

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Sistem Pendukung Keputusan atau *Decision Support System* yang selanjutnya kita singkat dalam Skripsi ini menjadi SPK, secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik, kemampuan pemecahan masalah, maupun kemampuan pengkomunikasian untuk masalah semi-terstruktur. Secara khusus, SPK didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manajer dalam memecahkan masalah semi-terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu.

Menurut Kusri (2007, h. 15) Sistem Pendukung Keputusan atau *Decision Support System* (DSS) “merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Dan sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat”.

Tujuan dari sistem pendukung keputusan antara lain :

1. Membantu pegawai dalam pengambil keputusan atas masalah semiterstruktur.
2. Memberikan dukungan atas pertimbangan pegawai dan bukannya dimaksudkan untuk menggantikan fungsi pegawai.

3. Meningkatkan efektifitas keputusan yang diambil pegawai lebih dari pada perbaikan efisiensinya.
4. Meningkatkan kecepatan komputasi, untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktifitas, produktifitas biasa ditingkatkan menggunakan peralatan optimalisasi yang menentukan cara terbaik untuk menjalankan sebuah bisnis.
6. Memberikan dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat. Dengan komputer menilai berbagai pengaruh secara cepat dan ekonomis.
7. Meningkatkan daya saing. Teknologi pengambil keputusan bisa menciptakan pemberdayaan dengan cara memperbolehkan seorang untuk membuat keputusan yang baik dan tepat.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

Pembuatan keputusan merupakan fungsi utama seorang manajer atau administrator. Kegiatan pembuatan keputusan meliputi pengidentifikasian masalah, pencarian alternatif penyelesaian masalah, evaluasi dari alternatif-alternatif tersebut dan pemilihan alternatif keputusan yang baik. Kemampuan seorang manajer dalam membuat keputusan dapat ditingkatkan apabila ia mengetahui dan menguasai teori dan teknik pembuatan keputusan. Dengan peningkatan kemampuan manajer dalam pembuatan keputusan diharapkan dapat

ditingkatkan kualitas keputusan yang dibuatnya, dan hal ini tentu akan meningkatkan efisiensi kerja manajer yang bersangkutan.

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

Dengan berbagai karakter khusus yang dimiliki Sistem Pendukung Keputusan, SPK (Sistem Pendukung Keputusan) dapat memberikan berbagai manfaat dan keuntungan. Keuntungan yang dapat diambil dari SPK adalah:

1. Mampu mendukung pencarian solusi dari masalah yang kompleks.
2. Respon cepat pada situasi yang tak diharapkan dalam kondisi yang berubah-ubah.
3. Mampu untuk menerapkan berbagai strategi yang berbeda pada konfigurasi berbeda secara cepat dan tepat.
4. Pandangan dan pembelajaran baru.
5. Mempasialisasi komunikasi.
6. Meningkatkan control manajemen dan kinerja.
7. Menghemat biaya.
8. Keputusan lebih cepat.
9. Meningkatkan produktifitas analisis.

### **II.2. Konsep dasar Multi Attribute Decision Making (MADM)**

MADM menyelesaikan masalah-masalah dalam ruang diskret. Oleh karena itu, pada MADM biasanya digunakan untuk melakukan penilaian atau seleksi terhadap beberapa alternatif dalam jumlah yang terbatas.

Menurut Kusumadewi dkk. (2006, h. 72), Pada dasarnya, proses MADM dilakukan melalui 3 tahap, yaitu penyusunan komponen-komponen situasi, analisis, dan sintesis informasi. Pada tahap penyusunan komponen, komponen situasi, akan dibentuk tabel taksiran yang berisi identifikasi alternatif dan spesifikasi tujuan, kriteria dan atribut.

*Multi Attribute Decision Making* (MADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Tahap analisis dilakukan melalui 2 langkah. Pertama, mendapatkan taksiran dari besaran yang potensial, kemungkinan dan ketidakpastian yang berhubungan dengan dampak-dampak yang mungkin pada setiap alternatif. Kedua, meliputi pemilihan dari preferensi pengambil keputusan untuk setiap nilai, dan ketidakpedulian terhadap resiko yang timbul. Pada langkah pertama, beberapa metode menggunakan fungsi distribusi  $|p_j(x)|$  yang menyatakan probabilitas kumpulan atribut  $|a_k|$  terhadap setiap alternatif  $|A_i|$ . Konsekuensi juga dapat ditentukan secara langsung dari agregasi sederhana yang dilakukan pada informasi terbaik yang tersedia. Demikian pula, ada beberapa cara untuk menentukan preferensi pengambil keputusan pada setiap konsekuensi yang dapat dilakukan pada langkah kedua. Metode yang paling sederhana adalah untuk menurunkan bobot atribut dan kriteria adalah dengan fungsi utilitas atau penjumlahan terbobot.

Dalam Kusumadewi dkk. (2006, h. 72), Zimmermann menyatakan bahwa, secara umum, model *multi attribute decision making* dapat didefinisikan sebagai berikut:

Misalkan  $A = \{a_i \mid i = 1, \dots, n\}$  adalah himpunan alternatif keputusan dan  $C = \{c_j \mid j = 1, \dots, m\}$  adalah himpunan tujuan yang diharapkan, maka akan ditentukan alternatif  $x^0$  yang memiliki derajat harapan tertinggi terhadap tujuan-tujuan yang relevan  $c_j$ .

Namun sebagian besar pendekatan MADM dilakukan melalui 2 langkah, yaitu : pertama, melakukan agregasi terhadap keputusan-keputusan yang tanggap terhadap semua tujuan pada setiap alternatif. Sedangkan yang kedua, melakukan perankingan alternatif-alternatif keputusan tersebut berdasarkan hasil agregasi keputusan.

Dengan demikian, bisa dikatakan bahwa, masalah *multi-attribute decision making* (MADM) adalah mengevaluasi  $m$  alternatif  $A_i$  ( $i=1,2,\dots,m$ ) terhadap sekumpulan atribut atau kriteria  $C_j$  ( $j=1,2,\dots,n$ ), dimana setiap atribut saling tidak bergantung satu dengan yang lainnya. Matriks keputusan setiap alternatif terhadap setiap atribut  $x$ , diberikan sebagai:

$$x = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}$$

Dimana  $x_{ij}$  merupakan rating kinerja alternatif ke- $i$  terhadap atribut ke- $j$ .

Nilai bobot yang menunjukkan tingkat kepentingan relatif setiap atribut, diberikan sebagai,  $w$  :

$$w = \{ w_1, w_2, \dots, w_n \}$$

Rating kinerja ( $x$ ), dan nilai bobot ( $w$ ) merupakan nilai utama yang merepresentasikan preferensi absolut dari pengambil keputusan/ masalah MADM

diakhiri dengan proses perankingan untuk mendapatkan alternatif terbaik yang diperoleh berdasarkan nilai keseluruhan preferensi yang diberikan (Yeh dalam Kusumadewi dkk. (2006, h. 73).

Beberapa metode yang dapat digunakan untuk menyelesaikan masalah MADM, antara lain sebagai berikut :

1. *Simple Additive Weighting Method (SAW)*
2. *Weighted Product Model (WPM)*
3. *ELECTRE*
4. *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)*
5. *Analytic Hierarchy Process (AHP)*

### **II.3. Pengertian *Simple Additive weighting (SAW)***

Metode *Simple Additive Weighting (SAW)* sering juga dikenal dengan istilah penjumlahan terbobot, metode yang paling simpel dan masih banyak digunakan pada metode MADM, Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Menurut (Fishburn, 1967) (MacCrimmon, 1968).

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode SAW mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap

atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya.

Pada metode SAW, ada kriteria yang dipersepsikan sebagai kriteria "benefit" dan "cost". Dimana kriteria "benefit" digunakan jika kriteria lebih baik ketika bernilai besar, sedangkan "cost" kriteria akan lebih baik ketika nilainya lebih kecil. Besar dan kecilnya nilai tersebut dilihat dari keterkaitannya dengan permasalahan yang diangkat.

### II.3.1 Tahapan SAW :

1. Menentukan kriteria-kriteria dan bobot yang akan dijadikan acuan dalam pengambilan keputusan, yaitu  $C_i$ .
2. Normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.
3. Menentukan kriteria *Cost* ataupun *Benefit*.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik ( $A_i$ ) sebagai solusi.

Berikut formulasi yang digunakan melakukan normalisasi (*Benefit*) :

$$r_{ij} = \frac{x_{ij}}{\text{Max } x_{ij}}$$

Berikut formulasi yang digunakan melakukan normalisasi (*Cost*) :

$$r_{ij} = \frac{\text{Min}_i x_{ij}}{x_{ij}}$$

Keterangan :

$r_{ij}$  = rating kinerja ternormalisasi dari alternatif  $A_i$  pada atribut  $C_j$

$i$  = 1,2,3,..., m

$j$  = 1,2,3,...n

$Maxx_{ij}$  = nilai maksimum dari setiap baris dan kolom

$Minx_{ij}$  = nilai minimum dari setiap baris dan kolom

$x_{ij}$  = baris dan kolom dari matriks

Nilai preferensi untuk setiap alternatif ( $V_i$ ) :

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Dimana :

$V_i$  = Nilai akhir dari alternatif

$w_j$  = Bobot yang telah ditentukan

$r_{ij}$  = Normalisasi matriks

### II.3.2 Algoritma Penyelesaian

Dalam penelitian ini menggunakan metode SAW, adapun langkah-langkah penyelesaiannya adalah :

1. Memberikan nilai setiap alternatif (cara menghitung nilai rating kerja ternormalisasi ( $r_{ij}$ ) dari alternatif  $A_i$  pada setiap criteria ( $C_j$ ) yang sudah ditentukan, dimana nilai  $i = 1,2,\dots,m$  dan  $j = 1,2,\dots,n$ .
2. Memberikan nilai bobot ( $W$ ) yang juga didapat berdasarkan nilai *crisp*.

3. Melakukan normalisasi matriks dengan cara menghitung nilai rating kinerja ternormalisasi ( $r_{ij}$ ) dari alternatif  $A_i$  pada atribut  $C_j$  berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan / benefit =MAKSIMUM atau atribut biaya /cost = MINIMUM). Apabila berupa atribut keuntungan maka nilai  $r_{ij}$  ( $X_{ij}$ ) dari setiap kolom atribut dibagi dengan nilai  $r_{ij}$  Max (MAX  $X_{ij}$ ) dari tiap kolom, sedangkan untuk atribut biaya, nilai  $r_{ij}$  MIN (MIN  $X_{ij}$ ) dari setiap kolom atribut dibagi dengan nilai  $r_{ij}$  ( $X_{ij}$ ) setiap kolom.
4. Melakukan proses perangkingan alternatif ( $V_i$ ) dengan cara mengalikan nilai bobot ( $W_i$ ) dengan nilai rating kinerja ternormalisasi ( $r_{ij}$ ).

#### **II.4. Makanan Pendamping ASI**

Makanan Pendamping ASI ( MP ASI ) adalah makanan atau minuman yang mengandung zat gizi dan diberikan kepada bayi atau anak usia 6-24 bulan guna memenuhi kebutuhan gizi selain dari ASI (Depkes RI, 2006). MP-ASI adalah makanan bergizi yang diberikan mendampingi ASI kepada bayi berusia 6 bulan ke atas atau berdasarkan indikasi medis, sampai anak berusia 24 bulan untuk mencapai kecukupan gizinya (WHO, 2003). Zat gizi pada ASI hanya memenuhi kebutuhan gizi bayi sampai usia 6 bulan, untuk itu ketika bayi berusia 6 bulan perlu diberi makanan pendamping ASI dan ASI tetap diberikan sampai usia 24 bulan atau lebih. (Jumiyati, SKM., M.Gizi, 2014).

#### **II.4.1 Karbohidrat**

Secara umum definisi karbohidrat adalah senyawa organik yang mengandung atom Karbon, Hidrogen dan Oksigen, dan pada umumnya unsur Hidrogen dan oksigen dalam komposisi menghasilkan H<sub>2</sub>O. Di dalam tubuh karbohidrat dapat dibentuk dari beberapa asam amino dan sebagian dari gliserol lemak. Akan tetapi sebagian besar karbohidrat diperoleh dari bahan makanan yang dikonsumsi sehari-hari, terutama sumber bahan makan yang berasal dari tumbuh-tumbuhan.

Sumber karbohidrat nabati dalam bentuk glikogen, hanya dijumpai pada otot dan hati dan karbohidrat dalam bentuk laktosa hanya dijumpai di dalam susu. Pada tumbuh-tumbuhan, karbohidrat dibentuk dari hasil reaksi CO<sub>2</sub> dan H<sub>2</sub>O melalui proses fotosintesis di dalam sel-sel tumbuh-tumbuhan yang mengandung hijau daun (klorofil). Matahari merupakan sumber dari seluruh kehidupan, tanpa matahari tanda-tanda dari kehidupan tidak akan dijumpai. (Dr. Halomoan Hutagalung, 2004).

#### **II.4.2. Protein**

Protein merupakan salah satu senyawa pendukung utama dalam kehidupan biologis suatu organisme, oleh karena itu protein harus tersedia dalam pangan. Kualitas protein pangan tergantung pada kandungan asam amino esensial. Penelitian ini bertujuan untuk mengembangkan cara sederhana estimasi kadar protein dalam berbagai bahan pangan dan pakan, dengan ketepatan dan kecermatan yang tinggi. (Sumarno, dkk, 2002).

### **II.4.3. Kalsium**

Kalsium merupakan mineral dengan jumlah terbesar yang terdapat dalam tubuh. Kebutuhan kalsium pada masa remaja sangat tinggi oleh karena masa pembentukan tulang terbesar terjadi pada masa ini. Karena kebutuhannya yang sangat tinggi, efisiensi penyerapan dan deposit kalsium meningkat hingga 2 kali lebih besar dari masa-masa sebelum ataupun sesudahnya. Sehingga suplai kalsium yang adekuat dari makanan menjadi sangat penting untuk memaksimalkan PBM dan menjaga keseimbangan kalsium tubuh yang optimal. (Sandra Fikawati, 2005).

### **II.4.4. Lemak**

Lemak dan minyak merupakan zat makanan yang penting untuk menjaga kesehatan tubuh manusia. Selain itu lemak dan minyak juga merupakan sumber energi yang lebih efektif dibanding dengan karbohidrat dan protein. Satu gram minyak atau lemak dapat menghasilkan 9 kkal, sedangkan karbohidrat dan protein hanya menghasilkan 4 kkal/gram. (Sandra Hermanto, 2013).

## **II.5. Pengertian *Unified Modeling Language* (UML)**

Saat ini piranti lunak luas dan besar lingkungannya, sehingga tidak bisa dibuat asal-asalan. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur, ketiga unsur itu adalah pemodelan (*notation*), proses (*process*) dan tool digunakan. Memahami notasi pemodelan tanpa mengetahui cara pemakaian yang sebenarnya akan membuat proyek gagal. Dan pemahaman terhadap pemodelan dan proses disempurnakan dengan penggunaan tool yang tepat.

Menurut Yuni Sugiarti (2013, h. 34) *Unified Modeling Language* adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologipemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch [1], metodologi coad [2], metodologi OOSE [3], metodologi OMT [4], metodologi shlaer-mellor [5], metodologi wirfsbrock [6], dsb. Masa itu terkenal dengan masa perang metodologi (*method war*)

dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

### II.5.1 Konsepsi Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams. Main concepts* bisa kita pandang sebagai item yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagram tersebut. (ikc.dinus.ac.id)

Lalu darimana kita mulai untuk menguasai UML, sebenarnya cukup dua hal yang harus kita perhatikan:

1. Menguasai pembuatan diagram UML.
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML.

Tulisan ini pada intinya akan mengupas kedua hal tersebut.

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:


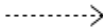


#### 1. Use Case Diagram

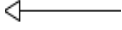





*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor

dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

**Tabel II.1 Diagram Use Case**

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

### **II.5.2. Tujuan UML (Unified Modeling Language)**

Adapun tujuan dari penggunaan UML (*Unified Modeling Language*)

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
2. Menyatakan praktek – praktek terbaik yang dapat terdapat terdalam pemodelan.
3. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan *detail*. Dengan cetak biru ini maka akan bisa diketahui informasi secara *detail* dan *coding* program atau bahkan membaca program dan menginterpretasikan kembali kedalam bentuk diagram (*reverse engineering*).

### **II.6. Pengertian Basis Data ( Database )**

Basis data (database) merupakan suatu data yang terintegrasi, diorganisasikan, dan disimpan dalam suatu cara yang baik tergantung pada aspek desain dan aspek realnya.

Menurut Marlinda (2005, h. 1) menyatakan bahwa “basis data adalah suatu susunan kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang terorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya”.

Menurut Sutabri (2005, h. 161) menyatakan bahwa “database merupakan suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain tidak perlu suatu kerangkapan data (*controlled redundancy*) dengan cara tertentu sehingga mudah digunakan atau ditampilkan kembali”.

### **II.6.1 Model Data**

Model data merupakan sejumlah konsep yang digunakan untuk membuat deskripsi struktur basis data, dengan deskripsi struktur basis data dapat ditemukan jenis data, hubungan dan konstrain data yang harus ditangani. Kebanyakan model data juga membuat spesifikasi untuk operasi dasar dalam pengaksesan dan pembaharuan data pada basis. Model data dapat dikelompokkan berdasarkan konsep pembuatan deskripsi struktur basis data, yaitu model data konseptual dan model data fisikal.

#### **1. Model data konseptual.**

Menyajikan tentang bagaimana pemakai basis data memandang atau memperlakukan data. Dalam model data konseptual digunakan konsep entity, atribut dan hubungan.

#### **2. Model data fisikal.**

Merupakan konsep bagaimana deskripsi detail data disimpan dalam komputer dengan menyajikan informasi tentang format rekaman, urutan rekaman dan jalur pengaksesan data. Informasi jalur pengaksesan dan merupakan struktur yang dapat membuat pencarian rekaman data lebih efisien.

## II.6.2 Hierarki Data Dalam Database

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis (dasar) dalam menyediakan informasi bagi para pemakai.

### 1. *Bit*

*Bit* merupakan bagian data terkecil, dapat berupa karakter numerik, huruf maupun karakter-karakter khusus yang membentuk suatu item data, dimana kumpulan karakter membentuk satu *field*.

### 2. *Byte*

*Byte* merupakan sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. Sistem biner merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin.

### 3. *Field*

Suatu *field* menggambarkan suatu atribut *record* yang menunjukkan suatu item dari data seperti nama, alamat, dimana kumpulan *field* membentuk suatu *record*.

### 4. *Record*

Suatu *record* menggambarkan satu kesatuan data yang sejenis, dimana kumpulan dari *file-file* membentuk *database*.

### 5. *File*

Suatu *file* menggambarkan kumpulan dari beberapa *record* yang dapat menampung data-data.

## 6. Database

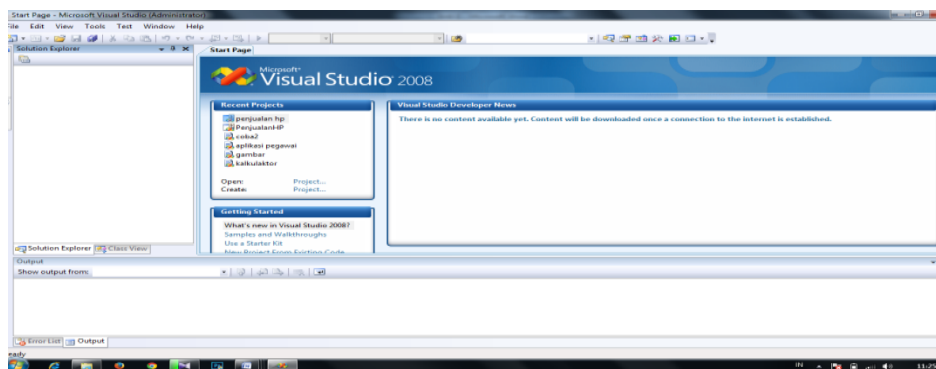
Suatu *database* menggambar data yang saling berhubungan antara satu dengan yang lainnya.

## II.7 Pengertian *Visual Basic*

*Visual Basic.Net* hanya merupakan salah satu aplikasi bahasa pemrograman yang dapat digunakan untuk membuat aplikasi komputer. Dalam hal ini kita sebaiknya memahami bahwa bahasa pemrograman hanya merupakan salah satu aspek dalam suatu aplikasi *Windows*.

Menurut Adi Nugroho (2010, h.50-63) *Visual basic.Net* adalah merupakan bahasa pemrograman aras tinggi (*high level programming language*). Salah satu bahasa pemrograman yang dapat kita gunakan untuk membuat aplikasi komputer.

Pada dasarnya tampilan baru ini memudahkan menggunakan microsoft visual basic seperti di bawah ini :



Gambar.II.2 *Visual Studio 2008*

Melalui lembar ini dapat membuat aplikasi visual basic yang diperlukan. Sebelum membuat aplikasi baru, ada dua buah istilah yang diperlu diketahui dalam visual basic 2008 :

1. Project untuk membukanya, pembaca cukup mengklik namanya. Dalam jendela kita juga membuat proyek yang baru denga mengklik link create yang ada dalam bagian bawah jendela project ini..
2. Suliton Explorer adalah jendela Suliton Explorer berkaitan dengan pengelolaan proyek perograman *VB.NET* yang sedang kita lakukan.

### **II.7.1 Ruang Lingkup *Visual Basic***

Setelah visual basic diaktifkan, maka akan muncul sebuah layar yang ada di bawah ini. Layar ini adalah layar utama dari Visual Basic yang nantinya akan berfungsi untuk merancang program aplikasi.

#### *1. Control Menu*

*Control menu* adalah menu yang digunakan untuk melakukan manipulasi jendela atau layar *Visual Basic*. Pada *control menu* ini dapat dilakukan perubahan ukuran, pemindahan jendela yang aktif. Untuk mengaktifkan *control menu* klik tombol mouse pada sudut sebelah kiri atas jendela. Kemudian akan muncul kotak dialog *Control Menu* dengan berbagai sub menu antara lain:

- |                    |  |
|--------------------|--|
| 1) <i>Restore</i>  | : Mengubah ukuran jendela pada yang sebelumnya |
| 2) <i>Size</i>     | : Mengubah ukuran jendela                      |
| 3) <i>Minimize</i> | : Meminimalkan ukuran jendela                  |
| 4) <i>Close</i>    | : Untuk menutup jendela                        |

#### *2. Menu*

*Menu* yang terdapat pada *visual basic* berisi perintah-perintah yang dapat melakukan perintah tertentu. Pada umumnya aplikasi *windows* memiliki model yang sama dengan format yang sama.

### 3. *Toolbar*

*Toolbarr* adalah salah satu bagian dari setiap jendela aplikasi yang dijalankan dengan sisten operasi *windows*. Dapat dikatakan selalu memiliki baris *tool* atau *toolbar* yang berisi tombol-tombol (*icon*) perintah. Pada menu *visual basic* biasanya selalu menampilkan *toolbar* standar dengan gambar berikut ini:

### 4. *Form Windows*

*Form* atau jendela *form* merupakan daerah kerja utama dari pembuatan program atau tempat perancangan aplikasi (*Container*). Pada daerah *form* inilah meletakkan dan menggambarkan objek interaktif seperti misalnya tombol-tombol, gambar, teks, garis, tabel, *combo*, *checkbox* dan *tools* lainnya. sehingga objek yang berada pada *form* tersebut akan ditampilkan pada layar *windows* jika program dijalankan. *Form* tersebut akan menjadi latar belakang atau tempat (*container*) dari objek dari sebuah program yang dijalankan.

### 5. *Toolbox*

*Toolbox* adalah fasilitas yang berisi objek atau kontrol untuk merancang pada jendela *form*. Secara *default* pada saat mengaktifkan *visual basic* jendela *toolbox* akan ditampilkan pada sebelah kiri layar *visual basic*.

### 6. *Project Explorer*

Jendela ini merupakan kumpulan dari sejumlah aplikasi yang sering disebut dengan *project*. Sementara *project* itu sendiri memiliki banyak *file* seperti *file form, modul, class* dan lainnya.

#### 7. *Property Windows*

Jendela ini adalah yang memiliki semua informasi tentang suatu objek yang terdapat pada *visual basic*. Properti merupakan suatu sifat dari objek dapat memiliki properti warna, ukuran, posisi, lebar, Jenis tipe dan sifat yang lainnya dan setiap objek tersebut memiliki properti yang saling berbeda-beda.

#### 8. *Code Window*

Salah satu jendela yang terpenting dalam Visual Basic adalah jendela *Code*. Dimana jendela ini berfungsi untuk penulisan dari instruksi-instruksi program untuk pembuatan program. Fungsi dari setiap objek dapat ditambahkan melalui fasilitas jendela *code* ini.

## **II.8 Pengenalan SQL Server Management Studio**

Dalam pemrograman basis data, salah satu bahasa yang harus kita kuasai adalah SQL. SQL bahasa komputer standar yang digunakan untuk berkomunikasi dengan sistem manajemen basis data (RDMS).

Menurut Ema Utami (2012, h. 63) *Structured Query Language* (SQL) adalah bahasa yang banyak digunakan dalam berbagai produk *database*. SQL dibangun di laboratorium *IBM-San Jose California* sekitar akhir tahun 70-an. Pertama kali dikembangkan sebagai bahasa di produk *database* DB2 yang sampai saat ini merupakan produk *database* andalan IBM. SQL sering di lafalkan dengan

“*sequel*”. Saat ini organisasi standar Amerika (ANSI) menetapkan standar bahasa SQL yaitu ANSI-92 standard. Masing-masing vendor *database* memiliki dialeknya sendiri sebagian besar spesifikasinya mengacu pada standar ANSI tersebut dengan berbagai ekstensi tambahan. SQL Server menggunakan bahasa *Transact-SQL* dalam produknya, sedangkan *Oracle* menggunakan PL/SQL.