

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem Informasi**

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Mara Destiningrum, 2017).

Adapun pengertian tentang informasi, yaitu data yang telah diproses ke dalam suatu bentuk yang mempunyai arti bagi penerima dan memiliki nilai nyata yang dibutuhkan untuk proses pengambilan keputusan saat ini maupun saat mendatang (Paulina Thomas, 2017).

Sistem informasi merupakan serangkaian komponen berupa manusia, prosedur, data, dan teknologi (seperti komputer) yang digunakan untuk menghasilkan informasi yang bernilai untuk pengambilan keputusan. Sistem informasi yang melekat dan merupakan infrastruktur penunjang keberhasilan bagi setiap organisasi dalam mencapai tujuannya (Darmanta Sukrianto, 2017).

Kualitas Informasi tergantung dari 3 hal, yaitu informasi harus : Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Tetap pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

## **II.2. Sistem Informasi Eksekutif**

### **II.2.1. Pengertian Sistem Informasi Eksekutif**

Sistem Informasi Eksekutif (SIE) merupakan sistem informasi yang berbasis komputer dengan para eksekutif sebagai *end-user* penggunaannya. SIE difokuskan untuk membantu para eksekutif mengetahui informasi secara cepat guna mengidentifikasi masalah dan mengambil keputusan berdasarkan informasi yang telah dihasilkan (Fatoni, 2019)

Sistem informasi eksekutif adalah satu jenis dari manajemen informasi sistem dimaksud untuk memudahkan dan mendukung keterangan dan pembuatan keputusan kebutuhan dari eksekutif senior dengan menyediakan kemudahan akses terhadap keduanya internal dan eksternal keterangan relevan untuk bertemu golongan strategis organisasi. Ini biasanya dipertimbangkan sebagai satu bentuk dikhususkan dari satu sistem mendukung keputusan. Penekanan dari sistem informasi eksekutif berada diatas peraga grafis dan mudah untuk pergunaan *interface* pemakaian. Sistem informasi ini sangat

penting karena sistem informasi berbasis komputer ini dapat memberikan informasi kepada eksekutif untuk mendukung pengambilan keputusan dalam hal menentukan strategi dan kebijakan organisasi. Di samping itu sesuai dengan dinamika bisnis dan persaingan antar perusahaan yang ketat menyebabkan eksekutif membutuhkan informasi yang cepat dan akurat.

Jadi bisa di simpulkan bahwa informasi eksekutif adalah suatu data yang diproses dan akan memberikan nilai arti serta bermanfaat, data yang diproses digunakan sebagai bahan keputusan seorang eksekutif untuk alat keterangan yang pasti dalam membuat kesimpulan. Sistem eksekutif berfungsi menyediakan kumpulan informasi untuk pengguna eksekutif disuatu organisasi guna membantu mengidentifikasi, menentukan pilihan putusan permasalahan serta memahaminya.

### **II.2.2. Keuntungan dan Kekurangan Sistem Informasi Eksekutif**

Kelebihan menggunakan Sistem Informasi Eksekutif dalam suatu organisasi atau perusahaan adalah sebagai berikut:

- a. Mempermudah para eksekutif untuk menggunakan pengalamannya dalam dunia komputer.
- b. Menyediakan pengiriman tepat waktu dari keterangan rangkuman perusahaan.
- c. Keterangan yang disediakan semakin mudah dimengerti.
- d. Biasanya menawarkan efisiensi unujuk membuat keputusan.
- e. Melakukan penyaringan data untuk manajemen.
- f. Meningkatkan pemeriksaan keterangan.

- g. Dapat mengakses dan memudahkan jangkauan data internal dan eksternal yang bersifat luas.

Kekurangan dari Sistem Informasi Eksekutif adalah sebagai berikut:

- a. Memiliki fungsi yang terbatas, tidak dapat melakukan perhitungan kompleks.
- b. Pada perusahaan kecil mungkin membutuhkan biaya lebih untuk membuat implementasi.
- c. Karena sistemnya besar, sehingga sulit untuk mengaturnya.
- d. Pembuatannya harus dapat memenuhi kebutuhan informasi bagi eksekutif.
- e. Eksekutif mungkin menghadapi beban terlalu berat untuk membuat keterangan.

### **II.3. Metode *Drill Down***

*Drill Down* suatu cara untuk mendapatkan informasi di mana seorang pemecah masalah akan dapat menampilkan tampilan ringkasan, kemudian secara berturut-turut menampilkan rincian-rincian di tingkat yang lebih rendah. Metode *drill down* dapat memberikan informasi dan dapat diakses secara hierarkis, mulai dari yang bersifat umum atau global kemudian dipecah hingga ke level yang lebih detail. Informasi yang ditampilkan berupa grafik maupun tabel (M. Hadi Prayitno, 2017).

Metode *Drill Down* banyak digunakan dalam penelitian baik untuk menentukan atau memberi informasi dan data penjualan kepada pihak eksekutif, yang pertama penelitian yang dilakukan oleh Inmon, menyimpulkan bahwa salah satu fasilitas yang sangat berguna dalam sistem informasi eksekutif. Dengan

fasilitas ini, eksekutif dapat mengakses informasi secara hierarkis, mulai dari yang bersifat umum atau global kemudian dipecah hingga ke level yang lebih detail. Informasi yang diberikan berupa grafik maupun tabel. Dengan memperoleh detail dari suatu informasi, eksekutif dapat melakukan analisis secara lebih akurat dan tepat sasaran

Dari kesimpulan diatas penulis dapat menyimpulkan bahwa metode *Drill Down* dapat digunakan untuk memberikan informasi kepada eksekutif dalam melakukan pengambilan keputusan.

#### **II.4. Pembiayaan Customer**

Pembiayaan berdasarkan Undang-Undang Nomor 21 pasal 1 ayat 4 tahun 2008 : Pembiayaan dapat berupa transaksi bagi hasil, transaksi sewa-menyewa, transaksi jual beli, transaksi pinjam meminjam, dan transaksi sewa-menyewa jasa atau multi jasa (Hamdani, 2016).

Dalam pemberian pembiayaan kepada customer terdapat prosedur yang harus dijalankan. Prosedur merupakan suatu urutan yang tersusun yang biasanya melibatkan beberapa orang dalam suatu bagian departmen atau lebih, serta disusun untuk menjamin penanganan secara seragam terhadap transaksi-transaksi perusahaan yang terjadi berulang-ulang. Prosedur adalah suatu kegiatan *clearical*, biasanya melibatkan beberapa orang dalam suatu department atau lebih, yang dibuat untuk menjamin penanganan secara seragam transaksi perusahaan yang terjadi berulang-ulang. Prosedur pembiayaan merupakan suatu metode yang harus ditempuh untuk melaksanakan kegiatan pembiayaan. Setiap

pejabat perusahaan yang berhubungan dengan pembiayaan harus menempuh prosedur pembiayaan yang sehat, meliputi prosedur persetujuan pembiayaan, prosedur administrasi, serta prosedur pengawasan pembiayaan.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa sebelum suatu fasilitas pembiayaan diberikan maka perusahaan harus merasa yakin bahwa pembiayaan yang diberikan benar-benar akan kembali. Keyakinan tersebut diperoleh dari hasil penilaian sebelum pembiayaan tersebut disalurkan. Penilaian pembiayaan oleh perusahaan dapat dilakukan dengan berbagai cara untuk mendapatkan keyakinan tentang nasabahnya, seperti melalui prosedur penilaian yang benar. Jadi sebelum memutuskan pemberian pembiayaan atau melakukan pencairan dan melalui kredit, maka kreditur harus melakukan penilaian terhadap calon debitur.

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

Menurut Shalahuddin berpendapat bahwa UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requerement*, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek (Rosa dan Shalahuddin ; 2017).


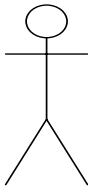

### **II.5.1. Use Case Diagram**






*Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara

satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada *Use Case Diagram*:

**Tabel II.1. Use Case Diagram**

NO	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem unit-unit yang saling bertukar peran antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi. Jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang, biasanya dinyatakan kata benda di awal frase nama aktor.
3.		<i>Association</i>	Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .

4.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <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.
5.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.		<i>Uses / Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

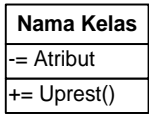
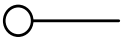



(Sumber : Ade Handini, 2016)

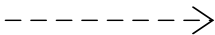
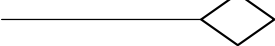
## II.5.2. Class Diagram (Diagram Kelas)

*Class diagram* merepresentasikan *class*, *package*, objek, dan atribut apa saja yang terdapat dalam sistem. *Class diagram* juga menampilkan relasi yang terjadi antar *class* (Mohammad Arifin, 2017).

Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

**Tabel II.2. Simbol-Simbol Class Diagram**

NO	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem.
2.		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.		<i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		<i>directed association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.		<i>Generalization</i>	Relasi antarkelas dengan makna generalisasi spesialisasi (umum-khusus).

6.		<i>dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
7.		<i>aggregation</i>	Relasi antarkelas dengan makna semua bagian ( <i>whole-part</i> ).

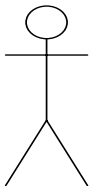
(Sumber : Ade Handini, 2016)

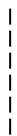
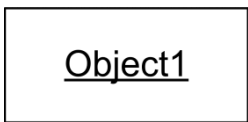

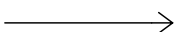


### II.5.3. Sequence Diagram

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Berikut simbol-simbol yang sering digunakan dalam *sequence diagram*:

**Tabel II.3. Sequence Diagram**

NO	Simbol	Nama	Keterangan
1.		Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan kata benda di awal frase nama aktor.

2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3.		<i>Object</i>	Menyatakan objek yang berinteraksi dengan pesan.
4.		<i>Activation</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.		<i>Message</i>	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
6.		<i>Message Call</i>	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.		<i>Recursive</i>	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.



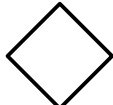



(Sumber : Ade Handini, 2016)

#### II.5.4. Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Handini, 2016).

Berikut simbol-simbol yang sering digunakan dalam *activity diagram*:

**Tabel II.4. Activity Diagram**

NO	Simbol	Nama	Keterangan
1.		<i>Initial State</i>	<i>Initial State</i> aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		<i>Decision</i>	<i>Decision</i> ialah percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		<i>Final State</i>	<i>Final State</i> yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah akhir.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Ade Handini, 2016)

### **II.5.5. Tujuan UML (*Unified Modeling Language*)**

Adapun tujuan dari penggunaan *UML (Unified Modeling Language)* sebagai berikut:

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemograman dan prosese rekayasa.
2. Menyatukan praktek-praktek terbaik yang dapat 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 menginteraksikan kembali ke dalam bentuk diagram (*reverse engineering*).

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

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa file teks apapun ataupun database management sistem (DBMS).

Kebutuhan basis data dalam sistem informasi meliputi:

- a. Memasukan, menyimpan dan mengambil data
- b. Membuat laporan berdasarkan data yang telah disimpan

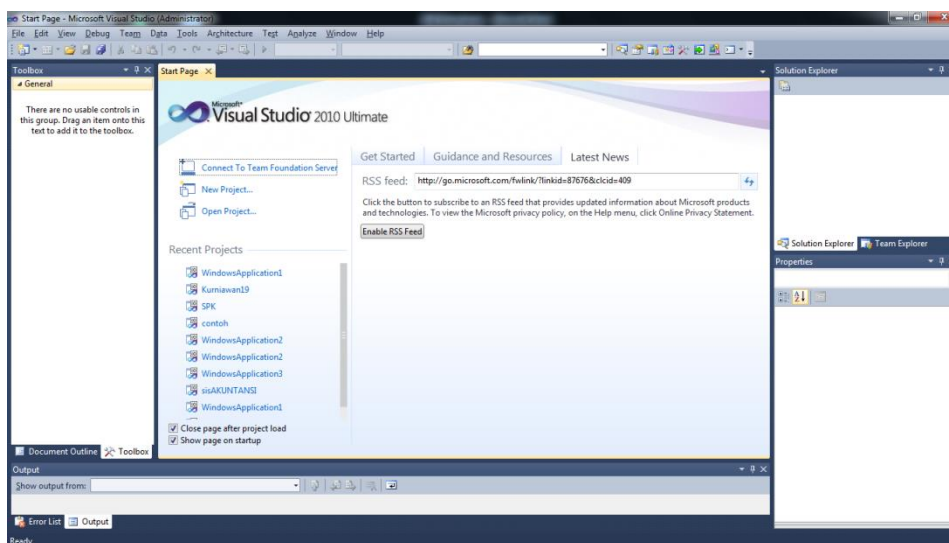
Tujuan dari dibuatnya tabel-tabel disini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (record) dimana setiap baris terdiri dari beberapa kolom.

## **II.7. *Microsoft Visual Studio 2010***

*Microsoft visual Studio 2010* adalah salah satu komponen *microsoft visual studio 2010*. Software ini diluncurkan *microsoft* pada tanggal 12 april 2010 dengan nama kode Dev10 dan menggunakan Net Framework 4.0. *Integrated Development Environment (IDE)* pada visual studio 2010 telah didesain ulang sehingga lebih enak dipandang dan digunakan programmer.

*Microsoft Visual Studio* adalah sebuah *Integrated Development Environment* buatan *Microsoft Coroporation*. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*) (Nency Extise Putri, 2019)

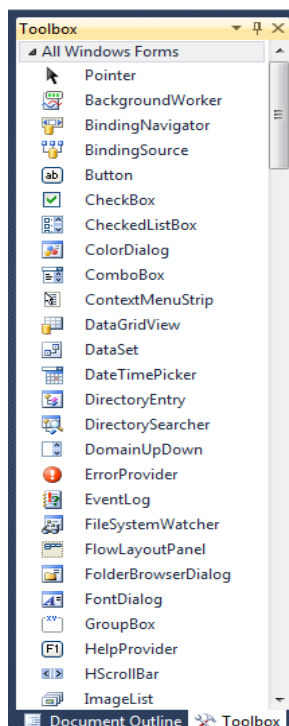
Untuk kode editor nya, visual basic 2010 telah menambah fitur *highlights reference*. Ketika satu simbol/kode dalam bahasa pemrogramannya dipilih, maka simbol/kode yang sama, meskipun penggunaannya berbeda akan terlihat berwarna yang sama. Misal jika kode *math* dipilih, seluruh kode *math* akan terlihat berwarna sama.



**Gambar II.3. Visual Studio 2010**

### II.7.1. Visual Studio Toolbox

*Toolbox* dapat diartikan sebagai kotak alat. Didalam *toolbox* terdapat banyak ikon di mana tiap ikon tersebut melambangkan objek yang disebut kontrol. Kontrol pada *toolbox* bisa digunakan untuk membuat *instance* atau kejadian baru di *form* yang digunakan selama pemrograman.



**Gambar II.4. Toolbox All WindowsForm**

Penjelasan fungsi-fungsi dari kontrol *toolbox* yang ada di *visual studio*

2010 seperti berikut:

**Tabel II.5. Kontrol Toolbox Visual Studio**

NO	Nama	Keterangan
1.	<i>Pointer</i>	Untuk memilih <i>pointermouse</i> atau melepaskan dari memilih objek-objek <i>toolbox</i> .
2.	<i>Background Worker</i>	Mengeksekusi operator tertentu di <i>thread</i> lain.
3.	<i>Binding Navigator</i>	Menyediakan antarmuka untuk navigasi dan manipulasi data yang dikaitkan dengan <i>form</i> .
4.	<i>Binding Source</i>	Mengenkapsulasin sumber data untuk sebuah <i>form</i> dan menyediakan fasilitas <i>update</i> , <i>navigasi</i> , <i>filter</i> , dan <i>sorting</i> .

5.	<i>Button</i>	Untuk menerima klik dari pengguna. Jika pengguna mengkliknya maka kode tertentu bisa dieksekusi.
6.	<i>Check Box</i>	Untuk menerima tanda centang dari pengguna. Jika pengguna mencentangnya maka item di kotak tersebut otomatis bisa dipakai untuk program.
7.	<i>Checked List Box</i>	Menampilkan item-item seperti <i>listbox</i> biasa, namun dengan <i>checkboxes</i> disebelah kirinya sehingga satu <i>listbox</i> bisa menerima dua item atau lebih.
8.	<i>Color Dialog</i>	Menampilkan kotak dialog untuk memilih warna dari palet standar.
9.	<i>Combo Box</i>	Menampilkan item-item namun dalam bentuk hanya satu yang terlihat. Jika pengguna mengklik <i>combobox</i> barulah semua item didalamnya akan terlihat.
10.	<i>Context Menu Strip</i>	Menampilkan menu <i>shortcut</i> jika pengguna mengklik kanan pada item.
11.	<i>Data Grid View</i>	Menampilkan data dalam bentuk tabel yang bisa dimodifikasi.
12.	<i>Data Set</i>	Merupakan representasi data dari sumber data di memori.
13.	<i>Data Time Picker</i>	Memungkinkan pengguna menampilkan tanggal dan memilihnya. Selanjutnya tanggal dan waktu dipilih dapat dimodifikasi menggunakan format tertentu.
14.	<i>Directory Entry</i>	Mengenkapsulasin node atau objek di hirarki <i>Active Directory</i> (AD) di <i>windows server</i> .
15.	<i>Directory Searcher</i>	Melakukan <i>query</i> terhadap <i>Active Directory</i> .
16.	<i>Domain Up Down</i>	Menampilkan nilai <i>string</i> yang ditampilkan pengguna dari daftar item dengan cara mengklik tombol up dan down.
17.	<i>Error Provider</i>	Menentukan dukungan untuk inisialisasi transaksi.

18.	<i>Event Log</i>	Menyediakan interaksi dengan <i>eventlog</i> dari windows.
19.	<i>File System Watcher</i>	Memonitor perubahan sistem file dan mengeksekusi <i>event</i> ketika ada perubahan direktori atau file.
20.	<i>Group Box</i>	Mengelompokkan control dalam satu kotak untuk mempermudah pengaturan dan manajemen antarmuka.
21.	<i>Image List</i>	Menampilkan daftar gambar.
22.	<i>Label</i>	Untuk menampilkan teks dimana nantinya isi didalamnya bisa diberi masukan oleh pengguna seperti halnya <i>textbox</i> .
23.	<i>List Box</i>	Menampilkan item-item dalam bentuk kotak di mana semua item ditampilkan di dalam daftar.
24.	<i>List View</i>	Menampilkan item-item dalam satu dari lima tampilan yang berbeda.
25.	<i>Picture Box</i>	Menampilkan file gambar seperti jpg, bmp, png dan ikon (.ico).
26.	<i>Radio Button</i>	Memungkinkan pengguna untuk memilih satu opsi dari beberapa pilihan yang dikaitkan dengan tombol radio lainnya.
27.	<i>Text Box</i>	Memungkinkan pengguna memasukan teks yang bisa lebih dari satu baris dan menyediakan kemampuan pengeditan dan <i>masking</i> kata sandi.

## II.8. Microsoft SQL Server

*Microsoft SQL Server* merupakan salah satu produk RDBMS. RDBMS memiliki kepanjangan *Relational Database Management System*. Merupakan salah satu produk andalan yang dibuat oleh microsoft yang berfungsi sebagai *relational database*. *Microsoft SQL Server* mendukung *SQL* sebagai bahasa Pemroses *query*. Seperti yang kita ketahui, *SQL* merupakan bahasa standar

internasional untuk proses *query*, database dan *SQL* ini sudah banyak sekali digunakan pada hampir semua aplikasi.

### **II.8.1. Hierarki Data Dalam Database**

*Hierarki* data dalam *database* mulai dari yang terbesar ke yang terkecil yaitu:

#### 1. *Database*

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

#### 2. *File*

Suatu *file* menggambarkan suatu kesatuan data yang sejenis, dimana kumpulan dari *file* membentuk suatu *database*.

#### 3. *Record*

Suatu *record* menggambarkan suatu unit data individu yang tertentu dimana kumpulan dari *record* membentuk suatu *file*.

#### 4. *Field*

Suatu *field* menggambarkan suatu attribute dari *record*, dimana kumpulan *field* membentuk *record*.

#### 5. *Byte*

Attribute dari *field* berupa huruf yang membentuk nilai dari sebuah *field*.

#### 6. *Bit*

Merupakan bagian terkecil dari data secara keseluruhan yaitu berupa karakter *ASCII* (*American Standar Code Form Information Interchange*).