

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Sistem**

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. (Konsep Dasar Sistem Informasi: 1) Teori sistem secara umum yang pertama kali diuraikan oleh Kenneth Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. (Tata Sutabri, 2012: 10)

Norman L. Enger dalam bukunya mengatakan bahwa subsistem adalah serangkaian kegiatan yang dapat ditentukan identitasnya yang berhubungan dalam suatu sistem. Gordon B Davis dalam bukunya mengatakan bahwa sistem terbagi atas beberapa subsistem-subsistem. Batasan dan penghubung atau antarface didalam suatu sistem ditelaah secara cermat untuk penjamin bahwa hubungan antar subsistem didefinisikan secara jelas dan bahwa jumlah semua subsistem merupakan keseluruhan sistem. (Tata Sutabri, 2012:12)

#### **II.1.1. Karakteristik Sistem**

Sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen—komponen sistem tersebut dapat berupa suatu bentuk subsistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*Maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini menjadi masukan bagi subsistem yang lain seperti sistem informasi.

### 7. Pengolahan Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

### 8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. (Tata Sutabri, 2012: 20)

## II.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang di antaranya:

#### 1. Sistem Abstrak dan sistem fisik

Sistem Abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

#### 2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam tidak dibuat oleh manusia, sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine sistem*.

#### 3. Sistem determinasi dan sistem probabilistic

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem *deterministic*, sedangkan sistem yang bersifat *probabilistic* adalah sistem

yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsure *probabilistic*.

#### 4. Sistem terbuka dan sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya, sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. (Tata Sutabri, 2012: 22)

## **II.2. Pengertian Informasi**

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Teori informasi lebih tepat disebut sebagai teori matematis dan komunikasi. Sumber informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu bentuk yang masih mentah yang belum dapat bercerita banyak sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi. (Tata Sutabri, 2012: 28)

### **II.2.1. Kualitas Informasi**

Kualitas informasi tergantung dari 3 hal, yaitu: informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*). Penjelasan tentang kualitas informasi tersebut akan dipaparkan dibawah ini.

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*timeline*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakaiannya. Relevansi informasi untuk orang satu dengan yang lain berbeda. (Tata Sutabri, 2012: 41)

### **II.3. Pengertian Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Tata Sutabri, 2012: 46)

#### **II.3.1. Komponen Sistem Informasi**

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), sebagai berikut:

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input yang dimaksud adalah metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi merupakan “*tool box*” dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data (*database block*)

Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras computer dan menggunakan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur

terjadi kesalahan-kesalahan dapat langsung cepat diatasi. (Tata Sutabri, 2012: 13)

#### **II.4. Pengertian Penggajian**

Variabel utama yang digunakan dalam penghitungan penggajian dalam sistem ini adalah absensi pegawai. Absensi didapat dari aplikasi *fingerprint*. Data absensi yang didapat dari sistem diolah dengan menggunakan absensi pegawai secara real yang dibandingkan dengan jadwal kehadiran pegawai yang sebelumnya telah direncanakan oleh supervisor dari masing-masing unit pada bulan sebelumnya. Dari sana bisa didapat informasi tentang presensi, keterlambatan, pulang lebih awal, jam kerja secara efektif dan lembur.

Data absensi yang telah didapat diolah lagi dengan menghitung tunjangan lembur, potongan keterlambatan, potongan ketidakhadiran serta ditambahkan dengan variabel gajipokok dan tunjangan yang sebelumnya telah ditetapkan oleh manajemen kepada pegawai tersebut berdasarkan tunjangan, masa kerja dan jabatan untuk mendapatkan nominal gaji pegawai. (Lukman Arif Sanjani, 2014: 89)

#### **II.5. Entity Relationship Diagram**

Pada dasarnya ERD (*Entity Relation Ship*) adalah sebuah diagram yang secara konseptual memetakan hubungan antara penyimpanan pada diagram DFD. ERD digunakan untuk melakukan pemodelan terhadap struktur data dan

hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

*Entity* dapat berarti sebuah objek yang dapat dibedakan dengan obyek lainnya, Obyek tersebut dapat memiliki komponen-komponen data (*atribut* atau *field*) yang membuatnya dapat dibedakan dari obyek lainnya, dalam database *entity* memiliki atribut yang menjelaskan karakteristik dari *entity* tersebut. Ada dua macam atribut yang dikenal dalam *entity* yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap *entity* memiliki himpunan yang diperlukan sebuah *primary key* untuk membedakan anggota-anggota dalam himpunan tersebut.

Atribut dapat memiliki sifat-sifat sebagai berikut:

1. *Atomic*, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi. Contohnya adalah *field* status dari tabel karyawan yang hanya berisi menikah atau *single*.
2. *Multivalued*, sifat ini menandakan atribut ini bisa memiliki lebih dari satu nilai untuk tiap *entity* tersebut. Misalnya adalah *field* hobi, hobi dari tiap karyawan mungkin dan hamper pasti lebih dari satu. Misalnya karyawan A hobi: membaca, nonton TV.
3. *Composite*, atribut yang bersifat *compost* adalah gabungan dari beberapa atribut yang bersifat *atomic*. Contohnya atribut alamat yang dapat dipecah menjadi atribut *atomic* berupa alamat, kode pos, no telepon. (Wahana Komputer, 2010:30).

## II.6. Normalisasi Data

Normalisasi data adalah proses dimana tabel-tabel pada database dites dalam hal kesalingtergantungan diantara *field-field* pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu *field* dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel. Banyaknya tabel pecahannya tergantung pada seberapa banyak ketergantungannya.

Ada beberapa langkah dalam normalisasi tabel, yaitu:

1. *Decomposition*, dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenal dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum didekomposisi
2. Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap *entity* masih ditampung dalam satu tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak rapi. Sebagai contoh tidak normal.

Tabel II.1 Bentuk Tidak Normal

Kd_agt	Nm_agt	Alamat	Kota	Telp	Tmp_lahir	Tgl_lahir	JK	Tgl_masuk

No_trans	Kd_agt	Kd_buku	Tgl_pjm	Tgl_kembali	TI_tempo	Status	Denda

Jml_pjm	Kd_buku	Tgl_Input	Jns_buku	Pengarang	Judul Buku	Penerbit

Thn_terbit	Harga	Tmp_buku	Kd_buku	jumlah

(Sumber: Wahana Komputer; 2010)

3. Normal *form* pertama (1<sup>st</sup> Normal *Form*), pada tahap ini tabel dikomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki *atribut* yang bernilai ganda dan komposit. Contohnya adalah tabel diatas yang dipisahkan menjadi 3 buah tabel anggota, tabel transaksi dan tabel buku.

Tabel II.2 Tabel Anggota

Kd_agt	Nm_agt	Alamat	Kota	Telp	Tmp_lahir	Tgl_lahir	JK	Tgl_masuk
A001	Herry	Krakatau	Smg	-	Smg	10/10/10	L	10/10/10
A002	Wati	Kali	Smg	-	Smg	11/11/11	P	11/11/11

(Sumber: Wahana Komputer; 2010)

Tabel II.3 Tabel Buku

Kd_bk	Tgl_Input	Jns_buku	Pengarang	Judul_buku	Penerbit
1.1.1	01/10/09	Fiksi	Acong	Akar pahit	Bunga
1.2.2	01/10/09	Fiksi	Aling	1,2 dan 3	Bunga

Thn_terbit	Harga	Tmp_buku	Ket_buku	Jumlah
2009	10000	01	Baru	5
2009	12400	01	Baru	4

(Sumber: Wahana Komputer; 2010)

Tabel II.4 Tabel Transaksi Pinjaman Buku

No_trans	Kd_agt	Kd_bk	Tgl_pjm	Tgl_kmbl	Tgl_tmp	Status	Denda	Jml_pjm
01	A001	1.1.1	01/11/09	03/11/09	05/11/09	P	0	1
02	A001	1.2.2	01/11/09	03/11/09	05/11/09	P	0	1

(Sumber: Wahana Komputer; 2010)

4. Normal *Form* Kedua (2<sup>nd</sup> Normal *Form*), pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut. Dalam hal ini semua tabel diatas telah memenuhi bentuk normal *form* yang kedua. Adapun kunci primer masing-masing tabel adalah sebagai berikut:
  - a. Untuk tabel anggota kunci primernya kode\_agt
  - b. Untuk tabel buku kunci primernya kd\_bk
  - c. Untuk tabel transaksi kunci primernya no\_trans
5. Normal *Form* Ketiga (3<sup>rd</sup> Normal *Form*), setiap *atribut* pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. Biasanya penggunaan bentuk normal (normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Semua tabel diatas juga telah memenuhi bentuk normal tahap ketiga.(Wahana Komputer: 2010:32)

## **II.7. Database**

*Database* atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan computer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. Database sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. Misalnya dari data nama siswa yang dan tanggal lahir siswa. Anda bisa mendapatkan informasi

nama siswa yang berulang tahun pada hari ini. Dalam database ada sebutan-sebutan untuk satuan data yaitu:

1. Karakter adalah satuan data terkecil. Data terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
2. *Field* adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir, dan lain-lain.
3. *Record* adalah kumpulan dari *field*.
4. Tabel adalah sekumpulan dari *record-record* yang memiliki kesamaan *entity* dalam dunia nyata. Kumpulan dari tabel adalah database, wujud fisik sebuah database dalam komputer adalah sebuah *file* yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan diatas.(Wahana Komputer, 2010: 24)

## II.8. SQL Server 2008

*Microsoft SQL Server* merupakan produk RDBMS (*relational database management system*) yang dibuat oleh Microsoft. *Microsoft SQL Server* juga mendukung SQL sebagai bahasa untuk memproses *query* ke dalam database. *Microsoft SQL Server* banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi database atau penyimpanan data. *SQL Server* sudah memiliki banyak versi yaitu *SQL Server*, *SQL Server 4.2*, *SQL Server 6.0*, *SQL Server 6.5 Enterprise Edition*, *SQL Server 7.0*, *SQL Server 2000*, *SQL Server 2005*, *SQL server 2008 R2* dan lain-lain.

*SQL Server 2008 Express Edition* merupakan sebuah versi *cut-down* (versi terbatas dan gratis tentunya) dari *software SQL Server 2008* unggulan *Microsoft* yang terkenal. Adapun tampilan awal *SQL Server 2008 R2*:



Gambar II.1 Tampilan *Setting New User SQL Server 2008 R2*  
(Sumber: Wahana Komputer; 2010:40)

*SQL Server Management Studio* membantu anda mengatur database dengan mudah. Anda dapat melakukan pengaturan atas beberapa server pada sebuah computer saja atau melakukan pengaturan server secara remote. Anda dapat juga membuat database, table, index dan melakukan manipulasi data terhadap database dan tabel-tabelnya. *SQL Server Management Studio* memiliki beberapa komponen penting yang mewakili kegunaannya dalam perancangan database, dan melakukan pengaturan sistem secara keseluruhan. Komponen-komponen tersebut adalah:

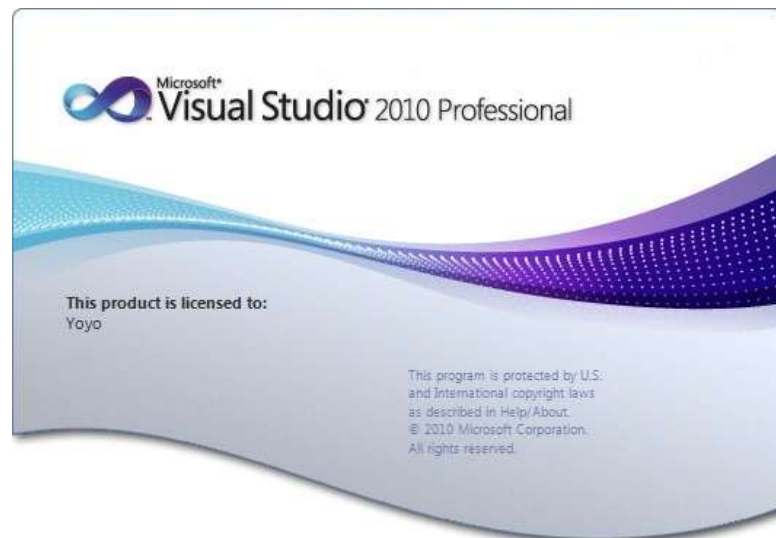
- a. Registered Server
- b. Object Explorer
- c. Query Editor. (Wahana Komputer ; 2010: 40)

## II.9. VB.net 2010

*Visual Studio* 2010 merupakan merupakan sebuah lingkungan kerja (*Integrated Development Environment (IDE)*) yang digunakan untuk pemrograman .NET yang dapat digunakan untuk beberapa bahasa pemrograman, seperti *Visual Basic (VB)*, *C# (baca C Sharp)*, *Visual C++*, *J# (baca J sharp)*, *F# (baca F Sharp)*, dan lain-lain. (Wahana Komputer ; 2012:2)

Teknologi .NET *Framwork* adalah sebuah *Application Programming Language (API)*, yaitu kumpulan kelas atau sebuah pustaka inti yang digunakan untuk melakukan pemrograman .NET. Kelas-kelas *core* (inti) .NET ini menyediakan berbagai macam kelas yang berfungsi untuk melakukan apapun yang diinginkan dilingkungan *windows*, ataupun lingkungan web, mulai dari bekerja dengan data hingga bekerja dengan *form* (jendela) dan kontrol.(Wahana Komputer ; 2012:2)

*Visual Studio Profesional* 2010 menyediakan berbagai *tool* yang lengkap bagi para pengembang untuk membangun aplikasi yang berjalan di .NET *Framwork*. Berbagai *tool*, antara lain *Toolbox* yang berisi komponen *visual*, sehingga anda tinggal *drag and drop* komponen dan *Visual Studio* 2010 akan menuliskan kode untuk anda.(Wahana Komputer ; 2012:7)



Gambar II.2 *Splash Screen Visual Studio 2010*  
Sumber : (Wahana Komputer; 2012 :17)

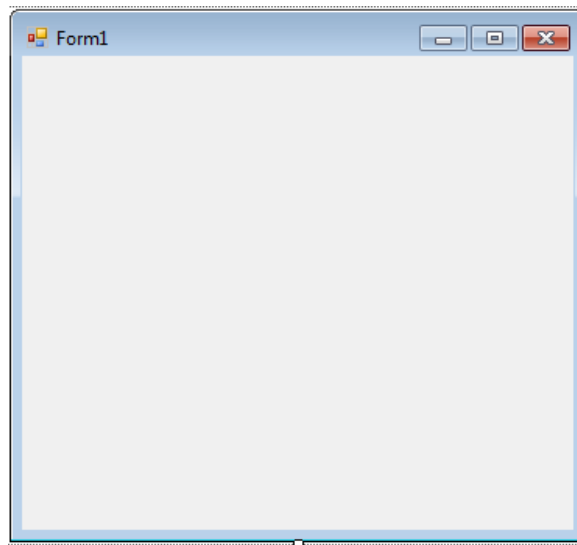


Gambar II.3 *IDE Visual Studio 2010*  
Sumber : (Wahana Komputer; 2012 :17)

Adapun objek-objek yang digunakan dalam program ini :

### 1. *Form*

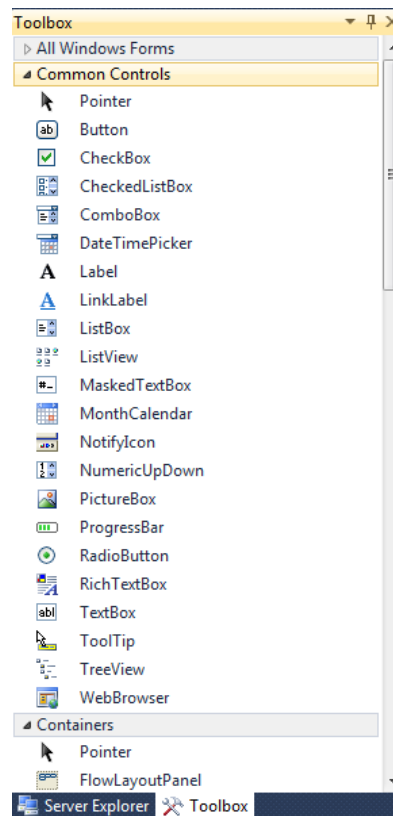
*Form* merupakan komponen VB yang memiliki sifat container, karena fungsi utama dari form adalah sebagai tempat komponen VB yang lain. Pembuatan form dalam VB dapat kita lakukan dengan menggunakan menu **Project> Add Windows Form**.



GambarII.4 Form Design  
Sumber : (Wahana Komputer; 2012 :16)

## 2. *ToolBox*

*ToolBox* merupakan jendela yang berisikan grup dari bermacam –macam control dan komponen yang dapat dipasang di dalam form. Kontrol seperti *Textbox*, *Button*, *Radio Button*, *Checkbox*, *Combo Box* dan lain-lain ditambahkan ke dalam *form* dengan *cara drag* dan *drop*, atau mengklik dan menggoreskannya pada *form*.



Gambar II.5 *ToolBox*

Sumber : (Wahana Komputer; 2012 :17)

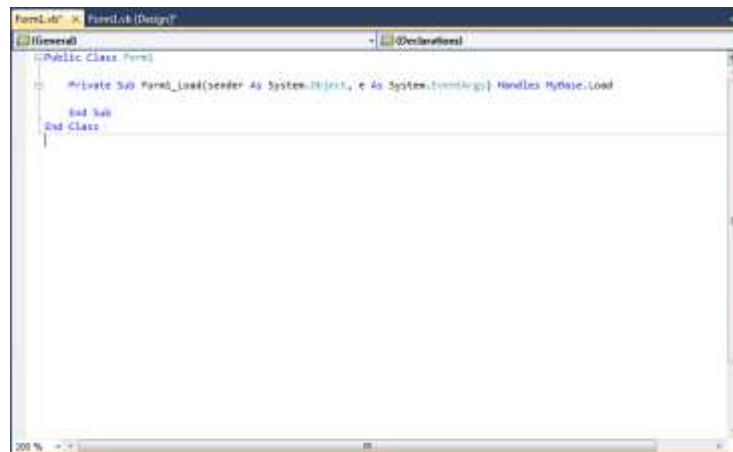
Beberapa kontrol pada ToolBox :

- a. *Button* :komponen yang digunakan untuk memberikan aksi saat ditekan.*Button* menjalankan proses menyimpan, mengubah, menghapus, dan lain-lain.
- b. *Label* :untuk menampilkan teks. Biasa digunakan untuk memberikan informasi pada kontrol lain.
- c. *Textbox* : untuk input data. Komponen ini paling sering digunakan bersama button.
- d. *CheckBox* : digunakan untuk memberikan pilihan *input* kepada *user* dapat memilih lebih dari 1 item data.
- e. *Combobox* : komponen yang menampilkan pilihan secara drop down.

- f. *ListBox* :sebuah kotak yang di dalamnya berisi item-item, *ListBox* menampilkan item lebih dari 1.
- g. *RadioButton* : untuk memilih pada suatu daftar pilihan, dengan satu pilihan yang dapat dipilih.(Wahana Komputer,2012:114)

### 3. Jendela Editor Kode

Agar kontrol dapat bekerja sesuai fungsinya dan interaktif, Anda harus menambahkan sebuah kode di belakang layar control tersebut.Untuk menulis kode ini anda harus masuk ke dalam jendela editor kode. Untuk membukanya, klik 2 kali control yang akan ditambahi kode.(Wahana Komputer ;2012:19)



Gambar II.6 Jendela Editor Kode  
Sumber : (Wahana Komputer; 2012 :19)

## II.10. UML

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2003:bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti.( Menggunakan UML; 2011:1)

Blok pembangun utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasi sistem yang mereka rancang. UML merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. (Menggunakan UML; 2011:1)

### II.10.1. Diagram – diagram UML




Beberapa literature menyebut bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung. Jenis diagram itu antara lain:

1. Diagram kelas. Bersifat statis.
2. Diagram Paket (*Package Diagram*). Bersifat statis.
3. Diagram *Use-Case*. Bersifat statis.
4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis.
5. Diagram Komunikasi (*Communication Diagram*). Bersifat dinamis.
6. Diagram Statechart (*Statechart Diagram*). Bersifat dinamis.
7. Diagram Aktivitas (*Activity Diagram*). Bersifat dinamis.
8. Diagram Komponen (*Component Diagram*). Bersifat statis.
9. Diagram *Deployment* (*Deployment Diagram*). Bersifat statis. (Menggunakan UML; 2011:3)

### II.10.2. Diagram Use Case

John Satzinger, 2010, dalam buku *System Analysis and Design in a Changing World* menyatakan bahwa “Use Case adalah sebuah kegiatan yang dilakukan oleh sistem biasanya dalam menanggapi permintaan dari pengguna sistem.”(Haviluddin dan Gede Suardika; 2011:6)

Tabel II.5 Tabel Simbol Use Case

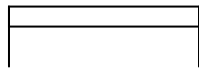
Notasi	Keterangan	Simbol
<i>Actor</i>	Stick Figure mewakili sebuah peran	
Garis Penghubung	Menunjukkan actor mana yang menjalankan use case yang mana	
<i>Use Case</i>	Bentuk fungsionalitas dari sebuah system	







(Sumber :Haviluddin dan Gede Suardika; 2011:4)

### II.10.3. Activity Diagram

John Satzinger, 2010, dalam buku *System Analysis and Design in Changing World* menyatakan bahwa “Activity Diagram adalah sebuah Diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), Orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.”(Haviluddin dan Gede Suardika; 2011:4 )

Tabel II.6 Tabel Simbol Activity Diagram

Notasi	Keterangan	Simbol
<i>Swimlane</i>	Mewakili agen yang melakukan aktivitas	


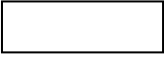
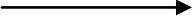
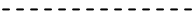
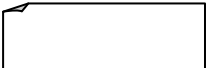
<i>InitialState</i>	Awal dari alur kerja	
<i>FinalState</i>	Akhir dari alur kerja	
<i>ActionState</i>	Aktivitas tersendiri dalam alur kerja	
<i>Decision</i>	Titik pengambil keputusan di mana aliran proses tersebut akan mengikuti satu jalur atau jalur lainnya	
<i>Transition</i>	Urutan diantara aktivitas	
<i>Synchronization</i>	Membagi alur kerja menjadi beberapa alur yang berbarengan ataupun menggabungkan lagi alur yang berbarengan	

(Sumber :Haviluddin dan Gede Suardika;2011:4)

#### II.10.4. Sequence Diagram

Menurut John Satzinger, 2010 dalam buku *System Analysis and Design in a Changing World* menyatakan bahwa “*System Sequence Diagram (SSD)* adalah diagram yang digunakan untuk mendefinisikan input dan output serta urutan interaksi antara pengguna dan sistem untuk sebuah *use case*.”(Haviluddin dan Gede Suardika; 2011:5)

Tabel II.7 Tabel Simbol *Sequence Diagram*

Notasi	Keterangan	Simbol
Actor	Peran yang berinteraksi dengan sistem	
Kotak Berlabel	Objek yang mewakili keseluruhan sistem yang terotomalisasi	
Anak Panah	Mewakili message yang dikirim atau diterima oleh actor dari sistem	
Garis Putus-putus Vertikal	Perpanjangan objek tersebut, baik actor maupun objek, sepanjang durasi dari sequence diagram	
Message diberi label	Menggambarkan maksud message dan input apapun yang sedang dikirim	

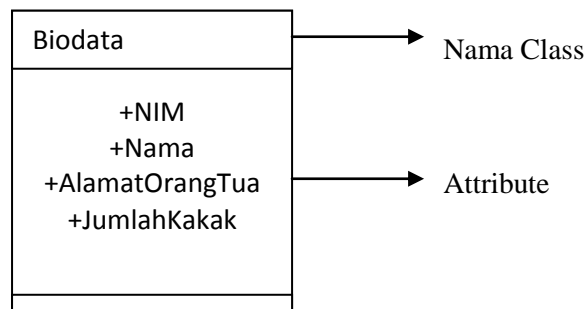
(Sumber : Havaluddin dan Gede Suardika, 2011)

### II.10.5. Class Diagram

John Satzinger, 2010 dalam buku *System Analysis and Design in a Changing World* menyatakan bahwa “dalam UML ada dua jenis *class diagram* yaitu: domain *class diagram* dan *design class diagram*.”

#### 1. Domain Class Diagram

Menurut Satzinger “Domain *Class Diagram* adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada *class* perangkat lunak yang nantinya akan anda rancang.

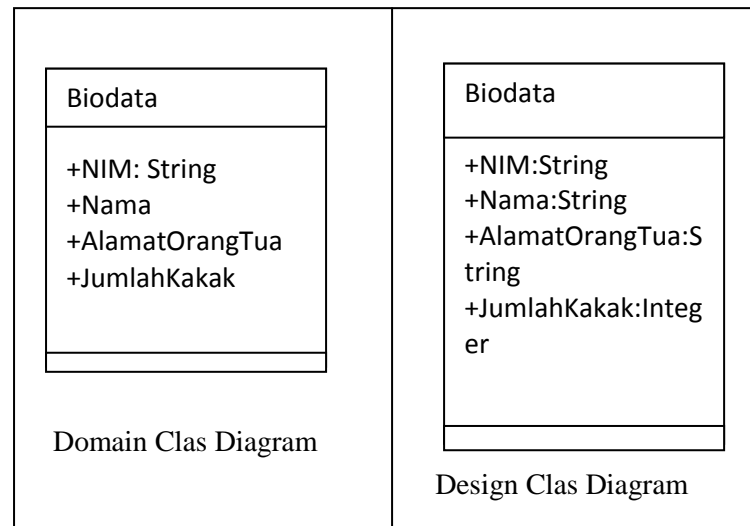


Gambar II.7 Notasi *Domain Class Diagram*

Sumber : (*Step By Step Desain Proyek Menggunakan UML*;2011:3)

#### 2. Design Class Diagram

Tujuan utamanya adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun. *Design class diagram* menggambarkan kelas berorientasi objek yang dibutuhkan dalam pemrograman, navigasi diantara kelas *attribute* name dan propertinya serta *method names* dan propertinya.(Havaluddin dan Gede Suardika;2012: 50)



Gambar II.8 Class Biodata yang digambarkan dengan *Domain Class Diagram* dan *Design Class Diagram*

Sumber : (*Step By Step* Desain Proyek Menggunakan UML; 2011: 3)