

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Menurut Jurnal (Hilyah Magdalena ; 2012:191) Sistem Pendukung Keputusan *Decision Support System* (DSS) pada tahun 1970-an sebagai pengganti istilah *Management Information System*. Tetapi pada dasarnya Sistem Pendukung Keputusan merupakan pengembangan lebih lanjut dari *Management Information System* yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Maksud dan tujuan dari adanya sistem pendukung keputusan, yaitu untuk mendukung pengambil keputusan memilih alternatif keputusan yang merupakan hasil pengolahan informasi-informasi yang diperoleh dengan menggunakan model-model pengambilan keputusan serta untuk menyelesaikan masalah yang bersifat terstruktur, semi struktur dan tidak terstruktur.

Pada dasarnya pengambilan keputusan adalah suatu pendekatan sistematis pada suatu masalah, pengumpulan fakta dan informasi, penentuan yang baik untuk alternatif yang dihadapi, dan pengambilan tindakan yang menurut analisis. Untuk kepentingan itu, sebagian besar pembuat keputusan dengan mempertimbangkan rasio manfaat/biaya, dihadapkan pada suatu keharusan untuk mengandalkan sistem yang mampu memecahkan suatu masalah secara efektif, yang kemudian disebut dengan sistem pendukung keputusan (SPK).

### **II.1.1. Pengertian Sistem**

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Suatu sistem didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berintegrasi untuk mencapai suatu tujuan. Suatu sistem mempunyai karakteristik yaitu mempunyai komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah dan sasaran (Asbon Hendra;2011:157).

### **II.1.2 Pengertian Keputusan**

Keputusan merupakan kegiatan memilih suatu strategi atau tindakan dalam pemecahan masalah tersebut. Tindakan memilih strategi aksi yang diyakini akan memberikan solusi terbaik atas sesuatu itu disebut pengambilan keputusan. Keputusan merupakan sebuah kesimpulan yang dicapai sesudah dilakukan pertimbangan (analisa), yang terjadi setelah satu kemungkinan dipilih, sementara yang lain dikesampingkan.

### **II.1.3 Konsep Sistem Pendukung Keputusan**

Menurut Jurnal (Kusrini;2007) sistem pendukung keputusan menurut *Gorry Dan Scout Morton* adalah sistem berbasis komputer interaktif, yang membantu para pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur.

Hal yang perlu diperhatikan di sini adalah bahwa keberadaan SPK bukan untuk menggantikan tugas manajer, tetapi untuk menjadi sarana penunjang bagi mereka. SPK merupakan implementasi teori teori pengambilan keputusan yang

telah diperkenalkan oleh ilmu ilmu seperti *operation research dan management science*. Hanya bedanya adalah bahwa dahulu untuk mencari penyelesaian masalah yang dihadapi harus dilakukan perhitungan iterasi secara manual. Dalam kedua bidang ilmu di atas, dikenal istilah *decision modeling, decision theory, decision analysis* yang pada hakekatnya adalah merepresentasikan permasalahan manajemen yang dihadapi setiap hari ke dalam bentuk kuantitatif .

## **II.2. Tanaman Karet**

Karet alam (*Hevea Brasiliensis*) merupakan suatu komoditas yang saat ini paling banyak dikembangkan di dunia terutama oleh negara-negara produsen karet alam. Negara tersebut antara lain Malaysia, Thailand dan Indonesia. Tujuan dari budidaya tanaman karet alam adalah memproduksi lateks dan bekuannya. Lateks dan bekuannya merupakan bahan utama bagi industri yang memproduksi produk berbahan karet seperti ban, sepatu karet, balon, dan produk-produk karet lainnya.

Karet juga merupakan salah satu bahan hasil pertanian yang banyak terdapat di Indonesia dan menjadi penyumbang devisa negara yang cukup besar dengan produksi sebanyak 1,6 juta ton per tahun. Disamping itu, komoditas karet alam merupakan sumber mata pencaharian secara langsung bagi 1,6 juta keluarga petani dan secara keseluruhan diperkirakan menjadi sumber penghidupan baik secara langsung maupun tidak langsung bagi sekitar 15 juta penduduk Indonesia. Kebutuhan karet alam maupun karet sintetik terus meningkat sejalan dengan meningkatnya standar hidup manusia. Kebutuhan karet sintetik relative lebih mudah dipenuhi karena sumber bahan baku relatif tersedia walaupun harganya

mahal, akan tetapi karet alam dikonsumsi sebagai bahan baku industri tetapi diproduksi sebagai komoditi perkebunan. (Wiyanto dan Nunung, 2013).

### **II.2.1. Penentuan Pemilihan Tempat Penanaman Karet**

Agar menghasilkan suatu lateks dan bekuan yang baik maka pemilihan tempat penanaman karet harus baik pula. Pemilihan tempat penanaman karet yang baik tergantung dengan beberapa hal seperti dataran lahan, iklim, tekstur tanah.

Untuk mendapatkan hasil budidaya tanaman karet secara optimal dibutuhkan syarat-syarat pertumbuhan sebagai berikut (PTPN III Medan, 2017) :

1. Lahan
  - a. Areal tidak terendam, yang dimaksud tidak terendam adalah saat banjir areal dapat kering dalam waktu 3-4 hari.
  - b. Kemiringan < 35%.
  - c. Solum (kedalaman tanah) cukup dalam, tidak dominan pasir dan batuan.
  - d. Tinggi dari permukaan laut 200 m sampai dengan 600 m.
  - e. pH tanah 3,0 – 8,0 dan yang optimal 4,0 – 6,5.
2. Iklim
  - a. Curah hujan 1500 – 4000 mm/tahun dan optimal 2500 mm/tahun yang terbagi merata sepanjang tahun.
  - b. Suhu 25 – 35 derajat celcius dan yang optimal rata-rata 28 derajat celcius.
  - c. Tidak terdapat angin kencang yang berpotensi merusak tanaman karet.

### 3. Tekstur Tanah

Tanaman karet dapat tumbuh pada berbagai jenis tanah seperti *Podsolik*, *Latosol*, *Hidromorfik Kelabu (HK)*, *Regosol*, *Andosol*, *Organosol* dan *Alluvial*.

### II.3. Dempster Shafer

Teori *Dempster-Shafer (DST)* merupakan teori matematika dari *evidence*. Teori ini dapat memberikan sebuah cara untuk menggabungkan *evidence* dari beberapa sumber dan mendatangkan atau memberikan tingkat kepercayaan (direpresentasikan melalui fungsi kepercayaan) dimana mengambil dari seluruh *evidence* yang tersedia. Teori tersebut pertama kali dikembangkan oleh Arthur P. Dempster dan Glenn Shafer.

Ada berbagai macam penalaran dengan model yang lengkap dan sangat konsisten, tetapi pada kenyataannya banyak permasalahan yang tidak dapat terselesaikan secara lengkap dan konsisten. Ketidakkonsistenan yang tersebut adalah akibat adanya penambahan fakta baru. Penalaran yang seperti itu disebut dengan penalaran *non monotonis*. Untuk mengatasi ketidakkonsistenan tersebut maka dapat menggunakan penalaran dengan teori *Dempster-Shafer*.

Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval :  $[Belief, Plausibility]$ . *Belief (Bel)* adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. *Plausibility (Pl)* dinotasikan sebagai :

$$Pl(s) = 1 - Bel(s) \quad \text{tahap (1)}$$

*Plausibility* juga bernilai 0 sampai 1. Jika yakin akan  $s$ , maka dapat dikatakan bahwa  $Bel(s)=1$ , dan  $Pl(s)=0$  tahap (2)

Pada teori *Dempster-Shafer* dikenal adanya *frame of discrement* yang dinotasikan dengan  $\Omega$ . *Frame* ini merupakan semesta pembicaraan dari sekumpulan *hipotesis*. Tujuannya adalah mengaitkan ukuran kepercayaan elemenelemen  $\Omega$ . Tidak semua *evidence* secara langsung mendukung tiap-tiap elemen.

Untuk itu perlu adanya probabilitas fungsi densitas ( $m$ ). Nilai  $m$  tidak hanya mendefinisikan elemen-elemen  $\Omega$  saja, namun juga semua subsetnya. Sehingga jika  $\Omega$  berisi  $n$  elemen, maka subset  $\Omega$  adalah  $2^n$ . Jumlah semua  $m$  dalam subset  $\Omega$  sama dengan 1. Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai :  $m\{\emptyset\} = 1,0$  (Deby dkk, 2015).

Apabila diketahui  $X$  adalah subset dari  $\Omega$ , dengan  $m_1$  sebagai fungsi densitasnya, dan  $Y$  juga merupakan subset dari  $\Omega$  dengan  $m_2$  sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi  $m_1$  dan  $m_2$  sebagai  $m_3$ , yaitu :

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)} \dots \dots \dots (3)$$

Dimana :

$m_1$  = Densitas untuk kriteria pertama

$m_2$  = Densitas untuk kriteria kedua

$m_3$  = Kombinasi dari kedua Densitas diatas

$\emptyset$  = Himpunan Kosong

$X \cap Y$  = Himpunan Evidence

## **II.4. Pemodelan Sistem**

Pemodelan secara umum, dimengerti sebagai proses merepresentasikan objek nyata atau realita sebagai seperangkat persamaan matematika, grafis ataupun bagan agar mudah dipahami oleh pihak yang berkepentingan. Lebih khusus lagi, istilah ini sering digunakan untuk proses menggambarkan konsep yang mewakili obyek-obyek dalam pengembangan sistem informasi. Pemodelan dalam pengembangan sistem informasi, berevolusi selaras perkembangan teknologi dan metodologi pengembangannya. Kesamaan pandang diperlukan dalam pemodelan, yang akhirnya menuntut adanya keseragaman atau standar (Aliq Zuhdi, 2017).

Teknologi obyek, membutuhkan pemodelan yang relevan, telah mendorong Jacobson dkk., menciptakan standar pemodelan untuk pengembangan sistem, dengan pendekatan obyek yang dikenal dengan UML (*Unified Modeling Language*). UML memiliki keuntungan yang menghasilkan representasi yang dapat diverifikasi melalui penalaran logis, pengujian, atau bahkan simulasi. Sebagai contoh, diagram proses bisnis dapat diuji dengan mental melewati kasus imajinasi, melalui pemikiran logis yang mungkin berbeda untuk melihat apakah itu merupakan upaya menghasilkan *output* yang diperlukan.

### **II.4.1. *Unified Modelling Language* (UML)**

Menurut Munawar (2005), UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa

pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Semakin kompleks bentukan sistem yang akan dibuat, maka semakin sulit komunikasi antara orang-orang yang saling terkait dalam pembuatan dan pengembangan software yang akan dibuat. Berikut ini adalah beberapa diagram UML :

1. *Class Diagram*
2. *Use Case Diagram*
3. *Activity Diagram*
4. *Sequence Diagram*

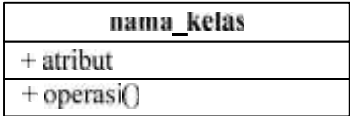
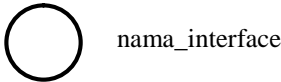


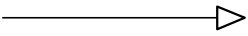

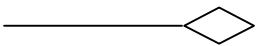
#### **II.4.2. *Class Diagram***

Diagram kelas atau *Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas dapat dilihat pada gambar tabel II.1

**Tabel II.1. Simbol-Simbol *Class Diagram***

Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur sistem</p>
<p>Antarmuka / <i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek</p>
<p>Asosiasi / <i>association</i></p> 	<p>Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i></p>
<p>Generalisasi / <i>Generalization</i></p> 	<p>Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Relasi antarkelas dengan makna kebergantungan antarkelas</p>
<p>Agregasi / <i>aggregation</i></p> 	<p>Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>)</p>

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 : Rosa A.S., M. Shalahuddin, 2014 : 141-147)

### II.4.3. Use Case Diagram

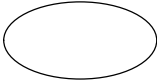
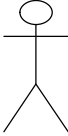

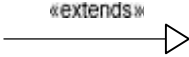

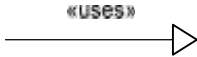
*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar tabel II.2 dibawah ini :

**Tabel II.2. Simbol-simbol Diagram *Use Case***

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor / <i>Actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama actor</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komuniikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>Extends</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / <i>Generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / <i>Include</i> / <i>Uses</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p>


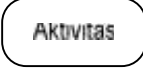



(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 156-158)

#### II.4.4. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas dapat dilihat pada gambar tabel II.3

**Tabel II.3. Simbol-simbol Diagram Aktivitas**

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	<i>Decision</i> , atau pilihan untuk mengambil keputusan.
Penggabungan / <i>Join</i> 	Arah tanda panah alur proses.
Status Akhir 	Titik akhir atau akhir dari aktivitas.

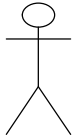
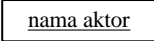


(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 162)


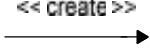
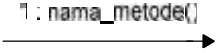
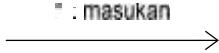
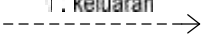
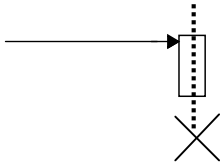
### II.4.5. Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sequence juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Berikut adalah simbol-simbol yang ada pada diagram sequence dapat dilihat pada gambar tabel II.4

**Tabel II.4. Simbol-simbol Diagram Sequence**

Simbol	Keterangan
Aktor / <i>Actor</i>  Atau  Tanpa waktu aktif	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan

<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang di akhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 165-167)

## II.5. Database

Banyak sekali definisi tentang database yang diberikan oleh para pakar dibidang ini. Database terdiri dari dua penggalan kata yaitu data dan *base*, yang artinya berbasiskan pada data. Tetapi secara konseptual, database diartikan sebuah koleksi atau kumpulan data yang saling berhubungan (*relation*), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi. Sebuah informasi yang berdiri sendiri tidaklah dikatakan *database* (Sonya, 2016).

Sebagai contoh, nomor telepon seorang pelanggan, disimpan dalam banyak tempat apakah itu difile pelanggan, difile alamat dan di lokasi yang lain. Antara file yang satu dengan file yang lainnya tidak saling berhubungan, sehingga apabila salah seorang pelanggan berganti nomor telepon dan anda hanya mengganti di file pelanggan saja, akibatnya akan terjadi ketidakcocokan data, karena di lokasi yang lain masih tersimpan data telepon yang lama.

Dalam sistem database hal ini tidak boleh dan tidak bisa terjadi, karena antara file yang satu dengan file yang lain saling berhubungan. Jika suatu data yang sama anda ubah, data tersebut di file yang lain akan otomatis berubah juga. Sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan, terhadap data secara mudah dan cepat (Efektif, Efisien dan Akurat).

Data adalah fakta, baik berupa sebuah objek, orang dan lain – lain yang dapat dinyatakan dengan suatu nilai tertentu (angka, simbol, dan karakter tertentu). Sedangkan informasi adalah data yang telah diolah sehingga bernilai guna dan dapat dijadikan bahan dalam pengambilan keputusan, (Yuhefizard, 2003).

Hubungan data dan informasi dapat digambarkan sebagai berikut :



**Gambar II.1. Data dan Informasi.**  
(Sumber : Yuhefizard , 2010)

Terdapat banyak *database* yang umum digunakan diantaranya MySQL dan SQL Server. Pada penelitian ini penulis menggunakan *database* SQL Sever.

### II.5.1. SQL Server

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang dibangun oleh *Microsoft*, yang digunakan untuk menyimpan dan mengolah data. Pada SQL Server 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL Server 2008, kita bisa membuat *object – object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, fuction, stored procedure, trigger* dan *view*. Selain *object*, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data.

Adapun beberapa dari kelebihan dan kekurangan dari SQL Server sebagai berikut :

#### 1. Kelebihan Microsoft SQL Server

- ) Dengan kemampuannya untuk mengolah data yang besar maka DBMS ini sangat cocok untuk perusahaan mikro, menengah hingga perusahaan besar sekalipun.
- ) DBMS jenis memiliki kelebihan memmanage user data serta masing-masing user dapat diatur hak aksesnya terhadap pengaksesan data base oleh DBA.
- ) Mempunyai tingkat keamanan data yang sangat baik.
- ) Dapat melakukan *back up, recovery* dan *rollback* data dengan mudah.
- ) Mempunyai kelebihan untuk membuat *database mirroring* dan *clustering*.

## 2. Kekurangan Microsoft SQL Server

- ) DBMS jenis ini hanya dapat berjalan pada sistem operasi / *platform windows* saja.
- ) *Software* ini mempunyai lisensi dari *Microsoft* sehingga pemakaiannya membutuhkan biaya yang cukup mahal.

## II.6. Visual Basic 2010

*Visual Basic 2010* merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. *Visual Studio* merupakan produk pemrograman andalan dari *Microsoft Corporation*, dimana didalamnya berisi beberapa jenis IDE pemrograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#*, dan *Visual F#*.

Semua IDE pemrograman tersebut sudah mendukung implementasi *.Net Framework* terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. Adapun *database* standart yang disertakan adalah *Microsoft Sql Server 2008 Express*.

*Visual Basic 2010* merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangannya yang terdapat didalamnya antara lain dukungan terhadap *Library* terbaru dari *Microsoft* (Wahana Komputer : 2010).