

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Pengertian Sistem Pendukung Keputusan**

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:1), Sistem pendukung keputusan (SPK) dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi Sistem pendukung keputusan (SPK) digunakan untuk pengambilan keputusan. Aplikasi Sistem pendukung keputusan (SPK) menggunakan *CBIS (Computer Based Information System)* yang fleksibel, interaktif dan dapat diadaptasi yang dikembangkan untuk mendukung solusi masalah manajemen spesifik yang tidak terstruktur.

Menurut Bonczek dkk, (1980) dalam buku "*Decission Support System and intelligent system* (Turban 2005:137) mendefinisikan Sistem pendukung keputusan (SPK) sebagai system berbasis computer yang terdiri dari tiga komponen yang saling berinteraksi, system bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen system pendukung keputusan lain), system pengetahuan (repository pengetahuan domain masalah yang ada pada Sistem pendukung keputusan (SPK) atau sebagai data atau sebagai prosedur), dan system pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

### **II.1.1 Karakteristik Sistem Pendukung Keputusan**

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:2), Karakteristik Sistem pendukung keputusan (SPK) yaitu :

- a. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan
- b. Adanya interface manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
- c. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
- d. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
- e. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

### **II.1.2 Ciri-Ciri Sistem Pendukung Keputusan**

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:2), Kriteria atau ciri-ciri sistem pendukung keputusan adalah sebagai berikut :

- a. Banyak pilihan/alternatif
- b. Ada kendala atau surat
- c. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
- d. Banyak input/Variabel
- e. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan .

### II.1.3 Fase Dalam Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:2), Tiga fase dalam pengambilan keputusan yaitu :

#### a. *Intelligence*

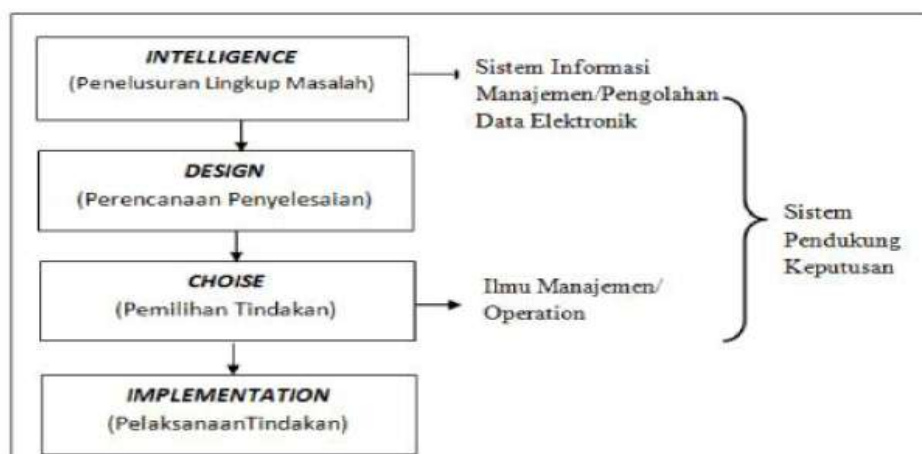
Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangka mengidentifikasi masalah.

#### b. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap melakukan pengujian kelayakan solusi.

#### c. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan kedalam proses pengambilan keputusan.



Gambar II.1 Fase Proses Pengambilan Keputusan

(Sumber :Dicky Nofriansyah, S.Kom M.Kom , konsep data mining vs sistem pendukung keputusan : 2014)

#### II.1.4 Komponen Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:3), Secara garis besar sistem pendukung keputusan dibangun oleh tiga komponen utama yaitu :

a. Sub Sistem Data (*Database*)

Sub sistem data merupakan komponen sistem pendukung keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan untuk diorganisasikan dalam sebuah basis data yang diorganisasikan oleh suatu sistem yang disebut dengan Sistem Manajemen Sistem Basis Data (*Database Management System*)

b. Sub sistem Model

Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang model adalah bahwa model yang dirancang tidak mampu mencerminkan seluruh variabel alam nyata, sehingga keputusan yang diambil tidak sesuai dengan kebutuhan. Oleh karena itu, dalam menyimpan berbagai model harus diperhatikan dan harus dijaga fleksibilitasnya. Hal lain yang harus diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian, keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

c. Sub sistem dialog (*User System Interface*)

Sub sistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif, yang dikenal dengan sub sistem dialog. Melalui sub sistem dialog sistem diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat.

### **II.1.5 Tujuan Sistem Pendukung Keputusan**

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:4), Tujuan dari system pendukung keputusan adalah sebaagi berikut :

- a. Membantu dalam pengambilan keputusan atas masalah yang terstruktur
- b. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
- c. Meningkatkan efektifitas keputusan yang diambil lebih dari perbaikan efesiensinya.
- d. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat dengan biaya yang sangat rendah.
- e. Peningkatan produktifitas membangun suatu kelompok pengambil keputusan, terutama para pakar bisa sangat mahal. Sistem pendukung keputusan komputerisasi mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yang berbeda-beda (menghemat biaya perjalanan). Selain itu produktifitas staf pendukung (misalnya analisis keuangan dan hokum) bisa ditingkatkan. Produktivitas juga ditingkatkan menggunakan peralatan optimalisasi yang menjalankan sebuah bisnis.

### **II.2. Metode *Analytical Hierarchi Process (AHP)***

Menurut Astri Herdiyanti dkk (2013:50), Dalam Jurnal Jurnal Ilmiah Komputer dan Informatika dengan judul “Pembangunan Sistem Pendukung Keputusan Rekrutmen Pegawai Baru di PT. ABC”, Metode AHP atau Proses Hirarki Analitik merupakan salah satu metode pengambilan keputusan dimana

faktor-faktor logika, intuisi, pengalaman, pengetahuan, emosi, dan rasa dicoba untuk dioptimalkan dalam suatu proses yang sistematis. Sistem yang dibangun didukung oleh metode AHP. Adapun langkah-langkah dalam perhitungan AHP adalah sebagai berikut:

- a. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
- b. Membuat struktur hierarki yang diawali dengan tujuan utama.
- c. Menjumlahkan setiap kolom ( $\Sigma$ kolom) pada matriks perbandingan.
- d. Normalisasi matriks, dengan membagi setiap kolom matriks dengan jumlah kolom ( $\Sigma$ kolom), kemudian dijumlahkan setiap barisnya ( $\Sigma$ baris).
- e. Menghitung total priority value (TPV) untuk mendapatkan bobot subkriteria.
- f. Menghitung uji konsistensi

Tahapan dalam melakukan uji konsistensi adalah sebagai berikut:

- a. Mengalikan nilai TPV dengan nilai kolom matriks pada nilai matriks perbandingan kemudian jumlahkan tiap barisnya.
- b. Mencari *consistency index* (CI) dengan rumus:

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

Dimana :

CI = *Consistency Index*

N = banyaknya elemen yang dibandingkan

$\lambda \max$  = *Eigen value maksimum*

c. Mencari Consistency Ratio (CR) dengan rumus:

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

Dimana :

CR = *Consistency Ratio*

CI = *Consistency Index*

RI = *Random Index*

Kriteria dan alternatif dinilai melalui perbandingan berpasangan. Menurut Thomas L. Saaty, untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik dalam mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty dapat dilihat pada table berikut :

**Tabel II.1 Skala Perbandingan**

Intensitas Kepentingan	Definisi
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
4	Elemen yang satu lebih penting daripada yang lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan

(sumber : *Jurnal Ilmiah Komputer dan Informatika KOMPUTA :2013*)

Sebuah nilai rasio dikatakan konsisten jika bernilai  $0 \leq \text{rasio} \leq 0.1$ , dengan demikian hasil perhitungan data dapat dibenarkan. Untuk menentukan rasio konsisten atau tidak dapat menggunakan tabel konsistensi rasio sebagai berikut :

**Tabel II.2 Konsistensi Rasio**

<i>N</i>	<i>RI</i>
1	0.00
2	0.00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
$\geq 15$	1.59

(sumber : *Jurnal Ilmiah Komputer dan Informatika KOMPUTA* :2013)

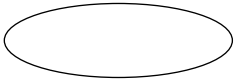
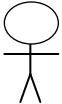


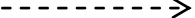
### II.3.1. UML (*Unified Modeling Language*)

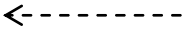
Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

## 1. Use case Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram,

**Tabel II.3. 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 mengidikasi aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidikasi 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




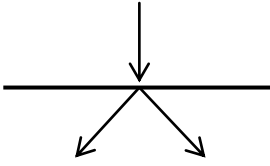
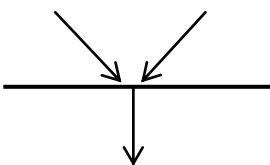
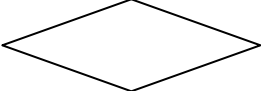

	program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

## 2. Diagram Aktivitas (*Activity Diagram*)

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

**Tabel II.4. Simbol *Activity Diagram***

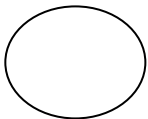
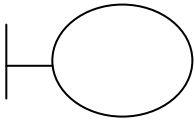
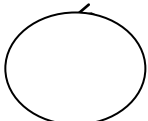
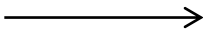
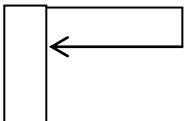


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel 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> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

### 3. Diagram Urutan (*Sequence Diagram*)

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

**Tabel II.5. 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)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.6. *Multiplicity Class Diagram***

<b>Multiplicity</b>	<b>Penjelasan</b>
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)

#### II.4. Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulan-kumpulan *file* yang saling berkaitan.

Menurut Kusrini (2010, p2), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol).

Basis data dapat didefinisikan dalam berbagai susut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

#### **II.4.1. Tujuan Basis Data**

Menurut Kusrini (2010, p2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

- a. Tidak adanya redudansi dan inkonsistensi data

Redudansis terjadi jika suatu informasi disimpan dibeberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya

terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

b. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan query ataupun dari tool yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari software DBMnnya.

c. *Multiple user*

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian server yang bisa diakses dari banyak *client*, sudah menyediakan akses kesemua pengguna dari komputer *client* ke sumber informasi yaitu basis data.

#### **II.4.2. Manfaat/Kelebihan Basis Data**

Menurut Kusrini (2010, p5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

a. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan metematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

b. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

c. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

d. Efisiensi ruang penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

e. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.

f. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

g. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang diaksesnya.

h. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur *interface* untuk pengguna.

i. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan oleh DBMS.

j. Kebebasan data

Perubahan dapat dilakukan pada *level* DBMS tanpa harus membongkar kembali program aplikasinya.

k. *User View*

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

### II.4.3. Operasi Dasar Database

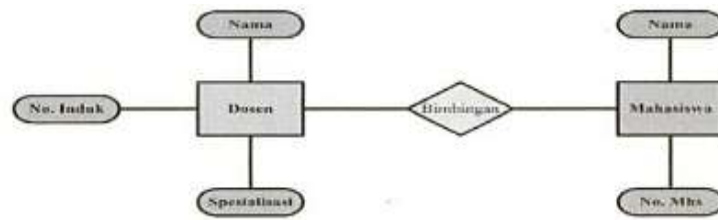
Menurut Kusrini (2010, p9), Beberapa operasi dasar basis data yaitu :

- a. Pembuatan basis data
- b. Penghapusan basis data
- c. Pembuatan file/tabel
- d. Penghapusan file/tabel
- e. Pengubahan tabel
- f. Penambahan/pengisian
- g. Pengambilan data
- h. Penghapusan data

### II.4.4. Pemodelan Basis Data

Menurut Samiaji Sarosa (2010:4), Model diperlukan untuk mendapatkan penyederhanaan dari kenyataan dan memungkinkan desainer program program aplikasi bereksperimen dengan berbagai macam variable sebelum diaplikasikan ke system yang berjalan. Untuk merancang suatu aplikasi basis data alat yang biasa digunakan adalah *Entity Relationship Diagram (ERD)*. ERD didasarkan dari artikel yang dipublikasikan oleh Peter Phin Shan Chen.

Ada beberapa case tool menamakan notasi ERD yang digunakan sebagai chen ERD. *Entity Relationship Model* adalah abstraksi konseptual yang mewakili struktur dari suatu basis data.



**Gambar II.2. Diagram Dengan Notasi Chen ERD**

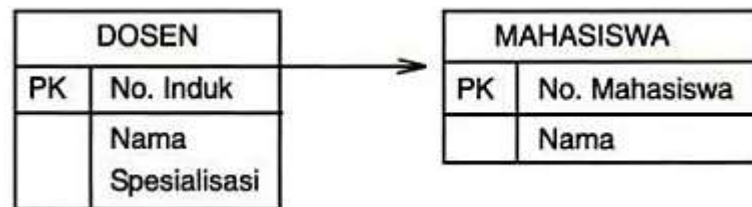
(sumber : Samiaji Sarosa , Sistem Informasi Akutansi :2010)

Dalam perkembangannya banyak diciptakan notasi ERD yang berbeda-beda seperti terlihat gambar dibawah ini.



**Gambar II.3. Diagram Dengan Notasi Crows Foot**

(sumber : Samiaji Sarosa , Sistem Informasi Akutansi :2010)



**Gambar II.4. Diagram Dengan Notasi Relational**

(sumber : Samiaji Sarosa , Sistem Informasi Akutansi :2010)

#### II.4.5. Normalisasi

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan duplikasi data dan menghindarkan basis data tersebut anomali. Suatu basis data dikatakan tidak normal jika terjadi 3 (tiga) anomali berikut :

a. *Insertion Anomaly*

Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam table.

b. *Update/Modification anomaly*

Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

Langkah-langkah normalisasi sampai pada bentuk 3NF adalah sebagai berikut :

a. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau *field* yang berulang. Syarat kedua harus ada *primary key (PK)* atau kunci unik, atau kunci yang membedakan satu baris dengan baris yang lain dalam satu table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

b. *Second Normal Form (2NF)*

Untuk menjadi 2NF suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *Partial dependencies* adalah suatu kondisi jika atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

c. *Third Normal Form (3NF)*

Untuk menjadi 3NF suatu table harus berada dalam kondisi 2NF dan tidak memiliki *transitive dependencies*. *Transitive dependencies* adalah suatu kondisi dengan adanya ketergantungan fungsional antara 2 atau lebih atribut non kunci (Non PK).

## II.5. Visual Basic 2010

Menurut Edi Winarno ST, M.Eng dkk (2010:1), Visual Basic adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer didunia. Pemograman ini dipakai oleh jutaan programmer dan tercatat sebagai program yang paling disukai oleh mayoritas orang.

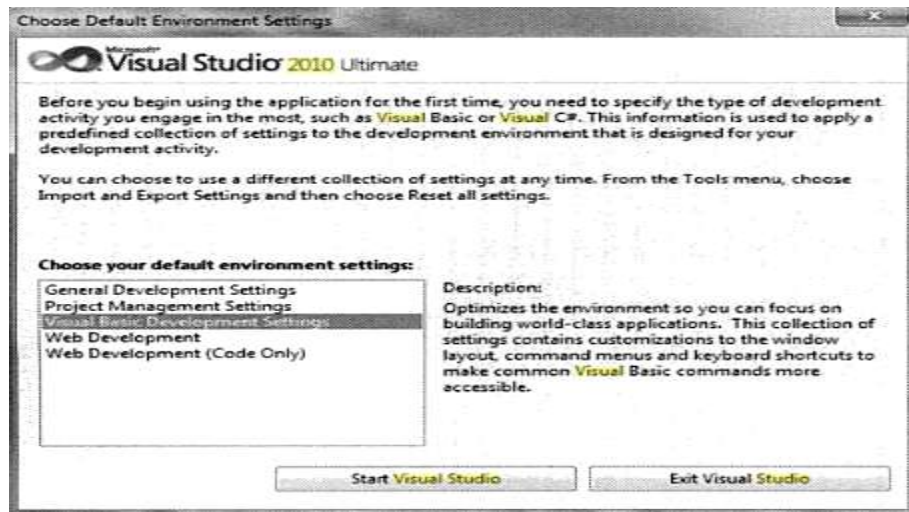
Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering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

Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

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.

### II.5.1. Antar Muka Visual Basic 2010

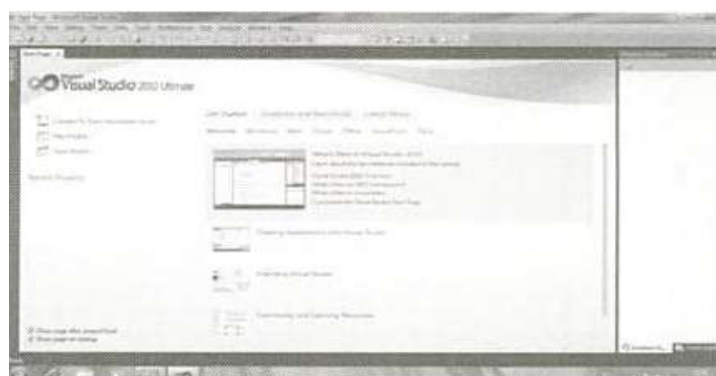
Saat menjalankan Visual Basic 2010 pertama kali muncul jendela *chosedefault environtment settings*. Disini bisa memilih apakah ingin memilih antar muka di Visual Studio. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre*.



**Gambar II.5 Form Chose Default Environment Settings**

*(sumber : Edi Winarno ST , Dasar-Dasar Pemograman Dengan Visual Basic 2010 :2010)*

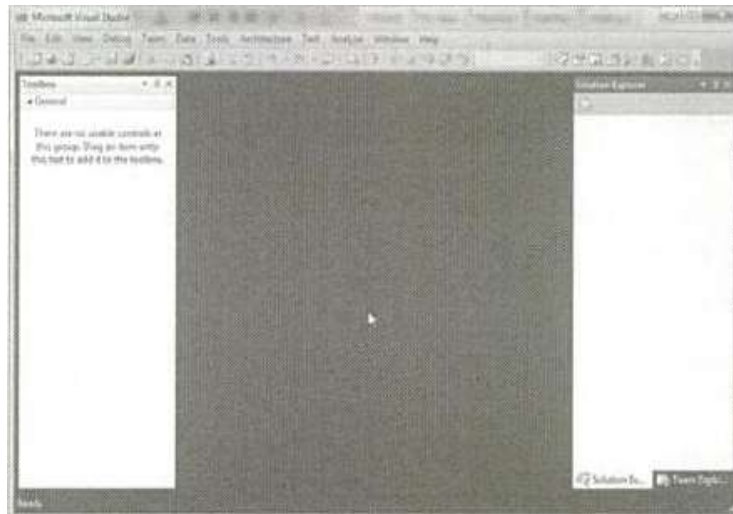
Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



**Gambar II.6. Start Page Visual Basic 2010**

*(sumber : Edi Winarno ST , Dasar-Dasar Pemograman Dengan Visual Basic 2010 :2010)*

Jika *start page* ditutup terlihat tampilan sebagai berikut :



**Gambar II.7. Tampilan IDE (*Integrated Development Environment*) setelah *Start Page* ditutup**

(sumber : *Edi Winarno ST , Dasar-Dasar Pemograman Dengan Visual Basic*

*2010 :2010*)



**Gambar II.8. Tampilan lengkap IDE**

(sumber : *Edi Winarno ST , Dasar-Dasar Pemograman Dengan Visual Basic*

*2010 :2010*)

Komponen-komponen dari IDE adalah :

1. Dibagian kiri terdapat *toolbox* yang menampilkan semua objek tool yang bisa dimasukkan kedalam *form* untuk membuat program.
2. Dibagian tengah terdapat tempat meletakkan form dan kode, baik disaat desain ataupun pada saat program dijalankan.
3. Dibagian kanan terdapat *solution explorer* yang merupakan explorer untuk melihat *file-file* disebuah objek.
4. Dikanan bawah terdapat propertis untuk melihat properti dari nilai-nilai pada objek yang dipilih dibagian tengah. (Edi Winarno ST, M.Eng dkk, 2010:1)

## **II.6. SQL Server 2008 *Express Edition***

Menurut Wahana Komputer (2010:2), SQL Server 2008 *Express Edition* sebuah terobosan baru dalam bidang database, SQL Server 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 *Express Edition* dibuat pada saat kemajuan dalam bidang hardware semakin pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 *Express Edition* membawa terobosan dalam bidang pengolahan dan penyimpanan data.

### **II.6.1 Kebutuhan *Hardware***

Adapun *hardware* yang diperlukan untuk instalasi SQL Server 2008 *Express Edition* minimal adalah sebagai berikut :

- a. Prosescor minimal 1 GHz
- b. Memori minimal 512 MB
- c. Sistem Operasi Windows

Biar dapat diinstal pada system computer dengan memoti 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah :

- a. *Sharer Memory*
- b. TCP/IP
- c. *Named Pipes*
- d. *Virtual Interface Adapter (VIA)*(Wahana Komputer, 2010:2).

### **II.6.2 Versi SQL Server 2008 *Express Edition***

Microsoft merilis SQL Server 2008 *Express Edition* dalam beberapa versi yang disesuaikan dengan segmen-segmen pasar yang dituju. Versi-versi tersebut adalag sebagai berikut :

- a. Menurut cara pemrosesan data pada prosesor makan Microsoft mengelompokkan produk ini berdasarkan dua jenis yaitu :
  1. Versi 32 Bit (x86), yang biasanya digunakan untuk komputer *single processor* (Pentium 4) atau lebih tepatnya processor 32 bit atau Windows XP
  2. Versi 64 Bit (x64), yang biasanya digunakan oleh computer yang lebih sari satu processor (Misalnya *Core 2 duo*) dan system operasi 64 bit, Vista dan Windows 7.

- b. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
1. Versi *Compact* ini adalah versi “tipis” dari semua versi yang ada
  2. Versi *Express* ini adalah versi “ringan”

### II.6.3 Instalasi SQL Server 2008 *Express Edition*

Proses instalasi SQL Server 2008 *Express Edition* tidak sama dengan instalasi versi-versi sebelumnya. Proses SQL Server 2008 *Express Edition* agak panjang melalui beberapa tahapan. Tahapan yang dilakukan akan membawa beberapa pilihan yang akan diisi dalam *setting* sebuah *server database*. Berikut ini adalah pilihan-pilihan yang akan dijumpai dalam proses instalasi SQL Server 2008 *Express Edition*.

1. Tempat direktori utama dan penyimpanan file database

Direktori utama adalah direktori dimana semua file program akan ditempatkan dan file-file tersebut tidak akan berubah selama anda menjalankan SQL server. Direktori utama secara standard akan berada dalam direktori “C:\Program Files\Microsoft SQL Server”.

2. Penggunaan *Multiple instance*

*Instance* adalah sebuah turunan dari server database SQL Server. Karena sebuah tiruan maka sebuah *Instance* memiliki fungsi yang sama dengan database server aslinya. Arti sebenarnya *Instance SQL Server* adalah sebuah server database yang tidak *sharing* sistemnya dan database user dengan database server lainnya yang ada dalam komputer yang sama.

3. *Jasa Autentification User* (Menggunakan Windows atau mixed)

*Autentification User* diperlukan supaya server tidak dapat dipergunakan oleh orang yang tidak bertanggungjawab dan tidak berhak. Dalam SQL server ada dua *Autentification User* yang dapat digunakan yaitu :

- a. Mode Windows, Pada mode ini SQL Server akan melakukan autentifikasi dengan menggunakan level login pada system operasi.
- b. Mode Mixel atau campuran, mode ini menginjinkan *user* untuk masuk kedalam system SQL server dengan menggunakan *Account* yang dibuat di system operasi windows atau juga menggunakan *account* yang di *set up* pada SQL Server (Wahana Komputer, 2010:2)