

BAB II

LANDASAN TEORI

II.1. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem berbasis komputer yang membantu para pengambil keputusan mengatasi berbagai masalah melalui interaksi Pengambilan keputusan (*Decision Making*) adalah melakukan penilaian dan menjatuhkan pilihan. Keputusan ini diambil setelah melalui beberapa perhitungan dan pertimbangan alternatif. *Decision support system* atau sistem pendukung keputusan secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semi terstruktur. Secara khusus, Sistem pendukung keputusan didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manager maupun sekelompok manager dalam memecahkan masalah semi terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu (Muhammad Yudin Ritonga ; 2014 : 19).

Sistem pendukung keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi sistem pendukung keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi sistem pendukung keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.

Sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur) dan sistem pemrosesan atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan (Dicky Nofriansyah ; 2014 : 1).

Definisi mengenai sistem pendukung keputusan (SPK) yang ideal yaitu :

- a. SPK adalah sebuah sistem berbasis komputer dengan antarmuka antara mesin/komputer dan pengguna.
- b. SPK ditujukan untuk membantu pembuat keputusan dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk mengganti posisi manusia sebagai pembuat keputusan.
- c. SPK mampu memberi alternatif solusi bagi masalah semi/tidak terstruktur baik bagi perseorangan atau kelompok dan dalam berbagai macam proses dan gaya pengambilan keputusan.
- d. SPK menggunakan data, basis data dan analisa model-model keputusan.
- e. SPK bersifat adaptif, efektif, interaktif , *easy to use* dan fleksibel
- f. SPK menyediakan akses terhadap berbagai macam format dan tipe sumber data (*data source*) (Heri Sulistiyo : 2012 : 2).

II.1.1. Karakteristik Sistem Pendukung Keputusan

Karakteristik dan kapabilitas sistem pendukung keputusan antara lain :

1. Dukungan untuk pengambilan keputusan, terutama pada situasi semi terstruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi. Masalah-masalah tersebut tidak dapat dipecahkan oleh sistem komputer lain atau oleh metode atau alat kuantitatif standar.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok. Masalah yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasional yang berbeda atau bahkan dari organisasi lain. DSS mendukung im virtual melalui alat-alat Web kolaboratif.
4. Dukungan untuk keputusan independen dan atau sekuensial. Keputusan dapat dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
5. Dukungan disemua fase proses pengambilan keputusan : intelegensi, desain, pilihan, dan implementasi
6. Dukungan diberbagai proses dan gaya pengambilan keputusan.
7. Adaptivitas sepanjang waktu. Pengambil keputusan seharusnya reaktif, dapat menghadapi perubahan kondisi secara cepat, dan dapat mengadaptasi DSS untuk memenuhi perubahan tersebut. DSS bersifat fleksibel dan karena itu pengguna dapat menambahkan, menghapus, menggabungkan, mengubah, atau menyusun kembali elemen-elemen dasar, DSS juga fleksibel dalam hal dapat dimodifikasi untuk memecahkan masalah lain yang sejenis.

8. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timeliness*, kualitas).
9. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah. DSS secara khusus menekankan untuk mendukung pengambil keputusan, bukannya menggantikan (Sri Hartati ; 2011 : 37).

II.1.2. Komponen Sistem Pendukung Keputusan

Sistem Pendukung Keputusan disusun dari beberapa subsistem yaitu :

1. Subsistem manajemen data
Basis data yang relevan dan dikelola menggunakan *software* yang disebut *database management system* (DBMS)
2. Subsistem manajemen model
Adalah paket *software* yang berisi model-model yang disebut dengan *modelbase management system* (MBMS)
3. Subsistem manajemen pengetahuan
Subsistem yang memberikan intelegensi dan mendukung subsistem yang lain.
4. Subsistem antarmuka pengguna
Pengguna berkomunikasi dan memerintah SPK melalui system ini.
5. Pengguna
Orang yang berhadapan dengan pengambil keputusan (Sri Hartati ; 2011 : 37).

II.2. Supervisor

Supervisor adalah jabatan yang sangat penting karena menjembatani dua kepentingan, yaitu manajemen dan pekerja operator. Sangat unik, karena posisinya di antara dua elemen penting dalam organisasi, yaitu manajemen dan kepemimpinan. Supervisor adalah posisi yang unik dalam organisasi. Supervisor dikenal sebagai pihak yang mewakili manajemen dalam berhubungan dengan para karyawan “kerah biru”. Tapi, di banyak perusahaan, supervisor justru bagian dari kerah biru alih-alih menjadi bagian “kerah putih”. Jadi, seperti tidak diakui sebagai kerah putih (*secara de facto*), lalu secara *de jure* juga tidak diakui sebagai kerah biru. Betapa sulitnya seorang supervisor memosisikan dirinya. Belum lagi tanggung jawab yang harus diimbang mengenai sumber daya manusia. Seorang supervisor (pada umumnya) harus mampu membawahi lebih dari 15 orang, tiga seksi kerja dengan puluhan mesin dan proses (Wawang Sukmoro ; 2010 : 11).

II.3. Pemilihan Supervisor

Pemilihan karyawan yang berkualitas dan layak sesuai kebutuhan perusahaan merupakan salah satu cara yang digunakan untuk menyeleksi dan mempromosikan karyawan untuk naik jabatan menjadi Supervisor. Seleksi ini digunakan untuk menilai kemampuan calon Supervisor secara objektif bukan subjektif. Penelitian ini mengambil data dari PT Asta Anugrah Medan berupa nilai bobot setiap atribut seperti aspek intelektual, aspek sikap kerja dan aspek perilaku kemudian dilakukan proses perangkingan yang akan menentukan alternatif yang optimal yaitu karyawan yang layak menjadi supervisor (Kardiawan Lius Sarumaha : 2014 : 63).

II.4. Metode *Naive Bayes*

Metode *Naive Bayes* merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. *Naive Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya sehingga dikenal sebagai Teorema Bayes. Teorema tersebut dikombinasikan dengan *Naive* dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi *Naive Bayes* diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya.

Persamaan dari teorema *Bayes* adalah :

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \dots\dots\dots(1)$$

Keterangan :

- X : Data dengan class yang belum diketahui
- H : Hipotesis data merupakan suatu class spesifik
- P (H|X) : Probabilitas hipotesis berdasar kondisi (posteriori probability)
- P (H) : Probabilitas hipotesis (prior probability)
- P (X|H) : Probabilitas berdasarkan kondisi pada hipotesis
- P (X) : Probabilitas

Untuk menjelaskan teorema *Naive Bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, teorema *bayes* di atas disesuaikan sebagai berikut :

$$P(C|F_1...F_n) = \frac{P(C)P(F_1...F_n|C)}{P(F_1...F_n)} \dots \dots \dots (2)$$

Keterangan :

Variabel C : merepresentasikan kelas

variabel $F_1...F_n$: merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi.

Dimana Variabel C merepresentasikan kelas, sementara variabel $F_1 \dots F_n$ merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (*Posterior*) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik karakteristik sampel pada kelas C (disebut juga *likelihood*), dibagi dengan peluang kemunculan karakteristik karakteristik sampel secara global (disebut juga *evidence*). Karena itu, rumus diatas dapat pula ditulis secara sederhana sebagai berikut :

$$Posterior = \frac{Prior \times likelihood}{evidence}$$

Keterangan :

Posterior : perbaikan terhadap nilai probabilitas

Prior : nilai probabilitas awal

Likehood : titik tertentu untuk memaksimumkan sebuah fungsi.

Evidence : nilai bukti

Nilai *Evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari *posterior* tersebut nantinya akan dibandingkan dengan nilai nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut rumus *Bayes* tersebut dilakukan dengan menjabarkan menggunakan aturan perkalian sebagai berikut :

$$\begin{aligned}
 P(C/F_1 \dots F_n) &= P(C)P(F_1 \dots F_n/C) \\
 &= P(C)P(F_1/C)P(F_2 \dots F_n/C, F_1) \\
 &= P(C)P(F_1/C)P(F_2/C, F_1)(F_1/C)P(F_3 \dots F_n/C, F_1, F_{n-1}) \dots \dots \dots (3)
 \end{aligned}$$

Dapat dilihat bahwa hasil penjabaran tersebut menyebabkan semakin banyak dan semakin kompleksnya faktor faktor syarat yang mempengaruhi nilai probabilitas, yang hampir mustahil untuk dianalisa satu persatu. Akibatnya, perhitungan tersebut menjadi sulit untuk dilakukan. Disinilah digunakan asumsi independensi yang sangat tinggi (*naif*), bahwa masing masing petunjuk saling bebas (*independen*) satu sama lain (Bustami : 2014 : 129).

II.5. Pengertian Visual Basic

VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada platform.NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer Visual Basic. Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas platform Windows. Fitur baru bahasa VB.NET dibandingkan

Visual Basic bahwa bahasa VB.NET mendukung object-oriented dan juga dynamics programming. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET (Agus Kurniawan ; 2013 : 10).

II.6. Pengertian Database

Database adalah sekumpulan data mentah yang disusun menurut logika tertentu dan terorganisasi dalam bentuk yang dapat disimpan dan diproses oleh komputer. contoh *database* dapat berisi data pegawai, data penjualan, pembayaran, dan lain-lain. data internal dari akunting, keuangan, penjualan dan bidang-bidang bisnis lainnya yang disimpan dalam suatu sistem komputer dan disusun menurut logika tertentu disebut sebagai internal *database*. *Database* seringkali disimpan dalam suatu perangkat tertentu pada komputer, seperti hard disk, compact disk, dan sebagainya. hubungan antarsistem *database* dan sistem software sangat kuat karena sistem *database* yang dipakai sangat menentukan kemudahan aksesnya data sementara software sendiri memungkinkan peneliti memanipulasi data untuk dianalisis (Dermawan Wibisono ; 2012 : 129).

II.6.1. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional. Teori normalisasi didasarkan pada konsep bentuk normal.

Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.6.1.1. Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1st Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

II.6. Pengertian SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. SQL Server adalah 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. Microsoft merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju (Wenny Widya ; 2012 : 3)


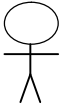

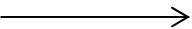

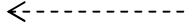
II.7. UML (*Unified Modeling Language*)

Menurut Evi Triandini (2012) UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan disain. Artifak di dalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan disain adalah bahwa setiap perubahan yang terjadi pada satu artifak harus juga dilakukan pada artifak sebelumnya.

1. *Use Case Diagram*

Use case adalah sebuah kegiatan yang dilakukan oleh sistem, biasanya dalam menanggapi permintaan dari pengguna sistem.

Tabel II.1. Simbol *Use Case*




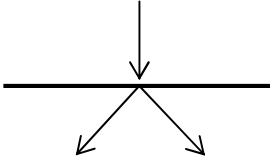
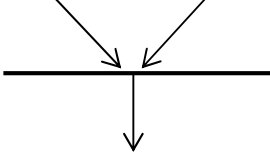
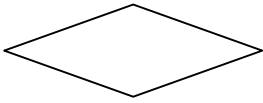

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<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.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

2. 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.

Tabel II.2. Simbol *Activity Diagram*

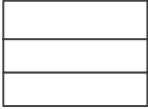



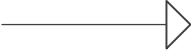
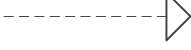

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
 New Swimline	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

3. Class Diagram

Class diagram bertujuan untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun.

Tabel II.3. Simbol Class Diagram

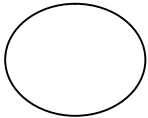
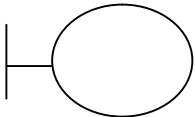


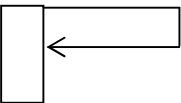


Gambar	Keterangan
	<i>Class</i> , Menambahkan kelas baru pada diagram
	<i>Interface</i> , Menambahkan kelas antarmuka (<i>interface</i>) pada diagram
	Association, Menggambar relasi asosiasi
	<i>Association class</i> , Menghubungkan kelas asosiasi (<i>association class</i>) pada suatu relasi asosiasi
	<i>Generalization</i> , Menggambarkan relasi generalisasi
	<i>Realize</i> , Menggambarkan relasi realisasi
	<i>Aggregation</i> , Menggambarkan relasi agregasi

(Sumber : Windu Gata ; 2013 : 9)

4. *Sequence Diagram*

Sequence diagram adalah diagram yang digunakan untuk mendefinisikan *input* dan *output* serta urutan interaksi antara pengguna dan sistem untuk sebuah *usecase*.

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)

Tabel II.5. 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

(Sumber : Windu Gata ; 2013 : 9)