

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Adapun penelitian terkait ini, peneliti akan membandingkan hasil penelitiannya dengan lima jurnal yaitu :

Penelitian pertama Fadila Pratiwi, Fince Tinus Waruwu, Dito Putro Utomo, Rian Syahputra (2019) berjudul “Penerapan Metode Aras Dalam Pemilihan Asisten Perkebunan Terbaik Pada PTPN V” penelitian ini berisikan sistem pengambilan keputusan dengan Aras. Hasil yang dicapai adalah dilakukan dengan cara menghitung nilai-nilai alternatif berdasarkan algoritma aras yang hasilnya bertujuan untuk mendapatkan alternatif dengan ranking tertinggi.

Penelitian kedua Lia Ciky Lumban Gaol, Nelly Astuti Hasibuan (2018) berjudul “Sistem Pendukung Keputusan Pemilihan Team Leader Shift Terbaik Dengan Menggunakan Metode Aras” penelitian ini membahas tentang sistem pengambilan keputusan dengan metode Aras. Hasil yang dicapai adalah untuk memecahkan masalah pemilihan team leader shift terbaik dengan menggunakan metode tersebut didapatkan bahwa kriteria yang paling dominan adalah kriteria Kejujuran dibandingkan dengan keempat kriteria lainnya yaitu: Pengalaman, Pelatihan, Penampilan dan Wawasan.

Penelitian Ketiga Liza Handayani, M. Syahrizal, Kennedy Tampubolon (2019) Berjudul “Pemilihan Kepling Teladan Menerapkan Metode Rank Order Centroid (Roc) Dan Metode Additive Ratio Assessment (Aras) Di Kecamatan

Medan Area”Penelitian Ini Menggunakan Sistem Pendukung Keputusan Dengan Metode Rank Order Centroid (Roc) Dan Metode Additive Ratio Assessment (Aras). Hasil yang dicapai adalah dalam pemilihan kepling teladan menggunakan sistem pendukung keputusan ini dapat menentukan kriteria dan nilai bobot untuk setiap alternatif dengan menggunakan metode rank order centroid (ROC) dapat menentukan nilai bobot dengan tingkat prioritasnya, dan dengan menggunakan metode additive ratio assesement (ARAS) dinilai dapat menyelesaikan permasalahan dalam pemilihan kepling teladan di Kecamatan Medan Area.

Penelitian keempat Andri Anto Tri Susilo (2017 ) berjudul “Penerapan Metode Profile Matching Pada Sistem Pendukung Keputusan Pemilihan Ketua Program Studi (Studi Kasus : Program Studi Teknik Informatika STMK Musi Rawas) ”metode yang digunakan metode Profile Matching . Hasil dari penelitian Sistem pendukung keputusan yang dibangun dapat membantu pimpinan yayasan untuk menentukan calon Ketua Program Studi Teknik Informatika STMIK Musi Rawas

Penelitian kelima Umayatul Choerohnur, Joan Angelina Widiyans, Islamiyah (2017) Berjudul “Sistem Pendukung Keputusan Seleksi Pemilihan Calon Kepala Desa Menggunakan Metode Profile Matching” penelitian ini menggunakan sistem pendukung keputusan dengan metode profile matching. Hasil pengembangannya adalah sub kriteria dan variabel nilai yang digunakan dalam proses seleksi calon kepala desa sangat berpengaruh dalam hasil perhitungan yang diperoleh.

Perbedaan penelitian yang penulis lakukan dengan penelitian sebelumnya adalah penulis menggunakan metode *Additive Ratio Assesment* dan *Profile Matching* dalam pemilihan kepala jurusan TKR pada SMK SWASTA PAB 5.

Hasil dari penelitian ini dapat digunakan sebagai bahan pertimbangan baik bagi lembaga pendidikan untuk menetapkan sistem pendukung keputusan pemilihan kepala jurusan dan dapat menjadi alat bantu atau solusi bagi pihak sekolah dalam pengambilan keputusan dalam pemilihan kepala jurusan TKR terbaik.

## **II.2. Landasan Teori**

### **II.2.1 Sistem Pendukung Keputusan**

Sistem pendukung keputusan merupakan sistem penghasil informasi yang ditujukan pada suatu masalah yang harus dibuat oleh manajer, sistem pendukung keputusan merupakan suatu sistem informasi yang ditujukan untuk membantu manajemen dalam memecahkan masalah yang dihadapinya. Defenisi selengkapnya adalah sistem penghasil informasi spesifik yang ditunjukan untuk memecahkan suatu masalah tertentu yang harus dipecahkan oleh manajer pada berbagai tingkatan. Sistem pendukung keputusan juga merupakan suatu sistem informasi berbasis computer yang menghasilkan berbagai alternative keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data atau model. Sistem Pendukung Keputusan (*Decision Support System*) adalah sistem berbasis computer yang

interaktif dalam membantu pengambilan keputusan dengan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tak terstruktur. (Pratiwi, 2020)

### **II.2.2 Kepala Jurusan**

Kepala jurusan merupakan guru SMK yang diberikan tugas tambahan khusus untuk membantu Kepala Sekolah dalam mengelola kegiatan akademik disekolahnya. Kegiatan akademik di Sekolah Menengah Kejuruan (SMK) meliputi Pembelajaran (Teori dan Praktik), Kunjungan Industri, Praktik Kerja Lapangan (PKL), Ujian Nasional, Uji Kompetensi Kejuruan (UKK), dsb. Disamping mengajar, guru yang diberi tugas tambahan sebagai Kepala Jurusan di SMK juga harus membuat laporan berkala terkait tugasnya sebagai Kepala Jurusan (Wasito, 2019)

### **II.2.3 Metode ARAS (*Additive Ratio Assesment*)**

Metode *aras* adalah sebuah utilitas nilai fungsi yang menentukan efisiensi relatif kompleks dari alternatif yang layak adalah langsung sebanding dengan efek relatif dari nilai dan bobot kriteria utama yang dipertimbangkan dalam proyek proyek. Dalam melakukan proses perankingan, metode aras memiliki tiga tahapan yang harus dilakukan untuk menghitung metode aras yaitu: (Judas, 2018)

Adapun langkah-langkah dari metode *Additive Ratio Assesment* (ARAS) sebagai berikut :

### 1. Pembentukan Decision Making Matriks

$$X = \begin{bmatrix} X_{0i} & X_{0j} & \dots & X_{0n} \\ X_{i1} & X_{ij} & \dots & X_{in} \\ \vdots & \vdots & \ddots & \vdots \\ X_{mi} & X_{mj} & \dots & X_{mn} \end{bmatrix} \quad (i = 0, m; \dots j = 1, n) \quad (1)$$

Dimana :

m = Jumlah Alternatif

n = Jumlah Kriteria

$X_{ij}$  = Nilai performa dari alternatif ; terhadap kriteria  $J_{xoj}$  = nilai optimum dari kriteria J

Jika nilai optimum kriteria J ( $x_{oj}$ ) Tidak diketahui, maka :

$$x_{oj} = \text{Max} \frac{\text{Max}}{i} = x_{ij} \text{, if } \frac{\text{Max}}{i} \text{, } x_{ij} \text{ is Preference} \quad (2)$$

$$x_{oj} = \text{Max} \frac{\text{Min}}{i} = x_{ij} \text{, if } \frac{\text{Min}}{i} \text{, } x_{ij} \text{ is Prefeerable} \quad (3)$$

### 2. Penormalisasian matriks keputusan untuk semua kriteria

a. Jika kriteria beneficial (Max) maka dilakukan normalisasi mengikuti :

$$X_{ij}^* = \frac{x_{ij}}{\sum_{i=0}^m x_{ij}} \quad \rightarrow \text{Dimana : } x_{ij}^* \text{ adalah nilai normalisasi} \quad (4)$$

b. Jika kriteria non beneficial maka dilakukan normalisasi :

$$\rightarrow \text{Tahap 1} = x_{ij} - \frac{1}{x_{ij}} \quad (5)$$

$$\rightarrow \text{Tahap 2} = R - \frac{x_{ij}}{\sum_{i=0}^m x_{ij}} \quad (6)$$

### 3. Menentukan bobot matriks yang sudah dinormalisasikan

$$D = [d_{ij}] m \times n = r_{ij} \cdot w_j \quad \rightarrow \text{Dimana : } w_j = \text{bobot kriteria} \quad (7)$$

4. Menentukan nilai fungsi optimalisasi ( $S_i$ )

$$S_i = \sum_{j=1}^n d_{ij} \quad (i = 1, 2, \dots, m; j = 1, 2, \dots, n) \quad (8)$$

Dimana  $S_i$  adalah nilai fungsi optimalisasi alternatif  $i$ . Nilai terbesar adalah nilai yang terbaik, dan nilai yang paling sedikit adalah yang terburuk. Dengan memperhitungkan proses, hubungan proporsional dengan nilai dan bobot kriteria yang diteliti berpengaruh pada hasil akhir.

5. Menentukan tingkat peringkat tertinggi dari alternative

$$K_i = \frac{S_i}{S_0} \quad (9)$$

Dimana  $S_i$  dan  $S_0$  merupakan nilai kriteria optimalitas, diperoleh dari persamaan sudah jelas. Itu dihitung nilai  $U_i$  berada pada interval dan merupakan pesanan yang diinginkan didahulu efisiensi relatif kompleks dari alternatif yang layak bisa ditemukan sesuai dengan nilai fungsi utilitas.

### Contoh Kasus :

Terdapat 11 orang Calon Polisi Militer sebagai berikut :

**Tabel II.1. Alternatif / Calon Polisi Militer**

Alternatif	Kriteria				
	Tepat Waktu	Sesuai Aturan	Jadwal Kegiatan	Melaksanakan Tugas	Tanggung Jawab
Sudarsono ( $X_0$ )	Iya	Iya	Sangat Banyak	Kurang Baik	Iya
Sundoro S.H ( $X_1$ )	Tidak	Tidak	Cukup Sedikit	Bagus	Iya
Wahyudi ( $X_2$ )	Iya	Iya	Sangat Sedikit	Bagus	Tidak
Azis Basri R, S.H ( $X_3$ )	Tidak	Iya	Sangat Banyak	Kurang Bagus	Tidak
Nanang HS ( $X_4$ )	Iya	Tidak	Sangat Banyak	Sangat Bagus	Iya
Moh Arfai Lukman ( $X_5$ )	Iya	Iya	Sangat Banyak	Kurang Bagus	Tidak
Ridwan ( $X_6$ )	Iya	Tidak	Cukup Sedikit	Sangat Bagus	Iya
Bambang Irawan ( $X_7$ )	Tidak	Tidak	Sangat Sedikit	Kurang Bagus	Iya
Sukarnan S, S.H ( $X_8$ )	Tidak	Iya	Sangat Sedikit	Bagus	Tidak
Keriadi ( $X_9$ )	Iya	Tidak	Sangat Sedikit	Sangat Bagus	Iya
Sutanto Juliardi ( $X_{10}$ )	Iya	Iya	Cukup Sedikit	Bagus	Tidak

Berikut ini adalah rating kecocokan dari setiap alternatif pada kriteria yang sudah ditentukan :

**Tabel II.2. Rating Kecocokan dari setiap alternative**

Alternatif	Kriteria				
	C1	C2	C3	C4	C5
$X_0$	100	100	100	60	100
$X_1$	80	80	80	80	100
$X_2$	100	100	60	80	80
$X_3$	80	100	100	60	80
$X_4$	100	80	100	100	100
$X_5$	100	80	80	100	100
$X_6$	80	80	60	60	100
$X_7$	80	100	60	80	80
$X_8$	100	80	60	100	100
$X_9$	100	100	80	80	80
$X_{10}$	100	100	100	60	80

Untuk menyelesaikan masalah diatas dengan metode ARAS akan dilakukan sesuai dengan langkahlangkah yang telah dijelaskan.

- a. Pembentukan matriks keputusan

**Tabel II.3. Matriks keputusan**

alternatif	Kriteria				
	C1	C2	C3	C4	C5
X <sub>0</sub>	100	100	100	60	100
X <sub>1</sub>	80	80	80	80	100
X <sub>2</sub>	100	100	60	80	80
X <sub>3</sub>	80	100	100	60	80
X <sub>4</sub>	100	80	100	100	100
X <sub>5</sub>	100	80	80	100	100
X <sub>6</sub>	80	80	60	60	100
X <sub>7</sub>	80	100	60	80	80
X <sub>8</sub>	100	80	60	100	100
X <sub>9</sub>	100	100	80	80	80
X <sub>10</sub>	100	100	100	60	80
Criteria Type	Max	Max	Max	Min	Max

- b. Merumuskan Matriks Keputusan (3.2)

$$X_{ij} = \begin{bmatrix} 100 & 100 & 100 & 60 & 100 \\ 80 & 80 & 80 & 80 & 100 \\ 100 & 100 & 60 & 80 & 80 \\ 80 & 100 & 100 & 60 & 80 \\ 100 & 80 & 100 & 100 & 100 \\ 100 & 80 & 80 & 100 & 100 \\ 80 & 80 & 60 & 60 & 100 \\ 80 & 100 & 60 & 80 & 80 \\ 100 & 80 & 60 & 100 & 100 \\ 100 & 100 & 80 & 80 & 80 \\ 100 & 100 & 100 & 60 & 80 \end{bmatrix}$$

- c. Normalisasi Matriks Keputusan (3.3)

C1 :

$$R_{01} = \frac{100}{1020} = 0,0981$$

$$R_{11} = \frac{80}{1020} = 0,0784$$

$$R_{21} = \frac{100}{1020} = 0,0981$$

$$R_{31} = \frac{80}{1020} = 0,0784$$

$$R_{41} = \frac{100}{1020} = 0,0981$$

$$R_{51} = \frac{100}{1020} = 0,0981$$

$$R_{61} = \frac{80}{1020} = 0,0784$$

$$R_{71} = \frac{80}{1020} = 0,0784$$

$$R_{81} = \frac{100}{1020} = 0,0981$$

$$R_{91} = \frac{100}{1020} = 0,0981$$

$$R_{101} = \frac{100}{1020} = 0,0981$$

Lanjutkan sampai C5 dengan cara dan langkah2 yang sama seperti C1. Dari perhitungan diatas dapat diperoleh Matriks Keputusan yang telah dinormalisasikan sebagai berikut :

$$X^* = \begin{bmatrix} 0,0981 & 0,1 & 0,125 & 0,016 & 0,1 \\ 0,0784 & 0,08 & 0,1 & 0,125 & 0,1 \\ 0,0981 & 0,1 & 0,075 & 0,125 & 0,08 \\ 0,0784 & 0,1 & 0,125 & 0,016 & 0,08 \\ 0,0981 & 0,08 & 0,125 & 0,01 & 0,1 \\ 0,0981 & 0,08 & 0,1 & 0,01 & 0,1 \\ 0,0784 & 0,08 & 0,75 & 0,016 & 0,1 \\ 0,0784 & 0,1 & 0,75 & 0,125 & 0,08 \\ 0,0981 & 0,08 & 0,75 & 0,01 & 0,1 \\ 0,0981 & 0,1 & 0,1 & 0,125 & 0,08 \\ 0,0981 & 0,1 & 0,125 & 0,016 & 0,08 \end{bmatrix}$$

Menentukan bobot matriks yang sudah dinormalisasi dengan melakukan perkalian matriks yang telah dinormalisasikan terhadap bobot kriteria (3,5)

- d. Menentukan bobot matriks yang sudah dinormalisasi dengan melakukan perkalian matriks yang telah dinormalisasikan terhadap bobot kriteria (3,5)

$$D01 = x01 * 1 = 0.0981 * 0.1 = 0.0098$$

$$D11 = x11 * 1 = 0.0784 * 0.1 = 0.0078$$

$$D21 = x21 * 1 = 0.0981 * 0.1 = 0.0098$$

$$D31 = x31 * 1 = 0.0784 * 0.1 = 0.0078$$

$$D41 = x41 * w1 = 0.0981 * 0.1 = 0.0098$$

$$D51 = x51 * 1 = 0.0981 * 0.1 = 0.0098$$

$$D61 = x61 * 1 = 0.0784 * 0.1 = 0.0078$$

$$D71 = x71 * 1 = 0.0784 * 0.1 = 0.0078$$

$$D81 = x81 * 1 = 0.0981 * 0.1 = 0.0098$$

$$D91 = x91 * 1 = 0.0981 * 0.1 = 0.0098$$

$$D101 = x101 * 1 = 0.0981 * 0.1 = 0.0098$$

Lakukan sampai D105 dengan cara dan langkah-langkah yang sama seperti D01. Dari perhitungan diatas dapat diperoleh hasil matriks sebagai berikut:

$$D = \begin{bmatrix} 0,0098 & 0,15 & 0,025 & 0,04 & 0,03 \\ 0,0078 & 0,12 & 0,02 & 0,3125 & 0,03 \\ 0,0098 & 0,15 & 0,015 & 0,3125 & 0,016 \\ 0,0078 & 0,15 & 0,025 & 0,04 & 0,016 \\ 0,0098 & 0,12 & 0,025 & 0,025 & 0,03 \\ 0,0098 & 0,12 & 0,02 & 0,025 & 0,03 \\ 0,0078 & 0,12 & 0,015 & 0,04 & 0,03 \\ 0,0078 & 0,15 & 0,015 & 0,3125 & 0,016 \\ 0,0098 & 0,12 & 0,015 & 0,025 & 0,03 \\ 0,0098 & 0,15 & 0,02 & 0,3125 & 0,016 \\ 0,0098 & 0,15 & 0,025 & 0,04 & 0,016 \end{bmatrix}$$

Menentukan nilai dari fungsi optimalisasi dengan menjumlahkan nilai kriteria pada setiap alternatif dari hasil perkalian matriks dengan bobot yang telah dilakukan sebelumnya.

$$S0 = 0,0098 + 0,15 + 0,025 + 0,04 + 0,03 = 0,2548$$

$$S1 = 0,0078 + 0,12 + 0,02 + 0,3125 + 0,03 = 0,4903$$

Lanjutkan sampai S10 dengan cara yang sama seperti langkah di atas.

- e. Menentukan tingkatan peringkat tertinggi dari setiap alternatif dengan cara membagikan nilai alternatif terhadap alternatif 0 (A0)(3.7).

$$K_0 = \frac{0,2548}{3,7448} = 0,0989$$

$$K_1 = \frac{0,4903}{3,7448} = 0,0881$$

Lanjutkan sampai K10 dengan cara yang sama seperti langkah di atas. Dari perhitungan diatas dapat diperoleh hasil tabel tingkatan peringkat dari setiap alternatif sebagai berikut:

**Tabel II.4. Nilai Untuk Masing-masing Alternatif**

Alternatif	Keterangan	C1	C2	C3	C4	C5	S	K
A <sub>0</sub>	Sudarsono	0,00 98	0, 15	0,0 25	0,04	0,0 3	0,25 48	0,06 81
A <sub>1</sub>	Sundoro, S.H	0,00 78	0, 12	0,0 2	0,31 25	0,0 3	0,49 03	0,13 09
A <sub>2</sub>	Wahyudi	0,00 98	0, 15	0,0 15	0,31 25	0,0 16	0,50 33	0,13 44
A <sub>3</sub>	Azis Basri R, S.H	0,00 78	0, 15	0,0 25	0,04	0,0 16	0,23 88	0,06 38
A <sub>4</sub>	Nanang HS	0,00 98	0, 12	0,0 25	0,02 5	0,0 3	0,20 98	0,05 60
A <sub>5</sub>	Moh Arfai Lukman	0,00 98	0, 12	0,0 2	0,02 5	0,0 3	0,20 48	0,05 47
A <sub>6</sub>	Ridwan	0,00 78	0, 12	0,0 15	0,04	0,0 3	0,21 28	0,05 68
A <sub>7</sub>	Bambang Irawan	0,00 78	0, 15	0,0 15	0,31 25	0,0 16	0,50 13	0,13 39
A <sub>8</sub>	Sukarman S, S.H	0,00 98	0, 12	0,0 15	0,02 5	0,0 3	0,19 98	0,05 34
A <sub>9</sub>	Keriadi	0,00 98	0, 15	0,0 2	0,31 25	0,0 16	0,68 83	0,18 34
A <sub>10</sub>	Sutanto Juliardi	0,00 98	0, 15	0,0 25	0,04	0,0 16	0,24 08	0,06 43

Maka dari hasil perhitungan tingkatan peringkat tertinggi dari alternatif. Dimana nilai dari masing-masing alternatif dibagi dengan A<sub>0</sub> sehingga menghasilkan nilai Utility yang akan dijadikan tingkatan peringkat dengan nilai tertinggi yang terpilih.

**Tabel II.5. Alternatif Digolongkan dari Nilai Tertinggi**

Alternatif	Keterangan	Nilai (KI)	Rangking
A <sub>9</sub>	Keriadi	0,1834	1
A <sub>2</sub>	Wahyudi	0,1344	2
A <sub>7</sub>	Bambang Irawan	0,1339	3
A <sub>1</sub>	Sundoro, S.H	0,1309	4
A <sub>0</sub>	Sudarsono	0,0681	5
A <sub>10</sub>	Sutanto Juliardi	0,0643	6
A <sub>3</sub>	Azis Basri R, S.H	0,0638	7
A <sub>6</sub>	Ridwan	0,0568	8
A <sub>4</sub>	Nanang HS	0,0560	9
A <sub>5</sub>	Moh Arfai Lukman	0,0547	10
A <sub>8</sub>	Sukarman S, S.H	0,0534	11

#### II.2.4 Metode *Profile Matching*

Metode *profile matching* merupakan suatu proses yang sangat penting dalam manajemen SDM dimana terlebih dahulu ditentukan kompetensi (kemampuan) yang diperlukan oleh suatu jabatan. Kompetensi atau kemampuan tersebut haruslah dapat dipenuhi oleh pemegang atau calon pemegang jabatan. Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara kompetensi individu kedalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga Gap), semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar yang berarti memiliki peluang lebih besar untuk pegawai yang menempati posisi tersebut (Kardiaman, 2018)

Dalam *profile matching* dilakukan identifikasi terhadap kelompok karyawan yang baik maupun buruk. Para karyawan dalam kelompok tersebut diukur menggunakan beberapa kriteria penilaian. Jikalau pelaksana yang baik memperoleh skor yang berbeda dari pelaksana yang buruk atau sebuah karakteristik, maka variabel tersebut berfaedah untuk memilih pelaksana yang baik

Terdapat beberapa tahapan dan perumusan perhitungan dengan metode *Profile Matching* yaitu :

1. Pemetaan Gap Kompetensi

Gap yang dimaksud disini adalah perbedaan antara profil penerima penghargaan dengan profil pegawai atau bisa ditunjukkan pada rumus dibawah ini :

$$\text{Gap} = \text{Profil Calon} - \text{Profil Ideal} \quad (1)$$

## 2. Pembobotan

Setelah diperoleh gap pada masing-masing pegawai, setiap profil pegawai diberi bobot nilai dengan patokan tabel bobot nilai gap. Seperti yang terlihat pada tabel dibawah ini :

**Tabel II.6 Tabel Bobot Nilai Gap**

No.	Selisih GAP	Bobot Nilai	Keterangan
1.	0	5	Kompetensi sesuai dengan yang dibutuhkan
2.	1	4,5	Kompetensi individu kelebihan 1 tingkat/level
3.	-1	4	Kompetensi individu kurang 1 tingkat/level
4.	2	3,5	Kompetensi individu kelebihan 2 tingkat/level
5.	-2	3	Kompetensi individu kurang 2 tingkat/level
6.	3	2,5	Kompetensi individu kelebihan 3 tingkat/level
7.	-3	2	Kompetensi individu kurang 3 tingkat/level
8.	4	1,5	Kompetensi individu kelebihan 4 tingkat/level
9.	-4	1	Kompetensi individu kurang 4 tingkat/level

Sumber: Kardiawan (2018:60)

### 3. Perhitungan dan Pengelompokan *Core* dan *Secondary Factor*

Setelah menentukan bobot nilai gap untuk setiap aspek, kemudian setiap aspek dikelompokkan menjadi 2 kelompok, yaitu kelompok *Core Factor* dan *Secondary Factor*. Pengelompokan *core factor* ditunjukkan menggunakan rumus dibawah ini :

$$NCF = \frac{\sum Nc}{\sum Ic} \quad (2)$$

Keterangan :

NCF : Nilai rata-rata *core factor*

NC : Jumlah total nilai *core factor*

IC : Jumlah item *core factor*

Sementara itu, perhitungan *secondary factor* bisa ditunjukkan dengan rumus berikut :

$$NSF = \frac{\sum Ns}{\sum Is} \quad (3)$$

Keterangan :

NSF : Nilai rata-rata *secondary factor*

NS : Jumlah total nilai *secondary factor*

IS : Jumlah item *secondary factor*

### 4. Penghitungan Nilai Total

Dari hasil perhitungan setiap aspek, berikutnya dihitung nilai total berdasarkan persentase dari *core* dan *secondary* yang diperkirakan

berpengaruh terhadap kinerja tiap-tiap profil. Penghitungan bisa dilihat pada rumus dibawah ini :

$$N = (x)\%NCF + (x)\%NSF \quad (4)$$

Keterangan :

N : Nilai Total dari aspek

(x)% :Nilai Persen yang diinputkan

NCF : Nilai rata-rata *Core Factor*

NSF : Nilai rata-rata *Secondary Factor*

#### 5. Perhitungan Penentuan Ranking

Hasil akhir dari proses profile matching adalah ranking dari kandidat. Penentuan ranking mengacu pada hasil perhitungan tertentu dengan aspek yang dicontohkan. Contoh perhitungan tersebut bisa ditunjukkan dengan rumus di bawah ini :

$$Ranking = (x)\%Ni + (x)\%Ns + (x)\%Np \quad (5)$$

Keterangan :

Ni : Nilai kapasitas intelektual

Ns : Nilai sikap kerja

Np : Nilai perilaku

(x)% : Nilai Persen yang diinputkan

**Contoh Kasus :**

Memberikan rekomendasi berupa karyawan terbaik berdasarkan peringkat. Pemberian peringkat ditujukan untuk karyawan yang berhak mendapat promosi jabatan atau mendapatkan bonus. Aspek atau kriteria yang dinilai adalah:

## 1. Kejujuran

Kejujuran merupakan aspek penting dalam SDM karyawan yang dipertimbangkan oleh perusahaan.

**Tabel II.7. Aspek Kejujuran (C1)**

Kejujuran	Nilai
Sangat jujur	5
Jujur	4
Sedang	3
Kurang jujur	2
Tidak jujur	1

## 2. Kesetiaan pada perusahaan

Setiap perusahaan pasti menginginkan adanya sikap loyal atau kesetiaan pada karyawan mereka.

**Tabel II.8. Aspek Kesetiaan (C2)**

Kesetiaan	Nilai
Bekerja pada perusahaan lebih dari 5 tahun	5
Bekerja pada perusahaan selama 3-5 tahun	4
Bekerja pada perusahaan selama 2-3 tahun	3
Bekerja pada perusahaan selama 1-2 tahun	2
Karyawan baru	1

## 3. Sikap (Attitude)

Sikap (attitude) dianggap penting karena berhubungan dengan bagaimana orang lain memperlakukan dan menerima seseorang.

**Tabel II.9. Aspek Sikap / Attitude (C3)**

Sikap ( <i>Attitude</i> )	Nilai
Sangat baik	5
Baik	4
Lumayan	3
Kurang baik	2
Tidak baik	1

## 4. Produktivitas

Seorang karyawan dapat dikatakan produktif apabila mampu menghasilkan barang atau jasa sesuai dengan diharapkan dalam waktu yang singkat dan tepat atau sesuai dengan yang diharapkan.

**Tabel II.10. Aspek Produktivitas (C4)**

Produktivitas	Nilai
Produktivitas tinggi dan memiliki semangat tinggi	5
Produktivitas baik	4
Produktivitas rata-rata	3
Produktivitas rendah	2
Produktivitas sangat rendah	1

## 5. Disiplin

Disiplin kerja ditunjukkan oleh ketepatan waktu karyawan ketika hadir di kantor.

**Tabel II.11. Aspek Disiplin (C5)**

Disiplin	Nilai
Tidak pernah / sangat jarang terlambat masuk kerja	5
Dalam 1 bulan 2-3 kali terlambat masuk kerja	4
Dalam 1 bulan 4-5 kali terlambat masuk kerja	3
Dalam 1 bulan 6-8 kali terlambat masuk kerja	2
Sering terlambat	1

## 6. Kemampuan bekerja sama

Kerjasama (*team work*) adalah keinginan seseorang untuk bekerja sama dengan orang lain secara kooperatif dan menjadi bagian dari kelompok.

**Tabel II.12. Aspek Kerja Sama (C6)**

<b>Produktivitas</b>	<b>Nilai</b>
Mampu bekerja sama dengan sangat baik	5
Lumayan mampu bekerja sama	4
Sedang	3
Kurang mampu bekerja sama	2
Tidak mampu bekerja sama	1

Proses pemilihan karyawan terbaik dengan menggunakan metode profile matching adalah sebagai berikut:

1. Misalkan aspek penilaian seperti terlihat pada tabel 8 berikut.

**Tabel II.13. Aspek Nilai Kriteria**

<b>Kode</b>	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>N4</b>	<b>N5</b>	<b>N6</b>
Hadi (K-0001)	4	1	4	3	5	5
Chris (K-0002)	5	4	5	4	4	4
Tuti (K-0003)	4	2	3	2	1	2

2. Dari data pada tabel 3.8, diambil kriteria ideal sebagai nilai gap (nilai gap dapat ditentukan sendiri) lalu kurangi dengan masing-masing aspek nilai, maka akan didapatkan perhitungan nilai gap seperti terlihat pada tabel 3.9 berikut.

**Tabel II.14. Perhitungan Selisih Nilai Gap**

<b>Kode</b>	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>N4</b>	<b>N5</b>	<b>N6</b>
Hadi (K-0001)	4	1	4	3	5	5
Chris (K-0002)	5	4	5	4	4	4
Tuti (K-0003)	4	2	3	2	1	2
<b>GAP</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>
Hadi (K-0001)	-1	-4	-1	-2	0	0
Chris (K-0002)	0	-1	0	-1	-1	-1
Tuti (K-0003)	-1	-3	-2	-3	-4	-3

3. Setelah diperoleh gap pada masing-masing. Setiap profil karyawan diberi bobot nilai dengan memperhatikan tabel bobot nilai gap seperti yang terlihat pada tabel 10 berikut.

**Tabel II.15. Bobot Nilai**

No	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih(kompetensi sesuai dengan yang dibutuhkan)
2	1	4,5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kekurangan 1 tingkat/level
4	2	3,5	Kompetensi individu kelebihan 2 tingkat/level
5	-2	3	Kompetensi individu kekurangan 2 tingkat/level
6	3	2,5	Kompetensi individu kelebihan 3 tingkat/level
7	-3	2	Kompetensi individu kekurangan 3 tingkat/level
8	4	1,5	Kompetensi individu kelebihan 4 tingkat/level
9	-4	1	Kompetensi individu kekurangan 4 tingkat/level

4. Dengan demikian, setiap karyawan akan memiliki nilai bobot seperti terlihat pada tabel 11 berikut.

**Tabel II.16. Tabel Bobot**

Kode	B1	B2	B3	B4	B5	B5
Hadi (K-0001)	4	1	4	3	5	5
Chris (K-0002)	5	4	5	4	4	4
Tuti (K-0003)	4	2	3	2	1	2

5. Yang menjadi *core factor* adalah kejujuran (C1), kesetiaan (C2), sikap (C3) dan produktivitas (C4), sedangkan yang menjadi *secondary factor* adalah disiplin (C5) dan kemampuan bekerja sama (C6). Selanjutnya dilakukan perhitungan *core factor* dan *secondary factor* sebagai berikut:

- a. Untuk Hadi

$$\text{NCF} = (\text{B1} + \text{B2} + \text{B3} + \text{B4}) / 4$$

$$= (4 + 1 + 4 + 3) / 4 = 3$$

$$\text{NSF} = (\text{B5} + \text{B6}) / 2$$

$$= (5 + 5) / 2 = 5$$

## b. Untuk Chris

$$\begin{aligned} \text{NCF} &= (B1 + B2 + B3 + B4) / 4 \\ &= (5 + 4 + 5 + 4) / 4 = 4.5 \end{aligned}$$

$$\begin{aligned} \text{NSF} &= (B5 + B6) / 2 \\ &= (4 + 4) / 2 = 4 \end{aligned}$$

## c. Untuk Tuti

$$\begin{aligned} \text{NCF} &= (B1 + B2 + B3 + B4) / 4 \\ &= (4 + 2 + 3 + 2) / 4 = 2.75 \end{aligned}$$

$$\begin{aligned} \text{NSF} &= (B5 + B6) / 2 \\ &= (1 + 2) / 2 = 1.5 \end{aligned}$$

6. Bila input persentase core factor adalah 60% dan secondary factor adalah 40%, maka perhitungan nilai total adalah sebagai berikut:

## a. Untuk Hadi

$$N = (60\% * \text{NCF}) + (40\% * \text{NSF})$$

$$N = (60\% * 3) + (40\% * 5)$$

$$N = 3.8$$

## b. Untuk Chris

$$N = (60\% * \text{NCF}) + (40\% * \text{NSF})$$

$$N = (60\% * 4.5) + (40\% * 4)$$

$$N = 4.3$$

## c. Untuk Tuti

$$N = (60\% * \text{NCF}) + (40\% * \text{NSF})$$

$$N = (60\% * 2.75) + (40\% * 1.5)$$

$$N = 2.25$$

7. Dengan demikian, pemberian peringkat pada masing-masing karyawan seperti dapat dilihat pada tabel 12 berikut.

**Tabel II.17. Hasil Peringkat Karyawan**

Karyawan	N	Peringkat
Chris (K-0002)	4,3	1
Hadi (K-0001)	3,8	2
Tuti (K-0003)	2,25	3

### II.2.5 UML (*Unified Modelling Language*)

*Unified Modelling Language* (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi. *Unified Modeling Language* (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sitem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.



- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*. (Alfina & Harahap, 2019)



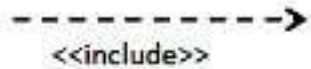
Diagram-diagram yang terdapat dalam UML sangat banyak, berikut ini beberapa diagram yang sering di gunakan dalam pengembangan sistem yaitu :


#### 1. Use Case Diagram

Use Case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu :

**Tabel II.18. Simbol Use Case Diagram**

No.	Simbol	Keterangan
1.		<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit yang dinyatakan dengan menggunakan kata kerja.
2.		<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang

		<p>mengaktifkan fungsi dari target sistem.</p> <p>Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>
3.		<p>Asosiasi antara actor dan <i>Use Case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data</p>
4.		<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem</p>
5.		<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program</p>
6.		<p><i>Extend</i>, merupakan perluasan dari <i>use</i></p>




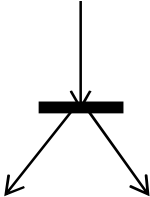
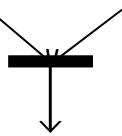
	<i>Caselain jika kondisi atau syarat terpenuhi.</i>
---	---

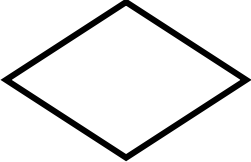
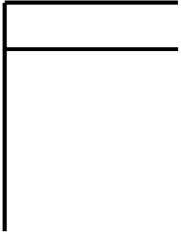
(Sumber : Rosa dan Shalahuddin, 2018)

## 2. Activity diagram

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem proses atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

**Tabel II.19. Simbol Activity Diagram**

No	Simbol	Keterangan
1.		<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
2.		<i>End Point</i> , akhir aktivitas.
3.		<i>Activities</i> menggambarkan suatu proses /kegiatan bisnis.
4.		<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
5.		<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.

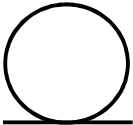
6.		pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
7.		<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

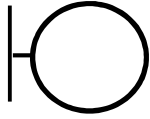
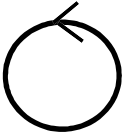

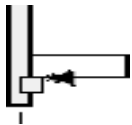

(Sumber : Rosa dan Shalahuddin, 2018)


### 3. Sequence Diagram

*Sequence diagram* menggambarkan kelakuan objek pada use case dengan mendiskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

**Tabel II.20. Simbol Sequence Diagram**

No.	Simbol	Keterangan
1.		<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadilandakan untuk menyusun basis data.
2.		<i>Boundary Class</i> , berisi kumpulan kelas

		yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.
3.		<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
4.		<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
5.		<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
6.		<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi..
7.		<i>Lifeline</i> , garis titik-titik yang terhubung

		dengan objek sepanjang <i>life line</i> terdapat  <i>activation</i>
--	---	---

(Sumber : Rosa dan Shalahuddin, 2018)

#### 4. Class Diagram

*Class diagram* adalah merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturanaturan dan tanggung jawab entitas yang menentukan perilaku sistem.

**Tabel II.21. Simbol *Class Diagram***

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

(Sumber : Rosa dan Shalahuddin, 2018)

#### II.2.6 Microsoft Visual Studio

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 (yang sering juga disebut dengan VB .Net 2010) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana

(tool) untuk menghasilkan program-program aplikasi berbasis windows. *Visual basic* adalah sebuah bahasa pemrograman yang berpusat pada *object (Object Oriented Programming)* digunakan dalam pembuatan aplikasi *Windows* yang berbasis *Graphical User Interface*, hal ini menjadikan *Visual Basic* menjadi bahasa pemrograman yang wajib diketahui dan dikuasai oleh setiap programmer. Beberapa karakteristik obyek tidak dapat dilakukan oleh *Visual Basic* misalnya seperti *Inheritance* tidak bisa module dan *Polymorphism* secara terbatas bisa dilakukan dengan deklarasi *class module* yang mempunyai *Interface* tertentu. Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti : (Wiliani, 2017)

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : *kontrol ActiveX, file Help*, aplikasi Internet dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program berakhiran EXE yang bersifat executable atau dapat langsung dijalankan.

*Visual Studio 2010* adalah bahasa yang cukup mudah untuk dipelajari. Bagi programmer pemula yang baru ingin belajar program, lingkungan *Visual Studio* dapat membantu membuat program dalam sekejap mata. Sedang bagi programmer tingkat lanjut, kemampuan yang besar dapat digunakan untuk membuat program-program yang kompleks, misalnya lingkungan net-working atau client server. Bahasa *Visual Studio* cukup sederhana dan menggunakan kata-kata bahasa Inggris yang umum digunakan. Kita tidak perlu lagi menghafalkan sintaks-sintaks maupun format-format bahasa yang bermacam-macam, di dalam Visual Basic semuanya

sudah disediakan dalam pilihan-pilihan yang tinggal diambil sesuai dengan kebutuhan. Selain itu, sarana pengembangannya yang bersifat visual memudahkan kita untuk mengembangkan aplikasi berbasis Windows, bersifat mouse-driven (digerakkan dengan *mouse*) dan berdaya guna tinggi (Wiliani, 2017)

### **II.2.7 Basis Data (*Database*)**

Basisdata merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lain. Basis data atau *database* merupakan salah satu komponen yang penting dalam sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi pemakainya, Sistem basis data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan dengan yang lainnya dan untuk membuatnya tersedia beberapa aplikasi yang bermacam-macam dalam suatu sistem organisasi. Sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan. (Sutopo et al., 2018)

### **II.2.7 Microsoft SQL Server**

SQL (*Structured Query Language*) adalah bahasa non procedural untuk mengakses data pada *database* relasional. SQL adalah bahasa *database* yang dipergunakan dalam menyelesaikan permasalahan dalam *database* serta mempunyai kelebihan dalam mengolah data. Standar SQL mula-mula

didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*the American National Standards Institute*) yang dikenal dengan sebutan SQL86.

Dengan menggunakan SQL, kita dapat melakukan hal-hal berikut:

1. Memodifikasi struktur *database* .
2. Mengubah, mengisi, menghapus isi *database*.
3. Mentransfer data antara *database* yang berbeda.

SQL ada yang dikembangkan untuk PC dan ada juga yang dikembangkan untuk dapat mengakomodasi *database* yang sangat besar. (Iswandy et al., 2018)