelola ini, masuk ke menu *user*, kelola data *user* ini dapat melakukan aksi menambah, mengubah, dan menghapus data *user*.

5.1.2.4.6.7 Activity Diagram User Hasil Pemilihan Peserta



Gambar 5.28 Activity Diagram User Hasil Pemilihan Peserta

Penjelasan :

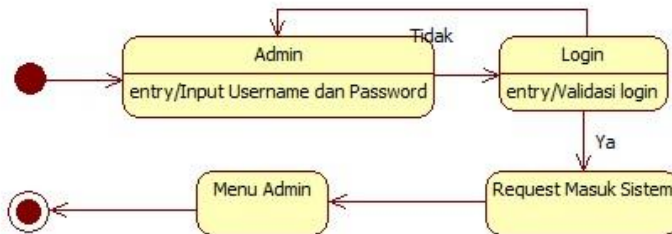
Pada *activity diagram* ini menjelaskan proses mengelola data pemilihan peserta. *User* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama *user*. Untuk memulai kelola ini, masuk ke menu kelola data pemilihan peserta, data pemilihan peserta ini dapat melakukan beberapa aksi yaitu tambah penilaian, ubah data, hitung data dan hapus data pemilihan

peserta, kemudian simpan jika data tidak dapat tersimpan maka sistem akan melakukan validasi kepada *database* jika ya maka data kegiatan telah selesai.

5.1.2.4.7 Statechart Diagram

Statechart Diagram menggambarkan transisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimulan yang diterima *statechart diagram* mendeskripsikan bagaimana suatu objek mengalami perubahan status adanya *trigger* dari *event-event*. Menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek.

5.1.2.4.7.1 Statechart Diagram Login Admin

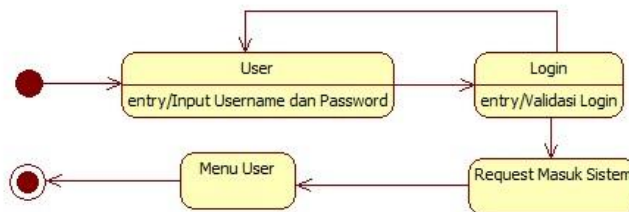


Gambar 5.29 Statechart Diagram Login Admin

Penjelasan :

Admin masuk ke menu *login* menginputkan *username* dan *password* jika *valid* maka akan masuk ke menu admin.

5.1.2.4.7.2 Statechart Diagram Login User

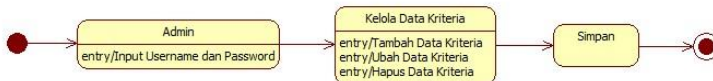


Gambar 5.30 Statechart Diagram Login User

Penjelasan :

Admin masuk ke menu *login* menginputkan *username* dan *password* jika *valid* maka akan masuk ke menu *user*.

5.1.2.4.7.3 Statechart Diagram Kelola Data Kriteria

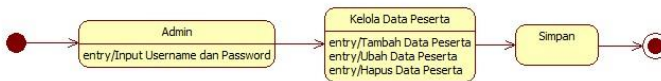


Gambar 5.31 Statechart Diagram Kelola Data Kriteria

Penjelasan :

Admin mengelola data kegiatan yang dengan perintah tambah, ubah dan hapus setelah disimpan maka *database* akan validasi *query* berhasil.

5.1.2.4.7.4 Statechart Diagram Kelola Data Peserta

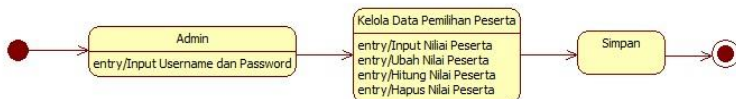


Gambar 5.32 Statechart Diagram Kelola Data Peserta

Penjelasan :

Admin mengelola data peserta yang dengan perintah tambah, ubah dan hapus setelah disimpan maka *database* akan validasi *query* berhasil.

5.1.2.4.7.5 Statechart Diagram Kelola Data Pemilihan Peserta

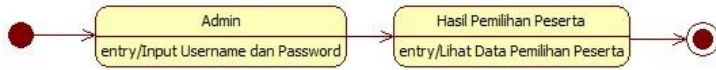


Gambar 5.33 Statechart Diagram Kelola Data Pemilihan Peserta

Penjelasan :

Admin mengelola data pemilihan peserta yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.1.2.4.7.6 Statechart Diagram Admin Hasil Pemilihan Peserta

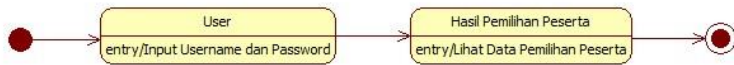


Gambar 5. 34 Statechart Diagram Admin Hasil Pemilihan Peserta

Penjelasan :

Admin melihat data hasil pemilihan peserta yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.1.2.4.7.7 Statechart Diagram User Hasil Pemilihan Peserta

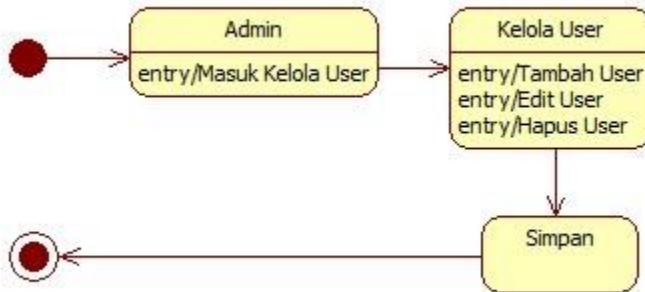


Gambar 5.35 Statechart Diagram User Hasil Pemilihan Peserta

Penjelasan :

User mengelola data kegiatan yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.1.2.4.7.8 Statechart Diagram Kelola Data User



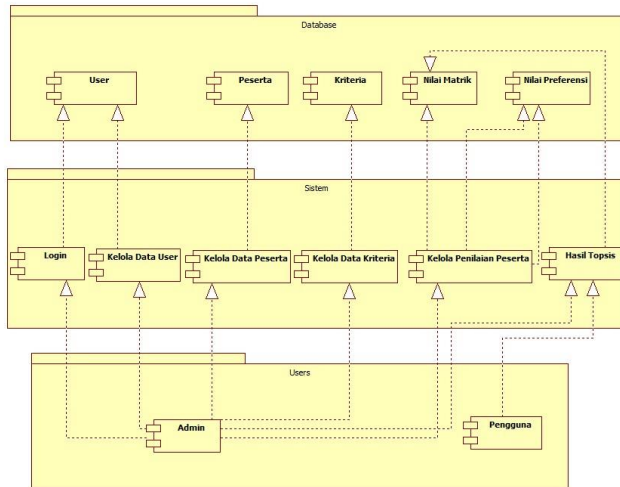
Gambar 5.36 Statechart Diagram Kelola Data User

Penjelasan :

Admin mengelola data *user* dengan aksi yang dipilih sehingga nanti di validasi ke *database* jika benar maka *query* berhasil.

5.1.2.4.8 Component Diagram

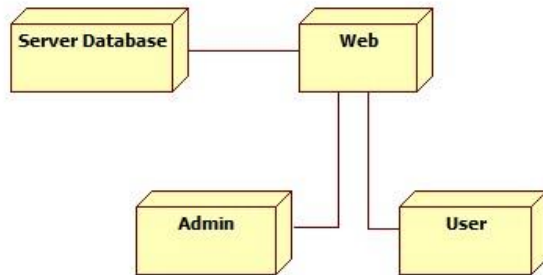
Component diagram yaitu salah satu jenis *diagram* pada *UML* yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa *file* data atau *.exe*, *source code*, *table*, dokumen, dll.



Gambar 5.37 Component Diagram

5.1.2.4.9 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal seperti sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi



Gambar 5.38 Deployment Diagram

5.2 Perancangan *User Interface*

5.2.1 *User Interface Login*

ACCOUNT LOGIN

Username	Password
SIGN IN	

KEMBALI

Gambar 5.39 User Interface Login

Keterangan Gambar :

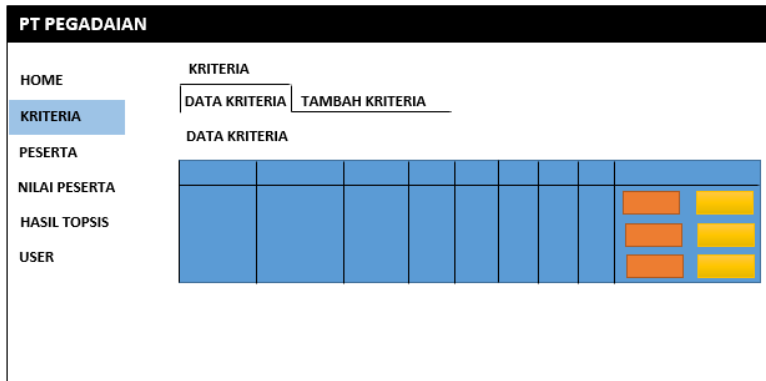
Textbox 1 : *Input Username*

Textbox 2 : *Input Password*

Command Button 1 : *Sign In*

Command Button 1 : Kembali

5.2.2 User Interface Kelola Data Kriteria



Gambar 5.40 User Interface Data Kegiatan

Keterangan Gambar :

Label 1 : Kegiatan

Label 2 : Data Kegiatan

Label 3 : Tambah Kegiatan

Table 1 : Tabel Data Kegiatan

5.2.3 User Interface Tambah Data Kegiatan

PT PEGADAIAN

HOME
KRITERIA
 PESERTA
 NILAI PESERTA
 HASIL TOPSIS
 USER

KRITERIA
 DATA KRITERIA
 TAMBAH KRITERIA

TAMBAH KRITERIA

KODE

KRITERIA

JUDUL KEGIATAN

BOBOT

POINT 1

POINT 2

POINT 3

POINT 4

POINT 5

SIFAT KRITERIA

Gambar 5.41 User Interface Tambah Data Kriteria

Keterangan Gambar :

Label 1 : Kriteria

Label 2 : Data Kriteria

Label 3 : Tambah Kriteria

- Textbox 1* : Tambah Data Kriteria
- Textbox 2* : Tambah Data Kriteria
- Textbox 3* : Tambah Data Kriteria
- Textbox 4* : Tambah Data Kriteria
- Textbox 5* : Tambah Data Kriteria
- Button 1* : Simpan

5.2.4 User Interface Data Peserta

PT PEGADAIAN							
HOME	PESERTA						
KRITERIA	DATA PESERTA TAMBAH PESERTA						
PESERTA	DATA PESERTA						
NILAI PESERTA							
HASIL TOPSIS							
USER							

Gambar 5.42 User Interface Data Peserta

Keterangan Gambar :

- Label 1* : Peserta
- Label 2* : Data Peserta
- Label 3* : Tambah Peserta
- Tabel 1* : Tabel Data Peserta

5.2.5 User Interface Tambah Data Peserta

Gambar 5. 43 User Interface Tambah Data Peserta

Keterangan Gambar :

- Label 1* : Peserta
- Label 2* : Data Peserta
- Label 3* : Tambah Peserta
- Textbox* : Data Peserta
- Button 1* : Simpan

5.2.6 User Interface Data Penilaian

PT PEGADAIAN

HOME
KRITERIA
 PESERTA
 NILAI PESERTA
 HASIL TOPSIS
 USER

NILAI PESERTA

ISI NILAI PESERTA

TAMBAH NILAI PESERTA

PESERTA

GRADE ☐ ☐ ☐ ☐ ☐

SERTIFIKASI ☐ ☐ ☐ ☐ ☐

KEIKUTSERTAAN ☐ ☐ ☐ ☐ ☐

TEMA ☐ ☐ ☐ ☐ ☐

KEDISIPLINAN ☐ ☐ ☐ ☐ ☐

Gambar 5.44 User Interface Data Penilaian

Keterangan Gambar :

- Label 1* : Penilaian
- Label 2* : Data Nilai
- Alternatif Box 1* : ID Peserta
- Textbox* : Data Peserta
- Radio Button* : Angka Nilai
- Button 1* : Simpan

5.2.7 User Interface Hasil TOPSIS Penilaian

PT PEGADAIAN						
HOME	HASIL TOPSIS					
KRITERIA	NILAI PESERTA	NILAI TERNORMALISASI NILAI BOBOT TERNORMALISASI				
PESERTA	MATRIKS IDEAL POSITIF/NEGATIF JARAK SOLUSI IDEAL POSITIF/NEGATIF NILAI PREFERENSI					
NILAI PESERTA	NILAI PESERTA					
HASIL TOPSIS						
USER						

Gambar 5.45 User Interface Hasil Topsis Penilaian

Keterangan Gambar :

Label 1 : Penilaian

Tabel 1 : Tabel Penilaian

5.2.8 User Interface Hasil Data Penilaian Ternormalisasi

PT PEGADAIAN

HOME

KRITERIA

PESERTA

NILAI PESERTA

HASIL TOPSIS

USER

HASIL TOPSIS

NILAI PESERTA

MATRIKS IDEAL POSITIF/NEGATIF

JARAK SOLUSI IDEAL POSITIF/NEGATIF

NILAI PREFERENSI

NILAI TERNORMALISASI

NILAI TERNORMALISASI

NILAI BOBOT TERNORMALISASI

Gambar 5.46 User Interface Hasil Data Penilaian Ternormalisasi

Keterangan Gambar :

Label 1 : Data Penilaian Ternormalisasi

Tabel 1 : Tabel Penilaian Ternormalisasi

5.2.9 User Interface Hasil Data Penilaian Bobot Ternormalisasi

PT PEGADAIAN						
HOME	HASIL TOPSIS					
KRITERIA	NILAI PESERTA	NILAI TERNORMALISASI	NILAI BOBOT TERNORMALISASI			
PESERTA	Matriks Ideal Positif/Negatif Jarak Solusi Ideal Positif/Negatif Nilai Preferensi					
NILAI PESERTA	NILAI BOBOT TERNORMALISASI					
HASIL TOPSIS						
USER						

Gambar 5.47 User Interface Hasil Data Penilaian Bobot Ternormalisasi

Keterangan Gambar :

Label 1 : Data Penilaian Bobot Ternormalisasi

Tabel 1 : Tabel Penilaian Bobot Ternormalisasi

5.2.10 User Interface Nilai Ideal Positif/Negatif

PT PEGADAIAN			
HOME	HASIL TOPSIS		
KRITERIA	NILAI PESERTA	NILAI TERNORMALISASI	NILAI BOBOT TERNORMALISASI
PESERTA	MATRIKS IDEAL POSITIF/NEGATIF	JARAK SOLUSI IDEAL POSITIF/NEGATIF	NILAI PREFERENSI
NILAI PESERTA	MATRIKS IDEAL POSITIF A ⁺		
HASIL TOPSIS			
USER			
	MATRIKS IDEAL NEGATIF A ⁻		

Gambar 5.48 User Interface Nilai Ideal Positif/Negatif

Keterangan Gambar :

Label 1 : Nilai Ideal Positif/Negatif

Label 2 : Nilai Ideal Positif A⁺

Label 1 : Nilai Ideal Negatif A⁻

Tabel 1 : Tabel Penilaian Ternormalisasi

Tabel 2 : Tabel Penilaian Ternormalisasi

5.2.11 User Interface Jarak Solusi Ideal Positif/Negatif

PT PEGADAIAN			
HOME	HASIL TOPSIS		
KRITERIA	NILAI PESERTA	NILAI TERNORMALISASI	NILAI BOBOT TERNORMALISASI
PESERTA	MATRIKS IDEAL POSITIF/NEGATIF	JARAK SOLUSI IDEAL POSITIF/NEGATIF	NILAI PREFERENSI
NILAI PESERTA	JARAK SOLUSI IDEAL POSITIF D ⁺		
HASIL TOPSIS			
USER			
	JARAK SOLUSI IDEAL NEGATIF D ⁻		

*Gambar 5.29 User Interface Jarak Solusi Ideal
Positif/Negatif*

Keterangan Gambar :

- Label 1* : Jarak Solusi Ideal Positif/Negatif
- Label 2* : Jarak Solusi Ideal Positif D⁺
- Label 1* : Jarak Solusi Ideal Negatif D⁻
- Tabel 1* : Tabel Penilaian Ternormalisasi
- Tabel 2* : Tabel Penilaian Ternormalisasi

5.2.12 User Interface Nilai Preferensi

PT PEGADAIAN						
HOME	HASIL TOPSIS					
KRITERIA	NILAI PESERTA	NILAI TERNORMALISASI	NILAI BOBOT TERNORMALISASI			
PESERTA	Matriks Ideal Positif/Negatif		Jarak Solusi Ideal Positif/Negatif		NILAI PREFERENSI	
NILAI PESERTA	NILAI PREFERENSI					
HASIL TOPSIS						
USER						

Gambar 5.49 User Interface Nilai Preferensi

Keterangan Gambar :

Label 1 : Nilai Preferensi

Tabel 1 : Tabel Nilai Preferensi

5.2.13 User Interface Kelola Data User

PT PEGADAIAN					
HOME	USER				
KRITERIA	DATA USER	TAMBAH USER			
PESERTA	DATA USER				
NILAI PESERTA					
HASIL TOPSIS					
USER					

Gambar 5.50 User Interface Kelola Data User

Keterangan Gambar :

Label 1 : User

Label 2 : Data User

Label 3 : Tambah User

Tabel 1 : Tabel Data User

5.2.14 User Interface Tambah Data User

Gambar 5.51 User Interface Tambah Data User

Keterangan Gambar :

Label 1 : User

Label 2 : Data User

Label 3 : Tambah User

Tabel 1 : Tabel Tambah User

5.3 Perancangan Arsitektur Sistem

5.3.1 Perancangan Arsitektur Perangkat Lunak Sistem

Perangkat Lunak yang digunakan dalam Sistem Implementasi Pemilihan Karyawan Pada Penjadwalan Akademik di PT Pegadaian (Persero) adalah sebagai berikut:

Tabel 5.6 Arsitektur Perangkat Lunak Sistem

No.	Tools/Software	Fungsi	Keterangan
1	<i>Windows 8 (min)</i>	Sistem Operasi	-
2	<i>Bizagi Modeller</i>	Pembuatan Flow Map	-
3	<i>Google Chrome</i>	Aplikasi web browser	-
4	<i>Star UML</i>	Pembuatan Diagram UML	-
5	<i>MySQL</i>	DBMS	-

6	<i>Microsoft Word</i>	Document	-
7	<i>Visual Studio Code</i>	Text Editor	-

5.3.2 Perancangan Arsitektur Perangkat Keras Sistem

Perangkat keras yang digunakan dalam Sistem Implementasi Pemilihan Peserta Pada Penjadwalan Kegiatan Akademik di PT Pegadaian ini adalah sebagai berikut :

Tabel 5.7 Arsitektur Perangkat Keras Sistem.

BAB 1

1.1 Metode Topsis

Pertama kalinya oleh Yoon dan Hwang dan dinilai oleh surveyor dan operator yang berbeda. Metode Topsis standar dirancang untuk menemukan alternatif optimum yang memiliki kedekatan minimum dengan penyelesaian ideal positif dan kedekatan maksimum terhadap penyelesaian ideal negatif. Penyelesaian ideal positif berupaya untuk memaksimalkan kriteria manfaat dan

meminimumkan kriteria biaya, sedangkan penyelesaian ideal negatif justru sebaliknya. Secara umum prosedur TOPSIS mengikuti terdiri dari beberapa langkah dimulai dari membuat matriks keputusan hingga menentukan nilai preferensi untuk pengambilan keputusan yang ideal.

Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) merupakan salah satu metode dalam pengambilan keputusan, yang mana dalam menghasilkan sebuah keputusan akan memilih alternatif yang tidak hanya paling mendekati solusi ideal positif, akan tetapi juga paling jauh dari solusi ideal negatif, maka langkah-langkah yang dilakukan dalam metode TOPSIS yaitu :

- a. Membangun matriks keputusan ternormalisasi
- b. Membangun matriks bobot ternormalisasi
- c. Menentukan solusi ideal positif dan negatif.
- d. Menghitung jarak setiap *alternative* keputusan dari solusi ideal positif dan negatif.
- e. Menentukan nilai preferensi untuk setiap alternative.

TOPSIS didasarkan pada konsep alternatif terpilih yang terbaik yang banyak digunakan pada beberapa model. Multi Attribute Decision Making (MADM) untuk

menyelesaikan masalah keputusan secara praktis. Metode ini banyak diimplementasikan ke berbagai bidang. Pada MADM klasik metode, seperti teknik untuk preferensi pesanan dengan kemiripan dengan solusi ideal (TOPSIS) yang dikembangkan oleh Hwang dan Yoon, Promothée, Viktor, Electre, bobot masing-masing atribut dan peringkat alternatif disajikan dengan angka tajam. Namun, di dunia nyata, pembuat keputusan mungkin lebih suka mengevaluasi atribut dengan menggunakan variabel linguistik (variabel yang bernilai kata/kalimat, bukan angka. Sebagai alasan menggunakan kata/kalimat dari pada angka karena peranan linguistik kurang spesifik dibandingkan angka namun informasi yang disampingkan lebih informatif) daripada nilai eksak karena pengetahuan parsial tentang atribut dan kurangnya kemampuan pemrosesan informasi dari domain masalah. Di situasi seperti itu, informasi preferensi alternatif disediakan oleh pembuat keputusan mungkin tidak jelas, tidak tepat, atau tidak lengkap.

TOPSIS adalah metode beberapa kriteria untuk mengidentifikasi solusi dari satu set alternatif terbatas.

Metode TOPSIS adalah teknik untuk urutan preferensi oleh kesamaan untuk solusi ideal. Solusi ideal (juga disebut solusi ideal positif) merupakan solusi yang dapat memaksimalkan kriteria / atribut manfaat dan meminimalkan kriteria/ atribut biaya, sedangkan solusi ideal negatif (juga disebut solusi anti-ideal) memaksimalkan kriteria/ atribut biaya dan meminimalkan kriteria/ atribut manfaat. Alternatif terbaik adalah salah satu yang paling dekat dengan solusi ideal positif dan terjauh dari solusi ideal negatif. TOPSIS adalah salah satu metode pengambilan keputusan multikriteria atau alternatif pilihan yang merupakan alternatif yang mempunyai jarak terkecil dari solusi ideal positif dan jarak terbesar dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean* (jarak garis lurus "biasa" antara dua titik dalam ruang Euclidean). Namun, alternatif yang mempunyai jarak terkecil dari solusi ideal positif, tidak harus mempunyai jarak terbesar dari solusi ideal negatif. Maka dari itu, TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif secara bersamaan. Solusi optimal dalam metode TOPSIS

didapat dengan menentukan kedekatan relatif suatu alternatif terhadap solusi ideal positif. TOPSIS akan merangking alternatif berdasarkan prioritas nilai kedekatan relatif suatu alternative terhadap solusi ideal positif. Alternatif-alternatif yang telah dirangking kemudian dijadikan sebagai referensi bagi pengambil keputusan untuk memilih solusi terbaik yang diinginkan. Metode ini merupakan metode kompromi yang banyak digunakan untuk menangani berbagai kriteria masalah pengambilan keputusan. Dasar pemikiran metode TOPSIS adalah bahwa alternatif yang dipilih harus memiliki jarak terdekat solusi positif ideal dan jarak terjauh dari solusi *negative* ideal. Bobot kriteria dalam metode TOPSIS diberikan apriori. Selain itu, solusi positif-ideal dan solusi *negative* ideal langsung dihasilkan dari matriks keputusan. Dalam Metode TOPSIS klasik, semua data keputusan diketahui secara tepat atau diberikan sebagai nilai yang tajam. Namun, ini bisa sulit untuk ditentukan data justru karena penilaian manusia sering tidak jelas dalam banyak kondisi dalam praktiknya

TOPSIS membutuhkan rating kinerja setiap alternatif A_i pada setiap kriteria C_j yang ternormalisasi yaitu :

$$r_{ij} = \frac{x^{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Keterangan = $i=1,2,\dots,m$ dan $j=1,2,\dots,n$.

Sebelum mencari nilai r_{ij} , maka cari mencari nilai dengan rumus seperti ini :

$$|X_n| = \sqrt{\sum_{i=1}^m x_{ij}^2}$$

Menghitung nilai matriks yang ternormalisasi yang terbobot (Y), untuk bobot yang sudah ditentukan (W), dengan rumus :

$$y_{ij} = w_i * r_{ij}$$

Solusi Ideal Positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan rating bobot ternormalisasi (y_{ij}) sebagai :

$$A^+ = \max(y_{1+}, y_{2+}, \dots, y_{n+})$$

$$A^- = \max(y_{1-}, y_{2-}, \dots, y_{n-})$$

Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan sebagai :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}$$

Jarak antara alternatif A_i dengan solusi ideal negatif dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}$$

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

Keterangan : $i=1,2,\dots,n$.

Nilai V_i yang lebih besar menunjukkan bahwa Alternatif A_i lebih dipilih

1.1.1 Kelebihan dan Kekurangan Metode TOPSIS

1. Kelebihan Metode TOPSIS
 - a. Konsepnya sederhana dan mudah dipahami
 - b. Komputasinya efisien
 - c. Dapat Menentukan nilai preferensi untuk setiap alternative

- d. Memiliki kemampuan untuk mengukur kinerja relative dari alternative-alternatif keputusan dalam bentuk matematis yang sederhana

2. Kekurangan Metode TOPSIS

Harus adanya bobot yang dihitung untuk melanjutkan hitungan data selanjutnya dengan metode TOPSIS.

1.2 Tujuan Metode

Pada dasarnya, proses MADM (*Multi Attribute Decision Making*) dilakukan melalui 3 tahap, yaitu:

1. Penyusunan komponen-komponen situasi.
2. Analisis.
3. Sintesis informasi.

Pada tahap penyusunan komponen, komponen situasi, akan dibentuk tabel taksiran yang berisi identifikasi alternatif dan spesifikasi tujuan, kriteria dan atribut.

MADM (*Multi Attribute Decision Making*) adalah salah satu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria-kriteria tertentu. Inti dari *Multiple Attribute Decision Making* (MADM) adalah menentukan nilai bobot untuk setiap

atribut/kriteria, yang kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Pada dasarnya, ada 3 (tiga) pendekatan untuk mencari nilai bobot atribut, yaitu :

1. Pendekatan subyektif
2. Pendekatan obyektif
3. Pendekatan integrasi antara subyektif & obyektif.

Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan .

1.3 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam pengumpulan data adalah :

1. Data primer adalah data yang diambil langsung dari obyek penelitian atau merupakan data yang berasal dari sumber asli atau pertama. Teknik

pengumpulan data primer dilakukan melalui teknik observasi dengan cara mengumpulkan informasi informasi langsung ke lokasi penelitian untuk mengamati bagaimana penentuan untuk pemilihan karyawan yang dilakukan oleh pihak admin dalam kegiatan akademik tersebut.

2. Data sekunder adalah data yang tidak didapatkan secara langsung dari objek penelitian, melainkan data yang berasal dari sumber yang telah dikumpulkan oleh pihak lain. Teknik pengumpulan data sekunder dilakukan dengan cara studi dokumentasi dan studi literature.

Tujuan dan Solusi Pemecahan Masalah Pada Sistem Pendukung Keputusan

a. Tujuan Sistem Pendukung Keputusan (SPK)

Tujuan dari SPK (Sistem Pendukung Keputusan) yang dikemukakan oleh Peter G.W Keen dan Scoot Morton, yaitu :

- a) Mendukung penilaian manajer bukan mencoba ataupun menggantinya

- b) Membantu manajer dalam membuat keputusan untuk memecahkan masalah semi terstruktur.
- c) Meningkatkan efektifitas pengambilan keputusan manajer dari pada efisiensinya.
- b. Komponen-komponen Sistem Pendukung Keputusan (SPK), terdiri :
 - a) Data Management. Termasuk database yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh software yang disebut dengan Databse Management System (DBMS).
 - b) Model Management. Model ini melibatkan model finansial, statistical, management science, atau berbagai model kuantitatif lainnya, sehingga memberikan suatu kemampuan analitis, dan manajemen software yang diperlukan.
 - c) Communication (dialog subsystem). User dapat berkomunikasi dan memberikan perintah DSS melalui sub sistem.
 - d) Knowledge Management. Subsistem optional ini mendukung sub sistem lain atau bertindak sebagai komponen yang berdiri sendiri.

1.4 Sejarah PHP (Hypertext Preprocessor)

Pertama kali PHP dikenal dengan kependekan dari (*Personal Home Page*) atau Situs Personal yang dibuat oleh Rasmus Lerdorf pada Tahun 1995. Saat pertama kali dibuat PHP masih bernama *Form Interpreted* (FI) yang berupa sekumpulan skrip untuk pengolahan data formulir dari web.



Logo PHP Pertama

Kemudian Rasmus merilis kode sumber tersebut pada bulan November tahun 1997 untuk umum dengan memberikan nama PHP/FI. Dengan merilis kode tersebut menjadi kode sumber terbuka sehingga banyak

pemrograman yang tertarik untuk mengembangkan PHP tersebut.

Di tahun yang sama, sebuah perusahaan bernama Zend menulis kembali interpreter PHP menjadi lebih cepat, rapih dan lebih baik. Sehingga pada bulan Juni tahun 1998, perusahaan tersebut merilis kembali interpreter baru untuk PHP dan meresmikan perilisan tersebut dengan nama PHP 3.0 serta mengubah nama menjadi akronim berulang yang diberi nama PHP (*Hypertext Preprocessing*). Pada pertengahan tahun 1999, perusahaan Zend merilis kembali interpreter PHP dengan nama PHP 4.0 pada versi 4.0 ini banyak digunakan pada awal abad ke-21. Karena versi ini memiliki kemampuan untuk membangun aplikasi web kompleks tetapi tetap memiliki stabilitas dan kecepatan yang tinggi. Setelah 5 tahun berlalu pada akhirnya Zend merilis kembali PHP dengan versi 5.0 pada bulan Juni tahun 2004. Pada versi ini, PHP memasukkan model pemrograman berorientasi objek untuk mengembangkan bahasa pemograman kearah paradigme berorientasi objek.



Gambar logo PHP 5

Di tanggal 18 Desember pada tahun 2014 Zend merilis bahasa pemrograman PHP menjadi versi 5.6.4 yang diperuntukan agar mempermudah pengembang menjalankan kode PHP tanpa menginstall *software server* terlebih dahulu. Tidak lama kemudian pada tanggal 17 Februari 2017 versi 5.6.4 di kembangkan kembali menjadi versi terbaru dan stabil pada bahasa pemrograman PHP saat ini menjadi versi 7.0.16 dan 7.1.2.



Gambar Logo PHP 7

1.5 Mengenal PHP

PHP merupakan bahasa pemrograman scripting karena bahasa pemrograman PHP diproses pada web server, sehingga semua kode-kode yang ditulis menggunakan PHP akan dijalankan dan disimpan di web server maka PHP menyatu dengan HTML (Kode dasar website) dan dijalankan pada server side. Artinya, semua sintaks PHP yang diberikan akan sepenuhnya dijalankan

pada server, sedangkan yang dikirimkan ke browser hanya hasilnya saja.

PHP digunakan untuk membuat website dinamis dan interaktif. Dinamis yaitu website tersebut dapat berubah-ubah tampilan dan kontennya sesuai kondisi tertentu.

Terdapat dua acara yang dapat dilakukan untuk menulis kode-kode PHP untuk membangun sebuah aplikasi dinamis, seperti menggunakan PHP native dan framework.

a. PHP Native

PHP Native merupakan metode penulisan kode-kode bahasa pemrograman PHP yang benar-benar dimulai dari awal atau dari nol.

Pada PHP native ini programmer akan membuat kerangka kerjanya sendiri untuk mengembangkan website dinamis. Biasanya pemula akan disarankan untuk mempelajari PHP Native.

b. Framework PHP

Untuk memudahkan pengembangan pada aplikasi web dengan menggunakan bahasa pemrograman PHP maka gunakanlah framework PHP, karena terdapat beberapa

keuntungan yang bisa didapatkan ketika membuat website menggunakan framework, diantaranya :

- 1) Tidak perl melakukan pengulangan coding
- 2) Aplikasi web yang dibangun dengan framework akan lebih stabil
- 3) Mengembangkan aplikasi web dapat lebih cepat
- 4) Membantu pemula dalam membuat aplikasi

Adapun beberapa framework PHP yang sering didengar oleh para developer web, yaitu :

- 1) CodeIgniter



Gambar Logo CodeIgniter

- 2) Laravel



Gambar Logo Laravel

3) Symfony



Gambar Logo Symfony

4) Zend Framework



Gambar logo Zend Framework

5) CakePHP



Gambar Logo Cake PHP

6) Seagull



Gambar Logo Framework Seagull

PHP dimulai dengan tulisan antara tag :

- a. `<? Dan ?>`
- b. `<?php dan ?>`
- c. `<script language="php">` dan `</script>`
- d. `<% dan %>`
- e. Setiap *statement* atau perintah yang digunakan biasanya diakhiri dengan titik-koma (;)
- f. *Case Sensitive* untuk nama identifier yang dibuat user yaitu berupa variable, konstanta, fungsi, dan lain-lain. Namun tidak *Case Sensitive* untuk identifier built-in dari PHP. Maka :

- a) \$nama ≠ \$Nama ≠ NAMA
- b) hitungLuas() ≠ HitungLuas()
- c) echo = ECHO
- d) while = WHILE

g. Mudah dipahami

1.5.1 Variabel

Variabel merupakan suatu lokasi penyimpanan yang berisikan nilai atau informasi yang nilainya tidak diketahui maupun telah diketahui.

Nilai dari variabel dapat di isi dengan informasi yang diinginkan dan dapat diubah nilainya pada saat kode program sedang berjalan. Sebuah variabel memiliki nama yang digunakan untuk mengakses nilai dari variabel itu. Jika anda memiliki pengetahuan dasar tentang bahasa pemrograman, tentunya tidak asing dengan istilah variabel.

- a. Variabel digunakan untuk menyimpan sebuah value, informasi atau data
- b. Penamaan variable diawali dengan tanda \$
- c. Panjang yang digunakan tidak ada batas maksimum penggunaan

- d. Setelah menggunakan tanda \$ diawali oleh huruf atau *under-score* (_). Maka karakter yang berikutnya bias terdiri dari angka, huruf dan karakter tertentu yang diperbolehkan.
- e. Case yang digunakan bersifat case-sensitive PHP membedakan variabel yang ditulis dengan huruf besar dan kecil (bersifat case sensitive), contoh \$nilai_kuadrat tidak sama dengan \$NilaiKuadrat dan \$NILAIKUADRAT ketiganya akan dianggap sebagai variabel yang berbeda.
- f. Tidak boleh mengandung spasi
- g. Tidak perlu di deklarasikan.

1.5.2 Tipe Data

Pada PHP tipe data variable secara otomatis ditentukan oleh interpreter PHP. Tetapi, PHP mendukung 8 buah tipe data primitive, seperti :

- a. Integer

Integer merupakan semua angka bulat baik itu positif ataupun negative bukan pecahan, seperti : 1,2,3,5,dst

- b. Float

Float atau floating point atau yang disebut dengan nomor pecahan atau juga bilangan real, seperti : 0.1,0.25,0.2, 0.314E1,dst.

c. Boolean

Boolean merupakan tipe data standar, Boolean hanya menyatakan kebenaran dengan kata kunci perintah TRUE (benar) atau FALSE (salah).

d. String

String merupakan rangkaian karakter, sehingga karakter tersebut sama seperti byte. String dapat didefinisikan dengan cara yang paling mudah yaitu menggunakan tanda satu petik (‘’) terdapat 265 karakter yang dapat dijadikan string.

e. Array

Array merupakan jenis tipe data yang unik, berbeda dengan tipe data lainnya, tipe data array dapat menyimpan satu atau lebih data dalam variabel tunggal. Tipe data array berguna untuk menyimpan banyak data dalam satu variabel.

f. Resource

Resource merupakan jenis tipe data yang menyimpan sebuah referensi external resource seperti bagian koneksi database.

g. Object

Object merupakan keluaran dari class. Object dapat menampilkan atau mengelola isi class.

h. Null

Null mendeklarasikan variabel tanpa value. Sebuah variabel dapat menjadi null jika :

- a) Variabel belum diset ke nilai apapun
- b) Menerima unset()
- c) Dialokasikan konstanta NULL

1.5.3 Konstanta

Dalam bahasa pemrograman, Konstanta atau *constant* merupakan suatu lokasi penyimpanan yang berisikan nilai yang memiliki sifat tetap dan tidak dapat berubah sepanjang program berjalan.

Konstanta memiliki fungsi yang sama seperti variabel namun nilainya statis/konstan dan tidak bisa berubah.

1.5.3.1 Aturan Penulisan Konstanta PHP

1. Cara Pendefinisikan Konstanta dalam PHP

Jika variabel pada PHP dibuat dengan menambahkan “tanda dollar”, seperti : \$nilai_kuadrat. Untuk membuat konstanta PHP memiliki 2 cara :

- a. Menggunakan kata kunci (*Keyword*) `const`
- b. Menggunakan fungsi `define`
2. Konstanta PHP bersifat case sensitive

Sama seperti variabel, konstanta dalam PHP bersifat case sensitif, sehingga perbedaan huruf besar dan kecil dianggap berbeda. GAJI, GaJi, dan gaji merupakan 3 konstanta yang berbeda.

3. Nilai konstanta PHP tidak dapat diubah
4. Konstanta hanya dapat berisi tipe data tertentu

Konstanta dalam PHP hanya dapat berisi tipe data sederhana, yaitu :

- a. Boolean
- b. Integer
- c. Float
- d. String

Hal ini berbeda dengan variabel yang dapat berisi tipe data turunan seperti array, objek atau resources.

5. Konstanta Sistem PHP (Predefined Constant)

Sama seperti variabel, PHP juga telah membuat beberapa konstanta yang telah didefinisikan dan tidak bisa di ubah nilainya. Namun karena banyaknya modul yang dapat ditambahkan kedalam PHP, Predefined Constant dalam PHP akan bertambah tergantung modul yang ada. Namun sebagai contoh, berikut adalah Predefined Constant dalam sistem inti PHP:

PHP_VERSION,PHP_MAJOR_VERSION,PHP_MINOR_VERSION,PHP_RELEASE_VERSION,PHP_VERSION_ID,PHP_EXTRA_VERSION,PHP_ZTS,PHP_DEBUG,PHP_MAXPATHLEN, PHP_OS, PHP_SAPI, PHP_EOL,
 PHP_INT_MAX,PHP_INT_SIZE,DEFAULT_INCLUDE_PATH,PEAR_INSTALL_DIR,PEAR_EXTENSION_DIR,PHP_EXTENSION_DIR,PHP_PREFIX,PHP_BINDIR,PHP_BINARY, PHP_MANDIR, PHP_LIBDIR, PHP_DATADIR, __LINE__ , __FILE__ , __DIR__ ,
 __FUNCTION__ , __CLASS__ , __TRAIT__ ,
 __METHOD__, __NAMESPACE__,

1.5.4 Mengenal Operator pada PHP

Operator merupakan simbol yang berfungsi untuk melakukan aksi/operasi tertentu pada nilai operand yang

umumnya pada hasil operasi tersebut menghasilkan nilai baru. Operator terdiri dari 4 macam yaitu :

1.5.4.1 Operator Assignment

Operator ini untuk mengalokasikan nilai tertentu dengan tanda sama dengan (=). Seperti :

Operator	Contoh	Artinya
=	$X=y$	$X=y$
$+=$	$X+=y$	$X=x+y$
$-=$	$X-=y$	$X=x-y$
$*=$	$X*=y$	$X=x*y$
$/=$	$x/=y$	$X=x/y$
$.=$	$x.=y$	$X=x.y$
$\%=$	$X\%=y$	$X=x\%y$

1.5.4.2 Operator Aritmeka

Operator ini akan melakukan perhitungan matematika. Seperti:

Operator	Nama	Contoh	Hasil
+	Penambahan	$1 + 4$	5
-	Pengurangan	$1 - 4$	-3
/	Pembagian	$1 / 4$	0.25
*	Perkalian	$1 * 4$	4
%	Sisa pembagian (Modulus)	$5 \% 2$	1
++	Inkremen	$X = 5$ $x ++$	$X = 6$
--	Dekremen	$X = 5$ $X --$	$X = 4$

1.5.4.3 Operator Perbandingan

Operator perbandingan ini digunakan untuk menghasilkan 2 nilai yang hasil akhirnya merupakan nilai Boolean True atau False. Seperti :

Operator	Nama	Contoh	Hasil
----------	------	--------	-------

$=$	Sama dengan	$6 = 6$	True
$!=$	Tidak sama dengan	$3 != 3$	False
$>$	Lebih besar	$1 > 5$	False
$>=$	Lebih besar sama dengan	$3 >= 4$	False
$<$	Lebih kecil	$2 < 4$	True
$<=$	Lebih kecil sama dengan	$5 <= 4$	False

1.5.4.4 Operator Logical

Operator logical ini berfungsi untuk mengoperasikan secara logical menggunakan and, or, atau not. Seperti :

Operator	Description	Example
$\&\&$	And	$X = 5$ $y = 2$

		(x < 10 && y > 1) hasilnya true
	Or	X = 4 y = 6 (x == 5 y == 6) hasilnya true
!	Not	X = 5 y = 3 !(x == y) hasilnya true

1.5.5 Kelebihan PHP

Sebagai salahsatu bahasa pemrograman PHP memiliki banyak kelebihan diantaranya :

- a. Komunitas yang besar

Komunitas PHP sangat besar dan tidak dapat dipungkiri bahwa komunitas ini tersebar diseluruh dunia. Tidak hanya di social media seperti Facebook bahkan komunitas PHP ini berada di Telegram, WhatsApp, dll.

- b. Mudah dipelajari

PHP merupakan bahasa pemrograman semua orang yang hampir semua orang yang bergelut dengan dunia Web

Development pernah menggunakannya atau mencobanya.

c. Resources yang melimpah

Dengan banyaknya komunitas yang besar maka akan berdampak pada kemudahan mencari resources yang berhubungan dengan PHP baik itu permasalahan yang terjadi di library, software, CMS hingga framework PHP.

d. Sempel

PHP memiliki syntax yang sangat sederhana dan mudah untuk dipelajari. Untuk memulai belajar PHP tidak perlu melakukan pengaturan apapun cukup hanya install XAMPP ataupun WAMP maka dapat memulai langsung belajar PHP.

e. Maintenance mudah

Sekali web yang menggunakan PHP berjalan, programmer dapat dengan mudah melakukan update dari software PHP dengan mudah jika memang diperlukan. karena sifat PHP yang merupakan interpreter. Aplikasi web yang dibuat dengan menggunakan PHP dapat dengan

mudah diupgrade versi PHP tanpa harus melakukan kompilasi ulang source code. berbeda sekali dengan bahasa pemrograman lain yang membutuhkan kompilasi ulang jika melakukan upgrade versi dari bahasa pemrograman. PHP juga dapat berjalan pada berbagai macam web server seperti apache, nginx, dan IIS.

f. Open Source

PHP merupakan sebuah proyek Open source dengan license yang dikeluarkan oleh PHP group yaitu PHP license V3.01. Inti dari license ini adalah setiap pengguna program PHP bebas menggunakan PHP secara gratis tanpa harus memberikan royalty apapun ke PHP group namun tetap wajib mencantumkan licensi atas PHP yang dimiliki PHP Group. Dengan kata lain selama pemakai program PHP tidak mengakui produk PHP adalah buaatannya maka perjual belian program yang menggunakan PHP diperbolehkan tanpa harus membayar licensi apapun.

g. Perkembangan pesat

Karena sifat PHP yang open source, banyak sekali bermunculan proyek proyek open source besar yang menggunakan PHP seperti Prestashop, WordPress, Drupal, dan lain lain. Hal ini menjadi keunggulan yang sangat besar bagi orang yang menguasai pemrograman PHP. Dengan sangat luasnya perkembangan PHP, maka kesempatan untuk bisnis ataupun kerja pada bidang pemrograman PHP sangatlah luas

1.5.6 Kekurangan PHP

Kekurangan utama PHP yaitu PHP merupakan bahasa yang *weak type* dimana variabel tidak memiliki tipe data sehingga menyulitkan ketika melakukan debugging. Pada weak type ini menyebabkan terjadinya *juggling* dimana variable yang terjadi berisi nilai integer misalnya dapat berubah menjadi nilai string atau tipe data lainnya.

Berikut merupakan kekurangan dalam bahasa pemrograman PHP :

a. Banyak kompetisi

Komunitas yang banyak tentu membawa kompetisi yang ketat. Para web developer yang menguasai PHP tiap hari semakin bertambah. Namun kekurangan ini

seharusnya menjadi pemacu bagi para pebisnis yang menginginkan produk IT untuk menggunakan PHP sebagai bahasa pemrograman yang digunakan untuk mengembangkan aplikasi bisnisnya karena terbukanya para programmer PHP yang sangat kompetitif dan tiap hari semakin banyak

b. Mudah di bajak

Karena sifat PHP yang merupakan interpreter, source code dari aplikasi php dapat dengan mudah di modifikasi dan diubah fungsinya. hal ini membuat PHP tidak cocok untuk digunakan mengembangkan aplikasi jika pemilik aplikasi memiliki source code yang ingin dijaga kerahasiaannya.

Meskipun ada cara untuk mengamankan source code yang menggunakan bahasa pemrograman PHP, namun dibutuhkan sebuah extensi yang berbayar yang dikeluarkan oleh Zend sebuah korporasi di bidang pemrograman PHP.

c. Terkesan kurang prestigious

Entry level yang berada pada tingkat pemula, yakni mudah dipelajari oleh programmer pemula membuat bahasa pemrograman PHP terkesan kurang prestisius jika

dibandingkan dengan bahasa pemrograman web lain yang terkesan lebih sulit untuk digunakan. Aplikasi web yang dihasilkan dari penggunaan bahasa pemrograman PHP terkesan kurang aman dan memiliki celah. Namun sebenarnya hal ini disebabkan karena faktor pengembang yang mungkin belum mempelajari secara penuh bagaimana standar dan cara membuat aplikasi yang benar dengan menggunakan PHP. Dari sisi performa, pemrograman PHP dan pemrograman lainnya jika digunakan dengan standar dan penerapan yang benar, akan menghasilkan sebuah aplikasi web yang berkualitas.

d. Tidak adanya type data pada PHP

PHP tidak memiliki type data. hal ini membuat kadang pada saat menggunakan bahasa pemrograman PHP muncul bug bug yang tidak diinginkan berkaitan dengan tidak adanya standar type data pada php. contohnya adalah data “1000” dan “1e3” jika dibandingkan akan memiliki type data yang sama karena secara implisit data tersebut dirubah menjadi floating point. namun kekurangan ini berkaitan sekali dengan pengalaman dari developer yang menggunakan bahasa pemrograman PHP.

developer yang sudah ahli tentunya sudah paham betul bagaimana mengatasi permasalahan ini.

1.6 Mengenal Data

Data merupakan fakta atau observasi mental yang biasanya mengenai fenomena fisik atau transaksi bisnis. Data pun merupakan suatu ukuran objektif dari atribut (karakteristik) dari entitas.

Dalam pemrograman tempat penyimpanan data disebut juga dengan basis data.

Basis data merupakan suatu focus utama pada aplikasi, pada awal tahun 1960 Charles Bachman di perusahaan General electric mendesain generasi pertama DBMS yang disebut dengan penyimpanan data terintegrasi (Integrated Data Store) karena hampir semua data computer disimpan pada pita magnetic, namun karena pita hanya dapat di proses secara berurut, maka data harus disimpan dalam bentuk daftar (atau biasa disebut file sekuensial).

Database terdiri dari kelompok table-tabel yang berhubungan. Pada kebanyakan kasus, masing-masing table berisi data dengan satu tema. Hubungan antara table-tabel direpresentasikan dengan cara memberikan masing-

masing baris ID yang unik dan menggunakan ID tersebut untuk menghubungkan sebuah baris dalam satu table dengan sebuah baris dalam table baris kedua. Key pada database merupakan kolom yang mengidentifikasi baris yang unik. Kemudian foreign key merupakan sebuah kolom dalam satu table yang merupakan identifier unik pada table kedua. Foreign key digunakan untuk memperlihatkan hubungan diantara table. Table dapat dikombinasikan dengan menggunakan SQL yang merupakan bahasa standar industry untuk pemrosesan table.

Sebuah sistem database terdiri dari empat unsur dasar yaitu :

- a. User
- b. Aplikasi database
- c. DBMS
- d. Database

User menggunakan sistem database untuk melakukan pekerjaannya. Aplikasi hasilkan form, queries dan report, mengeksekusi logika aplikasi dan mengendalikan pemrosesan DBMS menciptakan, memproses, dan mengadministrasikan database.

Sebuah *database* adalah sekumpulan records terintegrasi yang menggambarkan dirinya sendiri. Database terdiri dari data user, metadata, indeks, prosedur tersimpan (*stored procedur*), trigger, dan metadata aplikasi database.

Trigger merupakan prosedur yang aktif saat tindakan tertentu terjadi. Teknologi *database* dapat digunakan dalam aplikasi yang luas. Sebagian *database* digunakan oleh satu individu, sementara sebagian lagi digunakan oleh *workgroup*, dan yang lainnya digunakan oleh organisasi besar.

Seperti sistem informasi, sistem database dapat dikembangkan dengan menggunakan sebuah proses yang mencakup tiga fase, yaitu :

- a. Requirement
- b. Desain
- c. Implementasi

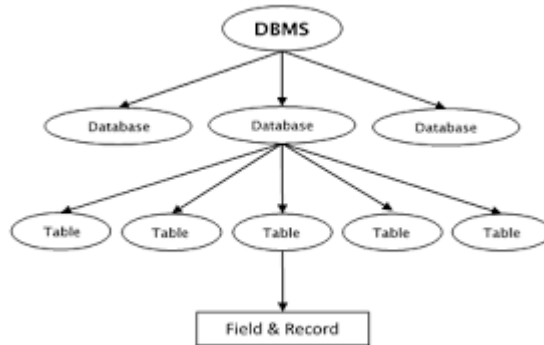
Selama fase *requirement*, sebuah model data atau representasi logis dari struktur database, dikembangkan.

Model data merupakan bagian penting karena desain database dan aplikasinya diturunkan dari model data.

Diagram entity-relationship merupakan sebuah tool yang digunakan untuk merepresentasikan model data.

Model data ditransformasikan menjadi table dan hubungan selama fase desain. Indeks, batasan, prosedur tersimpan, dan trigger juga dirancang dalam fase ini. Diagram struktur data kadang-kadang digunakan untuk mendokumentasikan table dan hubungannya.

Model *Database* Pertama, dengan suksesnya media penyimpanan disk pada tahun 1960an, maka kita dapat memiliki akses non-sekuensial atau langsung ke records (Perekaman). Dalam hal ini, database dirancang untuk menghilangkan masalah pemrosesan file sekuensial. Terdapat dua arsitektur atau model yang awalnya sukses. IBM mengembangkan dan mempromosikan DL/I atau *Data language one* yang membuat model data database dalam bentuk pohon hirarki.

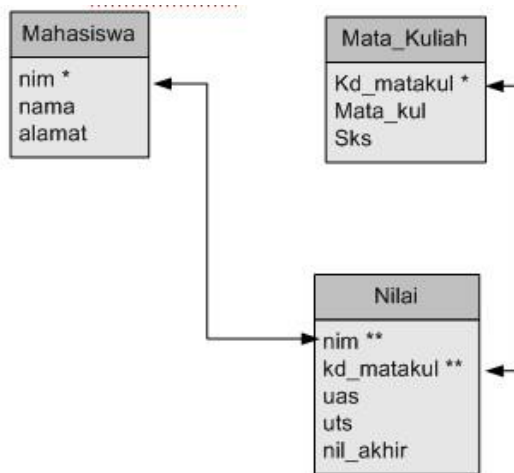


Gambar DBMS Hirarki

1. Model Relasional

Model *database* relasional pertama kali diajukan oleh E.F. Codd pada tahun 1970. Codd bekerja untuk IBM dan setelah 10 tahun melakukan penelitian, pengembangan dan lobi, dia dan temannya yang lain berhasil meyakinkan IBM untuk mengembangkan produk DBMS berdasarkan model relasional, model yang paling terkenal dalam produk ini yaitu DB2, yaitu DBMS yang masih aktif digunakan hingga saat ini.

Kemudian perusahaan seperti Oracle, Ingres, Sybase dan Informix mengembangkan juga produk DBMS berdasarkan model relasional.



Gambar Database Relasional

Dasar untuk model data jaringan terbentuk lalu di standarisasi oleh CODASYL (Conference on Data System Languages).

Pada tahun 1973 Bachman menerima penghargaan CM Turing Award yaitu nobel pada ilmu computer.

Menurut sejarah, system pemrosesan basis data terbentuk setelah masa system pemrosesan manual dan system pemrosesan berkas. System pemrosesan manual (Berbasis kertas) merupakan bentuk pemrosesan yang menggunakan dasar berupa setumpuk rekaman yang disimpan pada rak-rak berkas. Jika berkas-berkas tersebut diperlukan, berkas tersebut harus dicari pada rak-

rak tersebut. System pemrosesan berkas merupakan system komputer, dimana sekelompok rekaman disimpan pada sejumlah berkas secara terpisah. Perancangan system ini didasarkan pada kebutuhan individual pengguna, bukan kebutuhan sejumlah pengguna. Sehingga setiap aplikasi menuliskan data sendiri, alhasil ada kemungkinan data yang sama terdapat pada berkas-berkas lain yang digunakan oleh program aplikasi lain.

Sejarah DBMS (*Database Management System*), generasi pertama DBMS didesain oleh Charles Bachman di perusahaan General Electric pada awal tahun 1960, disebut sebagai penyimpanan data terintegrasi (*Integrated Data Store*). Dibentuk dasar untuk model data jaringan yang kemudian distandardisasi oleh CODASYL (*Conference on Data System Languages*).

Pada akhir 1960, IBM mengembangkan system manajemen informasi (*Information Management System*) DBMS. IMS dibentuk dari representasi data pada kerangka kerja yang disebut dengan model data hirarki. Dalam waktu yang sama, dikembangkan system SABRE sebagai hasil kerjasama antara IBM dengan perusahaan

penerbangan Amerika. *System* ini memungkinkan user untuk mengakses data yang sama pada jaringan komputer. Pada tahun 1985, Microsoft dan IBM mengumumkan perjanjian kerjasama jangka panjang untuk mengembangkan system operasi dan produk-produk perangkat lunak lainnya. Pengumuman ini adalah permulaan dimulainya OS/2, sebuah system operasi setelah masa kejayaan MS-DOS. OS/2 ini akan lebih hebat dan lebih canggih daripada MS-DOS, ia akan mampu menangani multitasking application dengan memanfaatkan kemampuan processor Intel yang terbaru yaitu 80286. OS/2 secara resmi diumumkan pada bulan april 1987, dan dijanjikan akan tersedia buat end-user pada akhir tahun tersebut.

Tetapi dalam waktu yang sangat singkat setelah pengumuman perjanjian kerjasama tersebut, IBM mengumumkan peluncuran sebuah versi spesial OS/2 yang disebut OS/2 Extended Edition. Versi lebih powerfull ini akan menyertakan sebuah Database SQL yang disebut OS/2 Database Manager, OS/2 Database Manager akan sangat berguna bagi pengembangan aplikasi yang sederhana dan kompatibel dengan DB/2 ,

sebuah Database server milik IBM yang beroperasi pada mainframe. OS/2 Database manager juga akan menyertakan SNA (System Network Architecture) communication service, yang disebut OS/2 Communication Manager. Sebagai bagian dari SSA (System Application Architecture)-nya. IBM menjanjikan semua produk-produk tersebut dapat saling bekerjasama pada masa yang akan datang.

Hal inilah yang membuat Microsoft segera mencari solusi sendiri. Pada tahun 1986, Microsoft sudah menghasilkan 197 JutaUS\$ per tahun, dengan 1153 pegawai. (Sepuluh tahun kemudian, Microsoft telah mendapatkan 6 Milyar US\$ dari bisnis softwarena, dengan hampir sekitar 18.000 pegawai). Produk-produk Microsoft hampir semuanya terfokus pada aplikasi desktop dengan produk utamanya adalah MS-DOS. Komputasi Client/Server pada saat itu belum menjadi fokus utama Microsoft dan industri komputer. Manajemen data pada sebuah PC hanyalah menjadi sebuah fantasi pada masa itu, User biasanya hanya menggunakan LOTUS 1-2-3 untuk menyimpan data. Produk dBASE buatan Ashton Tate segera menjadi sangat

populer setelah diluncurkan, Selanjutnya Anca Software merilis Paradox dan Micro Rim dengan prouk RBase-nya. Pada tahun 1986 tersebut, Microsoft belum mempunyai produk manajemen *Database* sendiri. (Tetapi pada tahun 1992, Microsoft mendapatkan sukses yang luar biasa dari produk manajemen database desktop-nya dengan Microsoft Access dan Microsoft FoxPro).

Tetapi IBM Database Manager sangatlah berbeda dari dBASE, Paradox atau RBase. Produk IBM ini walaupun tidak begitu user-friendly, tetapi ia mempunyai SQL query processor-nya sendiri dan bekerja berdasarkan transactions, hampir sama dengan Database Server yang berjalan diatas Minicomputer dan mainframe, seperti DB/2, Oracle ataupun Informix. Microsoft membutuhkan produk DBMS (Database Management System) sekaliber ini dan membutuhkannya segera!.

Microsoft berpaling kepada Sybase, Inc. Sebuah perusahaan pembuat software DBMS yang sedang 'Naik daun', yang merilis produk Data Server-nya pada bulan mei 1987 untuk Sun Microsystem yang berjalan diatas UNIX. Data Server mendapatkan reputasi berkat inovasi dan kreatifitasnya dengan Stored Procedure dan Trigger

serta paradigma baru dalam dunia komputasi yaitu: Client/Server.

Basis data merupakan kumpulan dari elemen data logis yang saling berhubungan. Basis data mengonsolidasi banyak catatan yang sebelumnya disimpan dalam file terpisah. Basis data pun merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh organisasi. Maka basis data merupakan tempat penyimpanan data yang besar dimana dapat digunakan oleh banyak pengguna. Seluruh item basis data tidak dimiliki oleh satu departemen, melainkan menjadi sumber daya perusahaan yang dapat digunakan bersama.

1.6.1 Object Oriented DBMS (OODBMS)

Pemrograman berorientasi objek (*object oriented*) mulai digunakan pada tahun 1980 dan akan dikembangkan menjadi produk DBMS berorientasi objek. Tujuan produk ini untuk menyimpan objek pemrograman berorientasi objek seperti C++ atau java, dalam sebuah database tanpa harus mentransformasikannya ke format relasional.

1.6.2 MySQL

MySQL merupakan sebuah perangkat lunak yang mengatur basis data relasional (RDBMS) yang di distribusikan gratis dengan lisensi GPL (General Public License). Pengguna dapat menggunakan MySQL secara bebas namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan bersifat komersial.

MySQL adalah server yang melayani database. Untuk membuat dan mengolah database, pada dMySQL ini dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL. Database diperlukan jika akan menginput data dari user menggunakan form HTML untuk diolah PHP agar dapat disimpan kedalam database MySQL.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya. SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.



Gambar Logo MySQL

1.6.3 Sejarah MySQL

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia. TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggap tidak cocok untuk database yang dinamis seperti web. TcX kemudian mencoba mencari

alternatif sistem database lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung indexing, sehingga performanya tidak terlalu bagus.

Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (programmer yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2.

Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis. David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di 'jual' dengan model bisnis baru. Ia mengusulkan agar

MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa “support” untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah Open Source.

Pada tahun 1995 itu juga, TcX berubah nama menjadi MySQL AB, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel “AB” di belakang MySQL, adalah singkatan dari “Aktiebolag”, istilah PT (Perseroan Terbatas) bagi perusahaan Swedia.

Keistimewaan MySQL :

MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis.

3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas

indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Kelebihan MySQL dibandingkan dengan RDBMS lainnya :

1. Berlisensi GPL dan Multi Platform.
2. Dapat diintegrasikan dengan beberapa bahasa Pemrograman seperti .Net, Java, Python, Perl yang merupakan bahasa pemrograman yang paling dominan di kalangan programmer.
3. Mendukung ODBC untuk sistem operasi Windows sehingga bisa digunakan aplikasi yang berjalan di windows.
4. Bisa dijalankan pada spesifikasi hardware yang rendah karena lebih hemat resource memory (dibandingkan database lain) sehingga mudah digunakan untuk bahan pembelajaran.
5. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa meskipun bahasa indonesia belum termasuk didalamnya.
6. MySQL dapat diintegrasikan dengan Hosting.

Kekurangan MySQL dengan RDBMS lainnya :

1. Banyak mengklaim kurang support terhadap pemrograman Visual/Desktop, sehingga sedikit yang menggunakan untuk aplikasi visual.
2. Karena berlisensi GPL sehingga sulit mendapatkan update untuk problem yang urgent, sehingga perusahaan skala menengah keatas lebih memilih RDBMS berlisensi dan disupport seperti Oracle dan MS SQL Server.
3. Sangat diragukan dalam menangani data skala besar, karena ada beberapa opini yang pro dan kontra terhadap kemampuan MySQL terhadap pengolahan data yang besar.

1.6.3.1 Kemampuan MySQL

Berikut merupakan fitur yang terdapat pada MySQL :

1. Unjuk kerja yang tinggi dalam memproses query sederhana, dalam arti dapat memproses lebih banyak SQL per satuan waktu.
2. Memiliki lebih banyak tipe data seperti: signed/unsigned integer yang memiliki panjang data sebesar 1,2,3,4 dan 8 byte, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB,

DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET dan tipe ENUM.

3. Mendukung field yang dijadikan Index, dengan maksimal 32 index dalam satu tabel. *
4. MYSQL memiliki beberapa lapisan keamanan, seperti subnetmask, nama host, dan izin akses user dengan sistem perijinan yang mendetail serta sandi/password terenkripsi.
5. Konektivitas , MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes(NT).
6. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik
7. Command and function, MySQL memiliki fungsi dan operator secara penuh yang mendukung perintah select dan where dalam query.
8. Structure Table, MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE dibandingkan DBMS lainnya.
9. Mendukung penuh terhadap kalimat SQL GROUP BY dan ORDER BY. Mendukung terhadap fungsi

penuh (COUNT(),COUNT(), DISTINCT()
AVG(), STD(), SUM(), MAX() dan MIN())

1.6.4.1 Bahasa Pemrograman pada MySQL

Terdapat beberapa API (Application Programming Interface) tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses basis data MySQL antara lain: bahasa pemrograman C, C++, C#, bahasa pemrograman Eiffel, bahasa pemrograman Smalltalk, bahasa pemrograman Java, bahasa pemrograman Lisp, Perl, PHP, bahasa pemrograman Python, Ruby, REALbasic dan Tcl. Sebuah antarmuka ODBC memanggil MyODBC yang memungkinkan setiap bahasa pemrograman yang mendukung ODBC untuk berkomunikasi dengan basis data MySQL. Kebanyakan kode sumber MySQL dalam ANSI C.

1.6.4 XAMPP



Gambar Logo XAMPP

XAMPP merupakan server yang banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi programmer pemula. Selain gratis, fiturnya tergolong lengkap dan gampang sehingga banyak digunakan oleh programmer tingkat awal.

Xampp juga merupakan salah satu aplikasi web server apache yang terintegrasi dengan mysql dan phpmyadmin. XAMPP adalah singkatan dari X, Apache Server, MySQL, PHPMyadmin, dan Python. Huruf X di depan menandakan XAMPP dapat diinstal pada berbagai operating system. XAMPP dapat diinstal pada Windows, Linux, MacOS, dan Solaris.

Sampai saat ini, XAMPP masih didapatkan secara gratis, bebas di download dan digunakan tanpa harus membayar.

Xampp merupakan buatan dari team apache friends. Xampp dapat di download langsung dari situs resminya.

Xampp 1.8.3 memiliki beberapa fitur seperti :

- a. Apache 2.4.6
- b. MySQL 5.6.14
- c. PHP 5.5.6
- d. phpMyAdmin 4.0.9
- e. FileZilla FTP Server 0.9.41
- f. Tomcat 7.0.42
- g. Strawberry Perl 5.16.3.1 Portable
- h. XAMPP Control Panel 3.2.1

BAB 2

2.1 Implementasi

Implementasi menurut KBBI (Kamus Besar Bahasa Indonesia) yaitu pelaksanaan atau penerapan. Sedangkan pengertian umum adalah suatu tindakan atau pelaksana rencana yang telah disusun secara cermat dan rinci (matang).

Kata implementasi sendiri berasal dari bahasa Inggris “to implement” artinya mengimplementasikan. Tak hanya sekedar aktivitas, implementasi merupakan suatu kegiatan yang direncanakan serta dilaksanakan dengan serius juga mengacu pada norma-norma tertentu guna mencapai tujuan kegiatan.

Implementasi merupakan tindakan yang dilakukan untuk mencapai suatu tujuan yang ditetapkan dalam suatu keputusan. Pada tindakan ini berusaha untuk mengubah keputusan tersebut menjadi pola-pola operasional serta berusaha mencapai perubahan-perubahan besar atau kecil sebagaimana yang telah diputuskan sebelumnya. Implementasi pada hakikatnya juga merupakan upaya pemahaman apa yang seharusnya terjadi setelah program dilaksanakan.

2.2 Implementasi Sistem

Implementasi sistem adalah tahap penerapan sistem yang akan dilakukan jika sistem disetujui termasuk program yang telah dibuat pada tahap perancangan sistem agar siap untuk dioperasikan. Implementasi sistem merupakan tahap penerapan sistem agar sistem tersebut

dapat digunakan atau dioperasikan untuk mencapai tujuan.

Adapun beberapa tujuan dari implementasi sistem yaitu sebagai berikut :

- a. Membuat desain sistem selama melakukan penelitian dan analisa.
- b. Menguji dan mendokumentasikan prosedur dan program yang diperlukan
- c. Menyelesaikan desain sistem yang telah distujui.
- d. Memperhitungkan sistem yang telah dibuat sesuai kebutuhan yang akan digunakan

Terdapat 4 tahap dalam pengimplementasian sistem, yaitu :

a) Membuat dan menguji basis data dan jaringan
Penerapan sistem yang baru atau perbaikan sistem dibuat pada basis data dan jaringan yang telah ada.

b) Membuat dan menguji program
Tahap ini merupakan tahap pertama untuk siklus pengembangan sistem yang spesifik, bertujuan untuk mengembangkan rencana yang lebih terperinci dalam pengembangan dan penguian.

- c) Memasang dan menguji sistem baru

Pada tahap ini dilakukan untuk meyakinkan bahwa kebutuhan sistem lama terpenuhi pada sistem yang baru.

- d) Serta mengirim sistem baru kedalam sistem yang lama.

Pada tahap ini yaitu mengubah secara perlahan-lahan sistem lama menjadi sistem baru.

2.3 Sistem

Sistem berasal dari bahasa Yunani “*Systema*” yang berarti kesatuan. Pengertian sistem diambil dari asal mula sistem yang berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) yang memiliki pengertian bahwa suatu sistem merupakan suatu kesatuan yang didalamnya terdiri dari komponen atau elemen yang berhubungan satu dengan yang lainnya, yang berfungsi untuk memudahkan aliran informasi, materi atau *energy*. Ada beberapa elemen yang membentuk sebuah sistem, yaitu :

- a. Tujuan

Setiap sistem memiliki tujuan agar setiap sistem yang akan dibuat ataupun telah terbuat menjadi terarah dengan keinginan ataupun kebutuhan pengguna.

b. Masukan

Masukan pada sistem yaitu segala sesuatu yang masuk kedalam sistem dan selanjutnya dapat menjadi bahan yang akan di proses. Masukan tersebut dapat berupa hal-hal yang berwujud maupun tidak berwujud.

c. Proses

Proses merupakan bagian yang melakukan perubahan ataupun transformasi dari masukan yang menjadi keluaran yang berguna dan lebih bernilai, seperti : informasi dan produk.

d. Keluaran

Keluaran merupakan hasil dari pemrosesan.

e. Batas

Pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem.

f. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian dengan umpan balik digunakan untuk mengendalikan suatu masukan maupun proses,

untuk mengatur agar sistem berjalan sesuai dengan tujuan yang diinginkan.

2.3.1 Kategori Sistem

Sistem terbagi menjadi 2 kategori :

- a. Atas dasar keterbukaan
 - 1) Sistem terbuka, dimana pihak luar dapat mengakses dan mempengaruhi sistem tersebut.
 - 2) Sistem tertutup
- b. Atas dasar komponen
 - 1) Sistem fisik, menggunakan komponen dengan energy
 - 2) Sistem non-fisik atau konsep yang berisikan ide-ide

2.3.2 Karakteristik Sistem

- a. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

- b. Batasan Sistem (*Boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

c. Lingkungan luar sistem (*Environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem.

Lingkungan dapat bersifat menguntungkan yang tetap harus dijaga dan yang merugikan harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem

d. Penghubung sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (output) dari subsistem akan menjadi masukan (input) untuk subsistem lain melalui penghubung.

e. Masukan sistem (Input)

Masukan adalah energy yang dimasukkan kedalam sistem, yang dapat berupa perawatan (maintenance input),

dan masukkan sinyal (signal input). Maintenance input merupakan energy yang dimasukkan agar sistem dapat beroperasi. Signal input adalah energy yang diproses untuk didapatkan keluaran.

f. Keluaran sistem (ouput)

Keluaran sistem adalah hasil dari energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolahan sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku mejadi bahan jadi.

h. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (goal) atau sasaran (objective). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.

2.3.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dalam beberapa sudut pandang, seperti :

a. Sistem Abstrak (abstract System)

Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik.

b. Sistem fisik (physical system)

Sistem fisik adalah sistem yang ada secara fisik atau bentuknya.

c. Sistem alamiah (natural system)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia.

d. Sistem buatan manusia

Sistem buatan manusia adalah sistem yang melibatkan interaksi antara manusia dengan mesin.

e. Sistem tertentu (deterministic system)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan.

f. Sistem tak tentu (Probabilistic system)

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat di prediksi karena mengandung unsur probabilistic.

g. Sistem tertutup (close system)

Sistem tertutup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem

bekerja otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya relatively closed system.

h. Sistem terbuka (open system)

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima input dan output dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik.

2.4 Sistem Penjadwalan

Sistem penjadwalan adalah informasi berupa data dalam proses mengatur rencana kegiatan dengan pembagian waktu pelaksanaan.

2.5 Penjadwalan

Penjadwalan Persoalan penjadwalan adalah persoalan pengalokasian pekerjaan ke mesin, pada kondisi mesin mempunyai kapasitas dan jumlah terbatas. Secara umum masalah penjadwalan dapat dijelaskan sebagai n job (J_1, J_2, \dots, J_n) yang harus diproses

di m mesin (M_1, M_2, \dots, M_n). Waktu yang diperlukan untuk memproses 6 pekerjaan J_1 pada mesin M adalah P setiap job harus diproses tanpa dihentikan selama waktu proses p mesin hanya dapat menangani satu job pada saat yang sama, dan secara terus menerus tersedia sejak waktu nol (time zero). Pemecahan permasalahan yang diinginkan adalah mendapatkan jadwal yang optimal, yaitu menyelesaikan semua pekerjaan dengan mendapatkan jadwal yang optimal.

Penjadwalan (*scheduling*), didefinisikan sebagai proses pengalokasian sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu. Definisi ini dapat dijabarkan dalam arti yang berbeda, yaitu :

1. Penjadwalan merupakan sebuah fungsi pengambilan keputusan dalam menentukan jadwal yang paling tepat.
2. Penjadwalan merupakan teori yang berisi sekumpulan prinsip, model, teknik, dan konklusi logis dalam proses pengambilan keputusan. Sistem penjadwalan adalah informasi berupa data dalam proses mengatur rencana kegiatan dengan pembagian waktu pelaksanaan. Priority

Scheduling merupakan algoritma penjadwalan berprioritas.

3. Kegiatan akademik adalah kegiatan pembelajaran di dalam dan/atau di luar ruang kuliah, studio dan/atau laboratorium, pengerjaan tugas-tugas, evaluasi pembelajaran, dan kegiatan administrasi yang menyertainya.

Berdasarkan Kamus Besar Bahasa Indonesia, jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja. Jadwal juga didefinisikan sebagai daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan penjadwalan merupakan proses, cara, perbuatan menjadwalkan atau memasukkan dalam jadwal

2.5.1 Tujuan Penjadwalan

Tujuan penjadwalan adalah untuk mengurangi waktu keterlambatan dari batas waktu yang ditentukan agar dapat memenuhi batas waktu yang telah disetujui dengan konsumen, penjadwalan juga dapat meningkatkan produktifitas mesin dan mengurangi waktu menganggur. Produktifitas mesin meningkat maka waktu menganggur

berkurang, secara tidak langsung perusahaan dapat mengurangi biaya produksi. Semakin baik suatu penjadwalan semakin menguntungkan juga bagi perusahaan dan bisa menjadi acuan untuk meningkatkan keuntungan dan strategi bagi perusahaan dalam pemuasan pelanggan.

2.5.2 Kelebihan dan Kekurangan dalam Penjadwalan

Forward scheduling (Penjadwalan maju) adalah penjadwalan yang dimulai segera setelah saat job siap, mulai dari time zero dan bergerak searah dengan pergerakan waktu.

1. Kelebihan Penjadwalan

Dalam Penjadwalan dapat disusun secara SPT (Shortest Processing Time) sehingga didapatkan suatu penjadwalan produksi dengan rata-rata flowtime yang minimum.

2. Kelemahan Penjadwalan

Adanya kemungkinan untuk melewati batas waktu (due date) yang ditetapkan konsumen.

2.6 Jadwal Kegiatan Akademik

Jadwal kegiatan akademik adalah pedoman dalam kegiatan akademik untuk jangka waktu satu tahun akademik. Jadwal akademik adalah tulang punggung waktu untuk kegiatan yang akan diselenggarakan dan informasi penting lainnya untuk diuraikan dalam kalender kegiatan

Jadwal kegiatan akademik merupakan suatu patokan waktu yang akan direncanakan baik dilaksanakan pada waktu yang akan datang, dengan adanya jadwal kegiatan akademik ini membuat penggunaanya terstruktur dalam mengerjakan suatu kegiatan baik akan dilaksanakan maupun direncanakan.

2.7 Karyawan

Karyawan adalah tenaga kerja yang melakukan pekerjaan dan memberikan hasil kerjanya kepada pengusaha yang mengerjakan dimana hasil karyanya itu sesuai dengan profesi atau pekerjaan atas dasar keahlian sebagai matapencariannya.

Karyawan merupakan salah satu alasan organisasi atau perusahaan dalam menentukan keberlangsungan aktifitas usaha.

Disisi lain, tanpa manajer yang mampu membangun produktivitas dan kinerja karyawannya dengan baik, maka tidak akan mewujudkan iklim manajemen SDM yang kondusif, alih-alih strategis. Oleh karena itu, bagi organisasi, pasrtisipasi antar keduanya sangat penting dalam upaya membangun produktivitas dan kinerja secara kolektif.