

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Sebagai bahan pertimbangan dalam penelitian ini akan dicantumkan beberapa hasil penelitian terdahulu yang berkaitan dengan pengambilan keputusan menggunakan metode *profile matching* dan SAW beberapa peneliti.

1. Yohana DEwi Lulu W. Rani Maya Sari, Heni Rachmawati Penelitian berjudul Sitem Pendukun SAW Di PT. Pertamina RU II Dumai. Perusahaan Pertamina melakukan proses ini untuk memenuhi karyawan terbaiknya. Adapun kriterianya yaitu pekerja berprestasi, pekerja aktif, pekerja peduli *safety*, pekerja sehat. Untuk membantu dalam proses pengolahan data maka akan dibangun sistem pendukung keputusan dengan model *fuzzy multiple attribute decision making*. Aplikasi ini diharapkan dapat membantu perusahaan dalam mengukur kinerja perusahaan, sehingga perusahaan dapat berkembang dengan pesat sesuai visi dan misi perusahaan tersebut.
2. I Gede Bendesa Sumbawa, I Made Agus Wirawan, I Made Gede Sunarya (2015). Penelitian ini berjudul Pengembangan Sistem Pendukung Keputusan Pemilihan Pegawai Terbaik Menggunakan Metode SAW Di PT. TIRTA JAYA ABADI SINGARAJA. Penelitian ini bertujuan agar pegawai selalu memacu semangat dalam dirinya untuk terus meningkatkan atau bahkan tetap mempertahankan dedikasi atau kinerjanya di perusahaan dari tahun ketahun. Perusahaan menerapkan SPK dalam pengambilan

keputusan terhadap suatu masalah bisa dilakukan dengan cepat (I Gede Bendesa, dkk, 2015 : 4).

3. Asfan Muqtakir, Irwan purdianto, (2013) Penelitian berjudul Sistem Pendukung Keputusan Kenaikan Jabatan Menggunakan Metode *Profile Matching*. Penelitian ini bertujuan untuk membuat sebuah sistem penunjang keputusan kenaikan jabatan di PT.IKSG dengan menggunakan metode *profile matching*, membantu pengambilan keputusan yang kesulitan dalam memutuskan karyawan yang cocok untuk menempati suatu jabatan tertentu.
4. Rahmat Hidayat, (2016) Penelitian berjudul Menentukan Promosi Jabatan Karyawan Dengan Menggunakan Metode *Profile Matching* Dan Metode *Promothee*. Penelitian ini bertujuan menerapkan metode *profile matching* dan *promethee* sebagai penunjang pemilihan karyawan yang layak dalam promosi jabatan. Kelebihan Sistem ini dapat membantu untuk mengambil keputusan promosi jabatan karyawan dengan tepat karna sudah memiliki kriteria yang ada pada suatu jabatan tertentu dengan menggunakan metode *promethee* dan *profile matching*.

II.2. Uraian Terkait

II.2.1 Sistem Pendukung Keputusan

Menurut Mat dan Watson, Sistem Penunjang Keputusan (SPK) merupakan suatu sistem interaktif yang membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-

masalah yang sifatnya semi terstruktur dan tidak terstruktur. Sedangkan menurut Moore dan Chang, SPK adalah sistem yang dapat dikembangkan, mampu mendukung analisis data dan pemodelan keputusan, berorientasi pada perencanaan masa mendatang, serta tidak bisa direncanakan *interval* (periode) waktu pemakaiannya. Bonezek, Hosapple dan Whinston mendefinisikan SPK sebagai suatu sistem yang berbasis komputer yang terdiri dari 3 komponen yang berinteraksi satu dengan yang lainnya, yaitu (Chaulina Alfianti Oktavia, et al., 2015 : 145) :

1. *Language system*, adalah suatu mekanisme untuk menjembatani (*interface*) pemakai dan komponen lainnya (Chaulina Alfianti Oktavia, et al., 2015 : 145).
2. *Knowledge system* adalah repositori pengetahuan yang berhubungan dengan masalah tertentu baik berupa data maupun prosedur (Chaulina Alfianti Oktavia, et al., 2015 : 145).
3. *Problem processing system* adalah sebagai penghubung kedua komponenlainnya, berisi satu atau beberapa kemampuan manipulasi atau menyediakan masalah secara umum, yang diperlukan dalam pengambilan keputusan (Chaulina Alfianti Oktavia, et al., 2015 : 145).

II.2.2 Karakteristik Sistem Pendukung Keputusan

Karakteristik dari Sistem Pendukung Keputusan yang membedakan dari sistem informasi lainnya adalah (Chaulina Alfianti Oktavia, et al., 2015 : 145) :

1. SPK dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur.

2. Dalam proses pengolahannya, SPK mengkombinasikan penggunaan model-model/teknik-teknik analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari/interogasi informasi.
3. SPK dirancang sedemikian rupa sehingga dapat digunakan/dioperasikan denganmu dan oleh orang-orang yang tidak memiliki dasar kemampuan pengoperasian komputer yang tinggi. Oleh karena itu pendekatan yang digunakan biasanya model interaktif.
4. SPK dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi. Sehingga mudah disesuaikan dengan berbagai perubahan lingkungan yang terjadi dan kebutuhan pemakai.

II.2.3 Manfaat Sistem Pendukung Keputusan

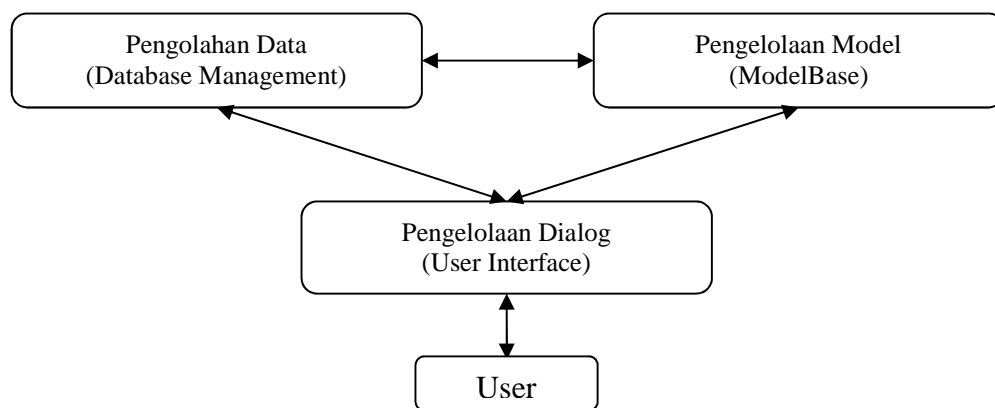
Sistem pendukung keputusan dapat memberikan berbagai manfaat atau keuntungan bagi pemakainya, antara lain:

1. Memperluas kemampuan pengambilan keputusan dalam memproses data / informasi bagi pemakainya.
2. Membantu pengambilan keputusan dalam hal penghematan waktu yang dibutuhkan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu sistem pendukung keputusan, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya,

karena sistem pendukung keputusan mampu menyajikan berbagai alternatif (Chaulina Alfianti Oktavia, et al., 2015 : 145).

II.2.4. Komponen Sistem Pendukung Keputusan

Secara umum Sistem Pendukung Keputusan dibangun oleh tiga komponen besar yaitu *database Management*, *Model Base* dan *Software System/User Interface*. Komponen SPK tersebut dapat digambarkan seperti ini:



Gambar II.1. Komponen Sistem Pendukung Keputusan (SPK)
(Sumber ; Khoirul Huda; dkk : 2)

II.3. *Unified Modeling Language (UML)*

UML adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artefact* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya). UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan peranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C (I Made Budi Adnyana, 2016 : 51).

1. Use Case Diagram

Use Case Diagram adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun (I Made Budi Adnyana, 2016 : 51).



Gambar II.3.1 Use Case Diagram
(Sumber ;I Made Budi Adnyana ; 2016 : 53)

2. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Tabel II.3. Multiplicity Class Diagram

Multiplicity	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 maksimum 4

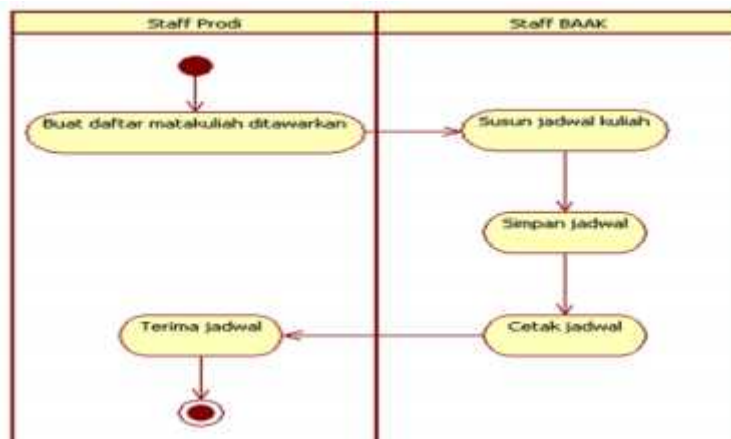
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti pewarisan, asosiasi (I Made Budi Adnyana, 2016 : 52).



Gambar II.3.2 Class Diagram
 (Sumber ;I Made Budi Adnyana ; 2016 : 54)

3. Activity Diagram

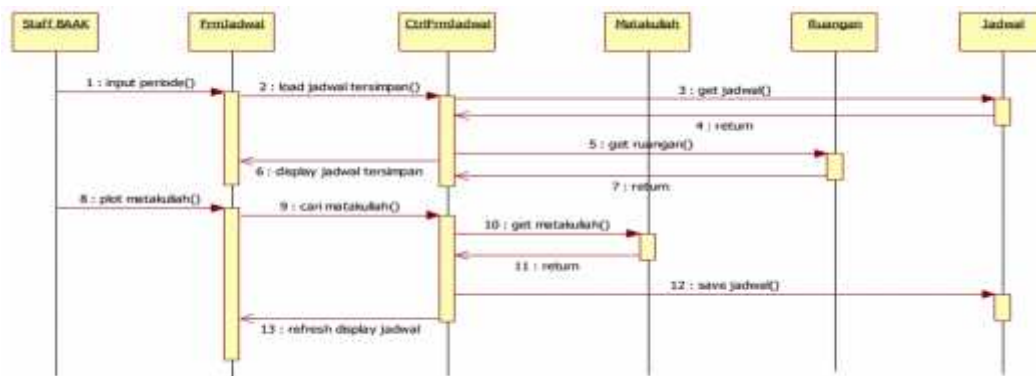
Activity diagram menggambarkan sebagai alur aktifitas dalam sistem yang sedang dirancang. Bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih. Aktifitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana *actor* menggunakan sistem untuk melakukan aktivitas. (I Made Budi Adnyana, 2016 : 52).



Gambar II.2.3. Activity Diagram
 (Sumber ;I Made Budi Adnyana ; 2016 : 53)

4. Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. (I Made Budi Adnyana, 2016 : 52)..



Gambar II.2.4 Sequence Diagram
(Sumber ;I Made Budi Adnyana ; 2016 : 54)

II.4. Database

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu *database* menunjukkan satu lingkup perusahaan atau instansi. *Database* juga merupakan kumpulan data yang umumnya menggambarkan aktifitas-aktifitas dan pelakunya dalam suatu organisasi. Sistem *database* merupakan sistem komputer yang digunakan untuk menyimpan dan mengelola data tersebut (Ade Hendini, 2016 : 107-108).

Konsep dasar dari basis data (*database*) adalah kumpulan dari catatan-catatan atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya, penjelasan ini disebut skema (Ade Hendini, 2016 : 107-108).

II.5. *Entity Relationship Diagram (ERD)*

Menurut Utomo (2010:15) “ERD merupakan *tool* analisis sistem pertama yang memusatkan pada data dan keterkaitan antar data serta pengorganisasian data”. Menurut Ladjamudin (2005:142) “*Entity Relationship Diagram (ERD)* adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam secara abstrak”.

Berikut ini komponen-komponen ERD, hubungan entitas terdiri dari (Mulia Rahmayu, 2015 : 161):

1. *Entity*

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokan dalam empat jenis nama yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

2. *Atribut*

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *relationship* (Mulia Rahmayu, 2015 : 161).

3. *Kardinalitas*

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu keentitas yang lain dan begitu juga sebaliknya (Mulia Rahmayu, 2015 : 161).

II.6. Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi. Kamus data terdapat pada tahapan analisis dan perancangan. Pada tahap analisis, kamus data berfungsi untuk mendefinisikan data yang mengalir pada sistem. Sedangkan pada tahap perancangan, kamus data ini digunakan untuk merancang masukan dan keluaran seperti laporan serta basis data. Pada DFD aliran data memiliki sifat global, sedangkan pada kamus data dibuat berdasarkan aliran data yang terdapat pada DFD (Indrajani, 2015 : 30).

Tabel II.V. Notasi Kamus Data

Notasi	Keterangan
=	<i>Is Composed Of</i>
+	<i>And</i>
()	<i>Optional (May be present or absent)</i>
{ }	<i>Iteration</i>
[]	<i>Select one of several alternative choices</i>
**	<i>Comment</i>
@	<i>Identifier (key field) for a store</i>
	<i>Separates alternative choices in the construct</i>

(Sumber ; Indrajani ; 2015 : 31)

Contoh kamus data, antara lain:

$name = courtesy-title + first-name + (middle-name) + last-name$

$courtesy-title = [Mr. | Miss | Mrs. | Ms. | Dr. | Profesor]$

$first-name = \{legal-character\}$

$middle-name = \{legal-character\}$

$last-name = \{legal-character\}$

$legal-character = [A-Z|a-z|0-9|'|_|]$

II.6. Normalisasi

Proses Normalisasi merupakan proses pengelompokan data ke dalam bentuk tabel atau relasi atau file untuk menyatakan entitas dan hubungan sehingga terwujud satu bentuk basis data yang mudah dimodifikasi. Tahap proses normalisasi untuk menghasilkan bentuk yang normal memerlukan beberapa langkah berikut (Janner Simarmata; 2007 : 197) :

1. Bentuk normalisasi tidak normal (*Un Normalized Form* / UNF)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu. Data dikumpulkan apa adanya sesuai dengan saat menginput.

Relasi *Un Normalized Form* (UNF) mempunyai kriteria sebagai berikut (Edy Sutanta ; 2011 : 176-179) :

- a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap)

- b. Jika relasi membuat *set atribut* berulang (*non single values*)
- c. Jika relasi membuat *atribut non atomic value*

2. Bentuk normalisasi pertama (*First Norm Form / 1NF*)

Bentuk normal kesatu memiliki ciri yakni pembentukan setiap data dalam satu *record*. Bentuk normal kesatu bisa dikenakan dalam tabel yang belum ternormalisasi. Tabel yang ternormalisasi adalah tabel yang memiliki atribut berulang sehingga tabel yang tidak normal bisa dikelompokkan menjadi atribut yang berulang dan tidak berulang (Janner Simarmata; 2007 : 200).

Relasi disebut juga *First Norm Form (1NF)* jika memenuhi kriteria sebagai berikut (Edy Sutanta ; 2011 : 176-179) :

- a. Jika seluruh atribut dalam relasi bernilai *atomic (atomic value)*
- b. Jika seluruh atribut dalam relasi bernilai tunggal (*single value*)
- c. Jika relasi tidak memuat set atribut berulang
- d. Jika semua *record* mempunyai sejumlah atribut yang sama.

Permasalahan dalam *First Norm Form (1NF)* adalah sebagai berikut :

- a. Tidak dapat menyisipkan informasi parsial

Terhapusnya informasi ketika menghapus sebuah *record*

3. Bentuk normalisasi kedua (*Second Normal Form / 2NF*)

Bentuk normal kedua dilakukan bila data berada dalam bentuk normal pertama. Semua atribut bukan kunci haruslah bergantung pada kunci

primer (Janner Simarmata; 2007 : 201). Relasi disebut sebagai *Second Normal Form* (2NF) jika memenuhi kriteria sebagai berikut (Sumber ; Edhy Sutanta ; 2011 : 179) :

- a. Jika memenuhi kriteria *First Norm Form* (1NF)
- b. Jika semua atribut nonkunci *Functional Dependence* (FD) pada *Primary Key* (PK)

Permasalahan dalam *Second Normal Form* / 2NF adalah sebagai berikut:

- a. Kerangkapan data (*data redundancy*)
- b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*)
- c. Proses pembaharuan data tidak efisien

Kriteria tersebut mengidentifikasi bahwa antara atribut dalam *Second Normal Form* masih mungkin mengalami *Third Norm Form*. Selain itu, relasi *Second Normal Form* (2NF) menuntut telah didefinisikan atribut *Primary Key* (PK) dalam relasi. Mengubah relasi *First Norm Form* (1NF) menjadi bentuk *Second Normal Form* (2NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasikan *Functional Dependence* (FD) relasi *First Norm Form* (1NF)
- b. Berdasarkan informasi tersebut, dekomposisi relasi *First Norm Form* (1NF) menjadi relasi-relasi baru sesuai *Functional Dependence* nya. Jika menggunakan diagram maka simpul-simpul

yang berada pada puncak diagram ketergantungan data bertindak *Primary Key* (PK) pada relasi baru.

4. Bentuk normalisasi ketiga (*Third Norm Form* / 3NF)

Suatu relasi disebut sebagai *Third Norm Form* jika memenuhi kriteria sebagai berikut :

- a. Jika setiap atribut nonkunci tidak (TDF) (*Non Transitive Dependency*) terhadap *Primary Key* (PK)

Permasalahan dalam *Third Norm Form* (3NF) adalah keberadaan penentu yang tidak merupakan bagian dari *Primary Key* (PK) menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai *Foreign Key* (FK) (duplikasi berbeda dengan keterangan data).

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Norm Form* (3NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasi TDF relasi *Second Normal Form* (2NF)
- b. Berdasarkan informasi tersebut, dekomposisi relasi *Second Normal Form* (2NF) menjadi relasi-relasi baru sesuai TDF-nya.

Untuk membantu pemahaman tentang bentuk dan proses normalisasi, berikut diberikan contoh relasi dalam bentuk tidak normal, kemudian secara bertahap diubah hingga menjadi optimal dalam bentuk 3NF. Contoh-contoh relasi yang digunakan sengaja dibuat sederhana untuk memfokuskan pembahasan pada bentuk dan proses normalisasi, bukan pada aspek kelengkapan data sesuai kondisi nyata lapangan ditunjukkan pada Tabel II.6.

Tabel II.6. Contoh Proses Normalisasi

Kode_Supplier	Status	Kota	Kode_Barang	Jumlah_Barang
S01	10	Jakarta	B01	100
			B02	150
			B03	200
S02	20	Surabaya	B04	250
			B05	200
S03	30	Yogyakarta	B06	150
			B07	100

(Sumber ; Edhy Sutanta ; 2011 : 179)

II.7. *SQLServer*

SQL Server 2008 adalah suatu *Relational Database Management System* (RDBMS) yang digunakan untuk menyimpan data. Data yang disimpan pada *database* bisa dalam skala kecil maupun besar.

SQL Server 2008 adalah sebuah DBMS (*Database Management System*) yang dibuat oleh microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL server 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (WidyaAnggraini,dkk, : 3).

II.8. *Microsoft Visual Basic .NET*

Microsoft Visual Basic.NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para *programmer*

dapat membangun aplikasi *Windows Forms*, Aplikasi web berbasis ASP.NET dan juga aplikasi *command line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#* atau *Visual J#*), atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio .NET*. Bahasa *Visual Basic .NET* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan di atas *.NET Framework*. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft* dan versi baru ini tidak kompatibel dengan versi terdahulu.

Teknologi *.NET* muncul karena adanya beberapa alasan pada teknologi aplikasi *Microsoft* yang lama (Ahmad Rais Ruli, 2017 : 11).

VB.NET mempunyai beberapa fitur baru yang membuat bahasa VB menjadi lebih kuat sehingga dapat mematahkan mitos bahwa VB hanya bahasa mainan (*toy language*) bila dibandingkan dengan bahasa lain seperti C++ dan Java. Fitur-fitur tersebut antara lain :

1. Dukungan *Object Oriented Programming*

VB.NET adalah bahasa pemrograman yang penuh *Object Oriented*. Jadi *VB.NET* mendukung fitur-fitur OOP seperti *inheritance*, *interface*, *method overloading* dan *polymorphism* (Ahmad Rais Ruli, 2017 : 11).

2. *Structure Exception Handling*

Untuk menggantikan perintah *On Error Goto* pada VB6, *VB.NET* menyediakan *error handling* yaitu *Try..Catch..Finally*. *Error handling*

pada *VB.NET* ini lebih mudah digunakan karena hanya cukup menaruh kode yang akan dicek dibagian *Try* dan menyiapkan *exception handling* nya di bagian *catch*.

3. GDI+

GDI+ adalah *library* grafik yang digunakan untuk mengembangkan aplikasi *windows form*(Ahmad Rais Ruli, 2017 : 12).

4. *Web Services* dan *Web Form*

Dengan penggunaan *VB.NET* kita dapat membuat aplikasi berbasis web dengan menggunakan *Web Form* (ASP.NET). Kita juga dapat membuat aplikasi *web service* untuk membuat *three tier application*.

5. *Cross-Languege Interoperability*

Oleh karena disetiap program yang berjalan di *.NET* dikompilasi menjadi *assembly* maka dapat dibuat aplikasi dengan bahasa pemrograman yang berbeda yang berjalan diatas *platform .NET* seperti C# dan C++. Jadi dapat juga menggunakan komponen yang dibuat menggunakan C# atau C++ untuk digunakan di VB.

6. *Multithreading*

Fitur ini sangat berguna apabila terdapat aplikasi yang proses komputasinya memakan waktu lama.

7. *Type Safe Collection*

Fitur ini mulai ada di *.NET 2.0* (VB 8 atau VB 2005). Dengan fitur ini dapat dibuat *object collection* yang *type safe*.

8. LINQ (*Language Integrated Query*)

Fitur ini mulai ada pada .NET 3.5 (VB 9 atau VB 2008). Fitur ini digunakan untuk dilakukan *query* data yang ditambahkan kedalam bahasa VB dan C# sehingga kedua bahasa tersebut dapat melakukan *query* ke objek *database* (Ahmad Rais Ruli, 2017 : 11-12).

II.9. Metode *Profile Matching*

Profile Matching atau pencocokan profil adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati. Untuk memudahkan pelaksanaan kaderisasi dan jenjang karir dari tiap pegawai maka dibuat sebuah sistem yang bertujuan memudahkan proses, penyusunan, dan pengenalan target (dalam hal ini pegawai) dalam memudahkan penyusunan jenjang karir dan kaderisasi dari suatu perusahaan (Rahmat Hidayat, 2016: 59-60).

Tabel II.8. Keterangan Sub Aspek Kriteria

Kriteria	Keterangan Sub Kriteria
Aspek Kecerdasan	CS : <i>Common Sense</i> VI : <i>Verbalisasi Ide</i> SB : <i>Sistematika Berfikir</i> PSR : <i>Penalaran dan Solusi Real</i> KN : <i>Konsentrasi</i> LP : <i>Logika Praktis</i> FB : <i>Fleksibilitas Berfikir</i> IK : <i>Imajinasi Kreatif</i> ANT : <i>Antisipasi</i>
Aspek Sikap Kerja	K : <i>Energi Psikis</i> KTJ : <i>Ketelitian dan Tanggung Jawab</i>

	KH : <i>Kehati-hatian</i> PP : <i>Pengendalian Perasaan</i> DB : <i>Dorongan Berprestasi</i> E : <i>Etika</i> VB: <i>Vitalitas dan Perencanaan</i>
Aspek Perilaku	D: <i>Dominance</i> I: <i>Influences</i> S: <i>Steadiness</i> C: <i>Compliance</i>

(Sumber; Rahmat Hidayat; 2016: 59-60)

II.10. Metode SAW

Metode SAW merupakan metode yang paling dikenal dan banyak digunakan orang dalam menghadapi situasi MADM. Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk sebuah *alternative* diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating yang dapat dibandingkan lintas atribut bobot dan tiap atribut. *Rating* tiap atribut telah melewati proses normalisasi sebelumnya.

Metode SAW di kenal sebagai istilah penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap *alternative* pada semua atribut.

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) kesuatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut.

Dengan r adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i= 1,2,\dots,m$ dan $j= 1,2,\dots,n$.

$$\text{Max } \frac{x_{ij}}{x_{ij}^{\text{max}}}$$

jika j adalah atribut keuntungan (*benefit*).

$\frac{Min \quad x_{ij}}{x_{ij}}$ jika j atribut biaya (*cost*).

Nilai preferensi setiap alternatif (V_i) diberikan sebagai:

$$v_i \sum_{j=1}^n w r_{ij} \dots\dots(1).$$

$V = W \times R$

Dengan:

$V =$ Nilai Matriks

$W =$ Matriks rating kepentingan (bobot)

$r =$ rating

Nilai V_i yang lebih besar mengindikasikan bahwa alternative A_i lebih

terpilih (Yohana Dewi Lulu, dkk : 2).