

2. TEORI PENUNJANG

2.1. Portal

Menurut Goldman Sach, model aplikasi web terbagi menjadi enam macam, antara lain :

- *Connectivity* : adalah model aplikasi web yang menyandarkan basis layanan pada akses internet kepada pelanggannya seperti yang dijalankan oleh ISP (*Internet Server Provider*). Contoh di Indonesia : Indosat.net, wasantara.net, CBN, Rad.net, Link.net, Indo.net, IPNET, dan masih banyak yang lain.
- *Context* : adalah model aplikasi web yang memberikan layanan berupa informasi dan hiburan. Seperti Kompas Cybermedia, Detik.com, Astaga.com, Lippostar.com, dan lain sebagainya.
- *Content* : adalah model aplikasi web yang memberikan layanan dengan basis berupa teks atau gambar sebagai inti aplikasi web. Seperti marketbiz.net, bizweb2000.com, DotNetArea.com.
- *Communication* : adalah layanan komunikasi berbasis internet dengan menggunakan media interaktif seperti media *web chat* maupun *voice chat*, *web phone* dan lain sebagainya.
- *Community* : adalah model aplikasi web yang membangun suatu komunitas tertentu dengan menggunakan media seperti *mailing list*, *message board*, *web chat*, maupun media penyedia *web mail*.
- *Commerce* : adalah model aplikasi web yang diperuntukan melakukan transaksi jual beli yang berlangsung secara online. Contoh aplikasi web commerce : clikbookshop.com, sanur.com, gado-gado.com, dan lain sebagainya.

Hanya saja pada saat ini banyak aplikasi web yang tidak hanya terdiri dari satu jenis model, melainkan gabungan dari beberapa model. Yang sangat umum dijumpai adalah aplikasi web portal seperti yahoo.com, astaga.com, dan lain sebagainya.

Secara umum web portal terdiri dari gabungan antara model *context*, *content*, serta *community*.

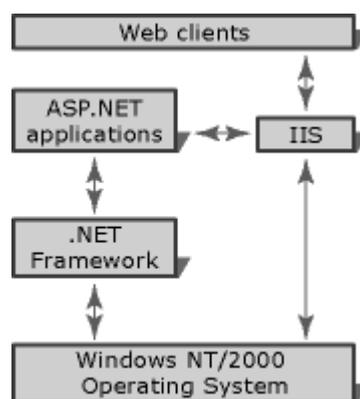
2.2. ASP.Net

ASP.Net adalah salah satu bahasa pemrograman yang bertujuan untuk membangun suatu aplikasi web. Bahasa pemrograman ini merupakan kelanjutan dari Active Server Pages (ASP). ASP.Net sebenarnya merupakan konsep baru dalam pemrograman aplikasi web. Di mana ASP.Net memberikan platform pemrograman yang jauh lebih bagus, lebih aman, lebih stabil dan memiliki banyak kemudahan.

ASP.Net adalah bahasa yang terkompilasi. Berbasiskan .Net Framework, sehingga dapat menggunakan beberapa bahasa pemrograman yang mendukung dengan .Net Framework seperti : Visual Basic .Net, C#, dan Jscript.

2.2.1 Arsitektur ASP.Net

Untuk dapat menjalankan ASP.Net, selain dibutuhkan program ASP.Net, .Net Framework, juga dibutuhkan Web Server (IIS). Untuk lebih memperjelas bagaimana hubungannya, di bawah ini adalah gambar arsitektur ASP.Net :



Gambar 2.1. Arsitektur ASP.Net

Pada gambar 2.1, arsitektur ASP.Net memiliki bagian-bagian yaitu :

- Web Clients** : Client meminta request kepada server. Pada umumnya software yang digunakan adalah web browser seperti IE, Netscape, dll.
- Aplikasi ASP.Net** : adalah aplikasi yang dibuat dengan menggunakan ASP.Net.

- c. IIS : adalah Web Server yang terinstal pada server tempat aplikasi dijalankan.
- d. .Net Framework : adalah model platform yang memudahkan dalam membangun lingkungan aplikasi yang terdistribusi di Internet. Platform .Net Framework terdiri dari dua bagian utama :
 1. Common Language Runtime (CLR)

CLR berfungsi untuk mengatur memori, mengatur *thread*, menjalankan kode, verifikasi keamanan kode, kompilasi, dan *system service* yang lain.
 2. .Net Framework Class Library (FCL):

.Net Framework Class Library berisi koleksi class yang dalam penggunaannya dapat diturunkan menjadi koleksi objek. Objek ini dapat digunakan untuk membangun aplikasi. Seperti menggunakan Web Forms, atau XML Web Service.

Untuk dapat menggunakan objek tersebut, dibutuhkan namespace yaitu deklarasi yang menghubungkan ke .Net Framework.

Contoh Deklarasi namespace :

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>
```

Namespace di atas digunakan untuk dapat menggunakan class pada .Net Framework yang berhubungan dengan pengaksesan database.

Untuk daftar namespace yang digunakan secara umum pada pemrograman ASP.Net dapat dilihat pada lampiran belakang buku.
- e. Operating System : Operating System yang mendukung adalah Windows NT / 2000 /XP.

2.2.2 Perbedaan ASP.Net dengan ASP

ASP.Net bukan hanya versi baru tetapi merupakan konsep baru dalam pemrograman aplikasi web. ASP.Net memberikan platform pemrograman yang jauh lebih bagus dan memiliki banyak keuntungan. Sedangkan untuk extension terdapat perbedaan yaitu pada ASP.Net menggunakan ekstensi baru yaitu .aspx. Di mana jika ada request file dengan extension .aspx maka akan diproses sesuai dengan pemrograman ASP.Net.

2.2.2.1 Penggunaan .Net Framework.

Keunggulan ASP.Net dibandingkan ASP dalam membuat aplikasi yang memiliki tingkat kompleksitas yang besar, jelas sangat berat bila developer hanya menggunakan ASP saja. Maka untuk membantu memrogram, *programmer* menggunakan komponen tambahan ataupun library yang telah mereka buat sebelumnya.

Hal ini jelas membutuhkan waktu yang cukup lama bagi mereka yang belum memiliki pengalaman dalam menggunakan komponen ataupun belum memiliki *library* yang banyak. Dengan ASP.Net masalah ini terpecahkan dengan adanya kumpulan library yang sangat banyak dari .Net Framework. Di mana developer tinggal menggunakan semudah “*drag and drop*” seperti bahasa pemrograman Visual.

2.2.2.2 Terkompilasi.

Selain itu ASP.Net menggunakan konsep kompilasi dengan menggunakan bahasa pemrograman Visual Basic atau Visual C# sedangkan ASP masih menggunakan interpreter dengan menggunakan bahasa VBScript atau JScript. Dari masalah bahasa penulisan jelas Visual Basic jauh lebih kaya akan fungsi dibandingkan dengan saudara mudanya yaitu VBScript atau JScript.

Interpreter dalam mengeksekusi perintah langsung pada saat ada request dari client dan pada saat itu kode diubah ke dalam bahasa mesin. Setelah itu, baru perintah akan request tersebut dijalankan. Sedangkan kompilasi langsung mengeksekusi bila ada request dari client yang masuk. Sehingga dalam hal kecepatan kompilasi jauh lebih cepat dari interpreter.

Kekurangan lain dari interpreter adalah kurangnya *strong type variable* dalam memrogram. Di mana Visual Basic dan C/C++ mendukung akan strong type. Hal yang lain adalah kurangnya fasilitas pendukung dari lingkungan kompilasi. Hal ini tentunya mengurangi tingkat performa dan skalabilitas program yang menggunakan ASP dibandingkan dengan ASP.Net.

2.2.2.3 Model Event

Pada umumnya terdapat dua model untuk mengeksekusi program yaitu secara *linier* dan *event-driven*.

Model program Linier adalah pengaksesan kode dari baris perbaris yaitu mulai dari baris 1 kemudian baris 2 dan seterusnya sampai habis. Sehingga untuk membuat program juga perlu memperhitungkan posisi / letak kode program yang hendak ditulis.

Seperti *function*, *loop*, *keyword if* atau yang lainnya, perlu diperhitungkan apakah posisinya tepat berhubungan dengan kode yang lain atau sebaliknya dapat mengganggu kode yang lain. ASP 3.0 ini menggunakan model seperti ini.

Model pemrograman *event-driven* berbeda dengan linier, model ini bergantung pada event yang dipanggil. Seperti user menekan tombol, maka terjadilah event. Event yang terjadi ini perlu ada yang menangani atau yang disebut sebagai *event handler*.

Model pemrograman seperti ini, sangat memudahkan bagi programmer untuk menuliskan kode. Karena programmer tidak perlu pusing memikirkan pada baris mana menulis kode agar tidak mengganggu kode yang lain. Penulisan kode pada model ini terbagi-bagi berdasarkan event. Sehingga programmer hanya menuliskan kode program pada suatu event yang nantinya akan terjadi.

Contoh model pemrograman *event driven* ini adalah Visual Basic, ASP.Net dan bahkan windows sendiri menggunakan model ini.

2.3. HTML

Dalam memrogram ASP.Net sangat diperlukan untuk mengerti HTML. Di mana antara sintaks ASP.Net dengan HTML akan bergabung menjadi satu. HTML atau Hypertext Markup Language adalah bahasa yang dipakai untuk menampilkan informasi pada halaman web. Hal ini karena HTML menampilkan informasi dalam bentuk hypertext dan bukan itu saja, HTML mendukung bagaimana informasi tersebut akan ditampilkan.

Sesuai namanya, bahasa ini menggunakan tanda (markup) untuk menandai perintah-perintahnya. HTML terdiri dari elemen dan atribut dari elemen. Di mana elemen dan atribut tersebut digunakan untuk mengatur tampilan web. Istilah untuk menyebut elemen HTML dapat juga disebut sebagai tag.

Untuk dapat membuat suatu halaman html, diperlukan susunan tag seperti di bawah ini :

```
<HTML>
  <HEAD>
    <TITLE> Judul dari halaman page yang ditulis </TITLE>
  </HEAD>
  <BODY>
    Isi Halaman
  </BODY>
</HTML>
```

Tag Utama adalah <HTML> ... </HTML> yang menandai awal dan akhir dari halaman HTML. Kemudian didalam tag ini terdapat dua bagian yaitu kepala halaman dan badan halaman. Kepala halaman dideklarasikan dengan menggunakan tag <HEAD> ... </HEAD>. Tag ini bertujuan untuk menyediakan informasi mengenai halaman HTML, seperti judul <TITLE> ... </TITLE>, versi, revisi dan sebagainya.

Bagian kedua adalah badan halaman atau body. Pendeklarasiannya menggunakan tag <BODY> ... </BODY>. Pada bagian inilah isi dari halaman web yang hendak ditampilkan berada di sini.

Struktur di atas adalah struktur tetap dari setiap halaman HTML yang akan dibuat. Jika salah satu tag tidak ditulis / hilang, maka akan muncul tampilan error.

Selain tag di atas, HTML memiliki banyak tag lain yang merupakan optional (pilihan) yang tidak wajib dituliskan. Tag tersebut akan ditambahkan / ditulis jika diperlukan oleh pembuat halaman HTML tersebut.

2.3.1 Heading

Heading pada halaman HTML digunakan untuk menampilkan nama bab, atau bagian tulisan yang dianggap penting atau ingin tampak jelas. Pada HTML terdapat beberapa macam heading yaitu mulai dari tag <H1> sampai dengan <H6>.

Untuk mendapatkan ukuran font yang besar dapat menggunakan heading pada HTML. Ukuran dari tiap heading ini berbeda. Heading terbesar adalah heading <H1> dan terkecil adalah <H6>.

2.3.2. Font

Pada HTML untuk dapat mengatur ukuran, warna, jenis font, dapat menggunakan tag dan . Tag tersebut diletakkan pada awal dan akhir dari font yang ingin atur.

Untuk dapat mengatur besar kecilnya font, dapat mengisi atribut size dengan suatu nilai. Nilai berkisar mulai dari 1 terkecil sampai dengan 7 sebagai ukuran terbesar.

2.3.3 Paragraf

Bagian body merupakan tempat untuk menuliskan isi dari halaman HTML yang hendak dibuat. Untuk menuliskan isi halaman HTML, diperlukan paragraf-paragraf yang tersusun rapi agar memudahkan untuk dibaca. Untuk membuat paragraf dibutuhkan tag <P> dan </P> pada awal dan akhir paragraf.

2.3.4 Gambar

Pada halaman web sangat umum untuk mempergunakan gambar dalam memperjelas informasi yang ingin disampaikan. Gambarpun sekaligus dapat mempercantik halaman web tersebut. Pada HTML dalam menampilkan gambar mempergunakan tag . Yaitu seperti tag berikut ini .

Secara umum tipe gambar yang ditampilkan pada web berektensi GIF, JPG dan BMP. Di mana ukuran dari file tersebut relatif kecil sehingga proses akses halaman web dapat lebih cepat.

2.3.5 Link

Pada halaman web terdapat cara untuk menghubungkan antara satu halaman web dengan halaman web yang lain. Untuk dapat melakukan hal tersebut dipergunakan tag <A> dan . Pemakaian tag ini diikuti dengan atribut HREF

sebagai alamat dari halaman web yang dituju. Berikut contoh membuat sebuah link :

```
<a href="Target.html"> Arsitektur ASP.Net </a>
```

Gambar pun dapat dibuat link, di mana apabila user menekan pada gambar akan berpindah ke halaman yang dituju. Berikut contoh membuat sebuah link dengan gambar :

```
<a href="Target.html"></a>
```

2.3.6 Tabel

Untuk dapat menata informasi dapat mempergunakan tabel. Tabel berfungsi untuk membagi-bagi halaman web. Kemudian pada setiap bagian tersebut informasi diletakkan. Untuk dapat membuat Tabel diperlukan tag `<TABLE>` `</TABLE>`.

Dalam membuat tabel tentunya akan dibagi-bagi berdasarkan baris dan kolom. Untuk membuat baris diperlukan tag `<TR>` dan `</TR>` dan untuk membuat kolom diperlukan tag `<TD>` dan `</TD>`.

2.3.7 Form

Dalam memrogram dengan ASP.Net, kita akan banyak berhubungan dengan Form. Salah satu server control dari ASP.Net yaitu Web Forms selalu menggunakan form dalam setiap aplikasinya.

Form pada HTML berguna untuk membuat formulir di halaman web. Form berkerja dengan cara mengambil data yang telah dimasukkan oleh user. Setelah data masuk, data akan diproses dengan menggunakan ASP.Net.

Form dalam pembuatannya menggunakan tag `<FORM>` dan `</FORM>`.

Berikut adalah struktur dasar dari form :

```
<form name="form1" method="post" action="www.DotnetArea.com">
...
...
</form>
```

Pada tag `<FORM >` memiliki beberapa atribut seperti name, method, dan action.

Berikut penjelasan kegunaan dari atribut di atas :

- Action : digunakan sebagai alamat / URL tempat pemrosesan form selanjutnya.

- Method : digunakan untuk menentukan bagaimana form diberlakukan, yaitu dengan metode Get atau Post.
- Name : adalah nama dari form. Nama dari form dapat diisi dengan bebas.

Seperti penjelasan di atas atribut method berfungsi untuk menentukan bagaimana form akan diberlakukan. Apabila menggunakan Submit Button, pada waktu menekan tombol maka form akan dikirimkan ke Web Server tujuan. Alamat dari web server ada di atribut action.

Untuk dapat mengetahui bagaimana cara form mengirim data maka hal ini berhubungan dengan atribut method yang terdapat pada form. Form memiliki dua macam metode yaitu metode Get dan Post.

Apabila menggunakan metode Get maka browser akan mengirim perintah HTTP GET ke server dengan data yang disisipkan menjadi satu di URL header. Sehingga pada header URL akan tampil seperti contoh di bawah ini :

```
http://www.DotNetArea.com/asp/list.aspx?article_id=12
```

Jika mengirim dengan metode POST maka form akan mengirim data ke server dengan menggunakan perintah HTTP POST. Untuk data yang hendak dikirimkan berada pada badan HTTP request atau body HTTP request. Sedangkan pada URL tidak tampak data yang telah dimasukkan tersebut. Berikut ini contoh menggunakan metode post

```
http://www.DotNetArea.com/asp/list.aspx
```

Jelas metode POST akan lebih cocok digunakan bila tidak menginginkan nilai yang terkandung tidak terlihat pada URL, tetapi jika tidak memperlmasalahkan maka dapat menggunakan metode GET.

Pada form dapat ditambahkan beberapa input seperti berupa Button, Radiobutton, CheckBox, ListBox dan sebagainya. Untuk dapat membuat input di atas dipergunakan tag <INPUT >. Strukturnya adalah :

```
<input type="TipeInputan" name="NamaInputan">
```

Kesemua input menggunakan tag yang sama yaitu <INPUT>. Hanya berbeda untuk jenis typenya. Jika menginginkan untuk membuat button, maka typenya adalah Submit atau Reset. Sedangkan untuk RadioButton, typenya adalah Radio.

Berikut ini adalah penjelasan secara lebih mendetail dari masing-masing input yang ada.

2.3.7.1 Button

Pada HTML terdapat dua macam button, yaitu Submit dan Reset. Submit bertujuan untuk mengirim data yang telah dimasukkan oleh user ke alamat di atribut Action. Sedangkan Reset berfungsi untuk menghilangkan semua nilai yang telah diisi pada form.

Untuk membuat input Button, pada tag <INPUT> atribut type diisi dengan Submit atau Reset. Maka akan tampil sebuah Button di browser. Berikut contoh sederhana membuat Button :

```
<form name="form1" method="post" action="www.DotNetArea.com">
  <input type="submit" name="Submit" value="Submit">
  <input type="Reset" name="Submit" value="Reset">
</form>
```

Pada HTML apabila button submit ditekan, maka proses selanjutnya akan menuju ke alamat : www.DotNetArea.com. Sedangkan jika tombol reset ditekan, maka akan menghapuskan nilai input yang lain. Pada bagian ini tidak ada input yang lain sehingga seakan-akan tidak terjadi apa-apa.

2.3.7.2 CheckBox

CheckBox adalah suatu input bagi user untuk memilih atau tidak. Dalam hal ini pilihan berbentuk centang. Tidak seperti RadioButton, CheckBox memperbolehkan user memilih lebih dari satu pilihan atau bahkan diperbolehkan untuk tidak memilih pilihan yang tersedia. Sedangkan pada RadioButton user hanya boleh memilih hanya salah satu pilihan saja.

Untuk membuat input CheckBox, pada tag <INPUT> atribut type diisi dengan checkbox. Maka akan tampil sebuah CheckBox di browser. Berikut contoh sederhana membuat CheckBox :

```
<form name="form1" method="post" action="">
  <input type="checkbox" name="Bahasa" value="ASP">
  Menggunakan ASP / ASP.Net
</form>
```

Jika menggunakan beberapa CheckBox, untuk menggolongkan menjadi satu dapat memberi nama atribut namanya menjadi satu nama yang sama.

Sedangkan value dapat berbeda-beda tergantung dengan keterangan yang ditulis pada samping CheckBox.

2.3.7.3 InputBox

InputBox adalah input yang memungkinkan user untuk memasukkan string atau angka. Untuk membuat input InputBox, pada tag <INPUT> atribut type diisi dengan text. Maka akan tampil sebuah InputBox di browser. Berikut contoh sederhana membuat InputBox :

```
<form name="form1" method="post" action="">
  <input type="text" name="InputBox1" value="nilai" size="10">
</form>
```

Atribut yang umum digunakan adalah name yaitu untuk menyatakan nama dari input tersebut. Atribut value untuk menyatakan isi dari InputBox. Sedangkan atribut size digunakan untuk mengatur panjang isi dari InputBox.

Apabila menginginkan agar nilai isi dari InputBox yang pada waktu diketik tidak terlihat atau digantikan dengan karakter *, dapat dilakukan dengan mengisi atribut type dengan password.

```
<form name="form1" method="post" action="">
  <input type="Password" name="InputBox1" value="nilai"
  size="10">
</form>
```

2.3.7.4 RadioButton

RadioButton adalah suatu input bagi user untuk memilih pilihan yang telah tersedia. RadioButton hanya memperbolehkan user untuk memilih satu pilihan saja tidak dapat dua atau lebih pilihan sekaligus. Untuk membuat input RadioButton, pada tag <INPUT> atribut type diisi dengan radio. Maka akan tampil sebuah RadioButton di browser. Berikut contoh sederhana membuat RadioButton:

```
<form name="form1" method="post" action="">
  <input type="radio" name="radiobutton" value="Item1">
  Item 1<br>
  <input type="radio" name="radiobutton" value="Item2">
  Item 2<br>
  <input type="radio" name="radiobutton" value="Item3">
  Item 3
</form>
```

Selain menggunakan tag <INPUT>, dapat pula menggunakan tag yang lain dalam membuat suatu input di form. Input yang lain seperti ListBox, Input File, Jump Menu dan sebagainya.

2.4. Bahasa Visual Basic.Net

Dalam melakukan pemrograman, ASP.Net mendukung secara *built-in* 3 jenis bahasa pemrograman yaitu : C#, Visual Basic.Net, dan JScript. Tetapi bahasa pemrograman yang akan dibahas dalam buku ini tidak semuanya melainkan hanya bahasa Visual Basic .Net saja.

2.4.1 Tipe Data

Pada tabel 2.1 menunjukkan tipe data pada Visual Basic .Net, jumlah ukuran yang disimpan serta batasan nilai untuk masing-masing tipe data.

Tabel 2.1 Jenis Tipe Data di Visual Basic .Net

Tipe Data	Alokasi Penyimpanan	Batasan Nilai
Boolean	2 bytes	Tipe Boolean memiliki 2 jenis nilai yaitu True atau False.
Byte	1 byte	Tipe Byte adalah bilangan integer dengan jangkauan 0 sampai 255.
Char	2 bytes	Tipe Char adalah bilangan integer dengan jangkauan 0 sampai 65535.
Date	8 bytes	Tipe Date ini mampu menyimpan mulai dengan 1 Januari 0001 jam : 0:00:00 Sampai 31 December 9999 jam : 11:59:59
Decimal	16 bytes	Tipe Decimal memiliki 3 jenis jangkauan nilai yaitu : <ul style="list-style-type: none"> • 0 Sampai +/- 79.228.162.514.264.337.593.543.950.3 35 (tanpa nilai desimal)

		<ul style="list-style-type: none"> • 0 Sampai +/- 7,9228162514264337593543950335 (dengan 28 tempat sebelah kanan dari desimal) • Nilai yang lebih kecil dari 0 : +/- 0,00000000000000000000000000000001 (+/-1E-28).
Double	8 bytes	<p>Tipe double adalah subtype data yang menampung nilai float. Tipe ini memiliki memiliki 2 jenis jangkauan nilai float yaitu :</p> <ul style="list-style-type: none"> • -1,79769313486231570E+308 sampai -4,94065645841246544E-324 untuk nilai negatif. • 4,94065645841246544E-324 sampai 1,79769313486231570E+308 untuk nilai positif.
Integer	4 bytes	Tipe Integer memiliki jangkauan nilai dari -2.147.483.648 sampai 2.147.483.647.
Long	8 bytes	Tipe Long memiliki jangkauan nilai dari -9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.807.
Object	4 bytes	Semua tipe dapat disimpan di variabel Object.
Short	2 bytes	Tipe Short memiliki jangkauan nilai dari -32.768 sampai 32.767.
Single	4 bytes	Tipe Single adalah subtype data yang menampung nilai float. Tipe ini memiliki memiliki 2 jenis jangkauan nilai float yaitu :

		<ul style="list-style-type: none"> • -3,4028235E+38 sampai -1,401298E-45 untuk nilai negatif. • 1,401298E-45 sampai 3,4028235E+38 untuk nilai positif.
String	Tergantung dengan pemakaian	Tipe String dapat menampung karakter Unicode 0 sampai 2 milyar.
User-Defined Type	Tergantung dengan pembuatan	Tipe data yang dapat didefinisikan sendiri.

2.4.2 Deklarasi Variabel

Variabel adalah suatu tempat untuk menyimpan suatu nilai. Untuk dapat menggunakan variabel, perlu melakukan deklarasi terlebih dahulu. Deklarasi ini bertujuan untuk menentukan tipe data yang akan digunakan pada variabel.

Deklarasi untuk variabel pada Visual Basic .Net biasa dilakukan dengan menggunakan atribut Dim atau Public kemudian diikuti dengan nama variabel serta tipe data dari variabel tersebut. Berikut ini contoh pendeklarasian variabel :

```
Dim x As Integer
Dim s As String
Dim s1, s2 As String
```

2.4.3 Konstanta

Konstanta adalah suatu nilai yang sifatnya tetap (tidak berubah). Dan untuk dapat menggunakannya, konstanta perlu didefinisikan dan diberikan nilai terlebih dahulu. Konstanta terdiri dari 2 macam yaitu instrinsik (konstanta dari Visual Basic) dan user-defined (konstanta yang dibuat sendiri).

Pembuatan konstanta dapat dilakukan dengan mendeklarasikan dengan keyword Const, di bawah ini adalah contoh user-defined konstanta.

```
`Menyimpan nilai matematika
Const conPi As decimal = 3.14159265358979
Const myCon As integer = 332
`Meyimpan nilai tanggal
Const conDate As Date = #1/1/1995#
```

Untuk konstanta yang nilai matematika, dapat langsung memasukkan nilai secara langsung seperti contoh di atas. Sedangkan untuk nilai bertipe tanggal, maka digunakan tanda # pada awal dan akhir tanggal.

Konstanta juga dapat menyimpan nilai string, berikut contoh konstanta yang menyimpan nilai string :

```
Const conVersion As string = "07.10.A"
Const MyCodeName As string = "Hellooo.."
```

2.4.4 Array

Pendeklarasian array sama dengan mendeklarasikan variabel yang lain, yaitu dengan menggunakan keyword Dim dan diikuti dengan nama array serta tipe datanya. Array dapat dideklarasikan dengan satu atau lebih dimensi (multidimensi) sesuai dengan kebutuhan aplikasi. Di bawah ini adalah deklarasi array satu dimensi :

```
Dim a(2) As String
a(0) = "1"
a(1) = "2"
a(2) = "3"
```

Array a dideklarasikan dengan tipe data string yang memiliki 3 elemen. Elemen array a dimulai dengan index 0 sampai index 2 sehingga total ada 3 elemen.

Berikut adalah deklarasi array multidimensi :

```
Dim b(2,2) As String
b(0,0) = "1"
b(1,0) = "2"
b(2,0) = "3"
```

Array b di atas adalah array dua dimensi yang memiliki 3x3 elemen.

2.4.5 Operator

Operator pada Visual Basic terdapat 4 macam yaitu Operator Aritmatika, Operator Penggabung String (Concatenation), Operator Pembandingan (comparison) dan Operator Logika.

2.4.5.1 Operator Arimatika

Operator Arimatika digunakan untuk melakukan perhitungan aritmatika.

Operator ini terdiri dari :

- Operator pangkat \wedge
- Operator perkalian $*$
- Operator pembagian $/$
- Operator pembagian interger \backslash
- Operator sisa pembagian Mod
- Operator penjumlahan $+$
- Operator pengurangan $-$

2.4.5.2 Operator Penggabung String

Untuk Operator Penggabung String atau concatenation terdiri dua macam operator, yaitu :

- Operator $\&$
- Operator $+$

Untuk menggabungkan dua buah string atau lebih, dapat menggunakan operator di atas. Namun secara umum banyak digunakan operator $\&$ sebagai penggabung antar string.

2.4.5.3 Operator Pembandingan

Operator Pembandingan digunakan untuk membandingkan dua ekspresi, variabel atau objek. Operator ini meliputi :

- Operator kurang dari $<$
- Operator kurang dari atau sama $<=$
- Operator lebih besar dari $>$
- Operator lebih besar dari atau sama dengan $>=$
- Operator sama dengan $=$
- Operator tidak sama dengan $<>$
- Operator Like
- Operator Is
- Operator TypeOf...Is

Untuk operator perbandingan ada beberapa hal yang harus diperhatikan yaitu:

- Perbandingan untuk ekspresi string. Perbandingan dua buah ekspresi string bersifat case sensitif (membedakan huruf besar dan huruf kecil). Seperti kata “Hello” tidak sama dengan “hello”.
- Perbandingan antara numerik dan string selalu menghasilkan numerik bernilai lebih kecil dari string.
- Perbandingan antara variabel empty dengan variabel bernilai, akan menyebabkan Visual Basic mengganti nilai empty yang dibandingkan sesuai dengan tipe variabel dari perbandingan lainnya. Sebagai contoh nilai 0 untuk numerik dan string kosong “” untuk string.
- Apabila dua perbandingan bernilai empty maka hasil perbandingan adalah sama besar.

2.4.5.4 Operator Logika

Terakhir Operator Logika. Pada Visual Basic dapat menggunakan enam macam operator logika. Di bawah ini adalah tabel dari masing-masing operator beserta ekspresi dan hasil perbandingan kedua ekspresi yaitu:

- Operator And
- Operator Not
- Operator Or
- Operator Xor
- Operator AndAlso
- Operator OrElse

2.4.6 Alur Percabangan

2.4.6.1 If...Then...Else

Keyword If digunakan untuk menentukan aliran program dari dua kondisi logika yang berbeda dari sebuah ekspresi. Sintaksnya adalah :

```
If Pengujian_ekspresi
Then perintah1_bila_ekspresi_benar
Else perintah2_bila_ekspresi_salah
```

Di bawah ini adalah potongan program yang menggunakan ekspresi If :

```
Dim X,Y As integer
```

```

Dim Hasil As string
X = 10
Y = 20
If X < Y then
    Hasil = "Nilai X lebih kecil dari Y"
Else Hasil = "Nilai Y lebih kecil dari X"
End if

```

Jika menuliskan lebih dari satu baris perintah maka harus mempergunakan blok penutup end if. Seperti sintaks di bawah ini :

```

If Ekspresi
Then
    Perintah 1
Perintah 2
[Else
    Perintah 1a
    Perintah 1b]
End if

```

2.4.6.2 Select...Case

Keyword ini juga digunakan untuk menentukan aliran program seperti keyword If. Hanya keyword case memberikan langsung pilihan-pilihan yang salah satunya merupakan pilihan. Di mana pilihan-pilihan tersebut telah didefinisikan terlebih dahulu, serta mempercepat aliran program (langsung kebagian yang sesuai). Sintaksnya adalah :

```

Select Case Ekspresi
    Case nilai1
        Perintah1
    Case nilai2
        Perintah2_1
    Case nilai3
        Perintah3_1
    [Case else
        Perintah]
End select

```

Keyword case else hanya digunakan jika kondisi yang ada tidak sesuai dari nilai ekspresi. Di bawah ini adalah contoh potongan program dengan menggunakan keyword Case :

```

Select Case FirstName
    Case "John"
        Name = "Hai, Aku adalah John"
    Case "Paul"
        Name = "Hai, Aku adalah John"
    Case "Ringo"
        Name = "Hai, Aku adalah John"
    Case Else
        Name = "Hai, Aku adalah " & FirstName
End Select

```

2.4.7. Pengulangan

Dalam membuat suatu program sering kali diperlukan pengulangan perintah. Di mana pengulangan ini akan berhenti sampai pada suatu kondisi tertentu terpenuhi atau tidak terpenuhi. Hal ini bergantung pada keinginan dari pembuat program.

Pada Visual Basic terdapat beberapa macam pengulangan yaitu

2.4.7.1 For Loop

Keyword ini akan mengulang kumpulan perintah sebanyak counter yang dijalankan. Sintaks dari keyword ini adalah :

```
For counter = mulai to selesai [step jangkauan]
    Perintah1
    Perintah2
Next
```

Perintah 1 dan Perintah 2 akan diulangi sebanyak selesai – mulai + 1\ jangkauan kali. Pada pengulangan ini terdapat fasilitas untuk keluar dari pengulangan walaupun kondisi belum selesai, yaitu dengan menggunakan keyword exit for. Contoh penggunaan exit for :

```
For I =1 to 100 (step 2)
    ....
    If I mod 5 = 0 then exit for
    ....
Next
```

Keyword For juga memungkinkan untuk dipakai secara bertingkat atau nested loops. Contoh nested loops :

```
For I = 1 to 10
    For J = 1 to 5
        For k = 1 to 3
            Perintah 1
            Perintah 2
            Perintah 3
        Next
    Next
Next
```

Pengulangan di atas merupakan pengulangan increment atau nilai variabel counter yang digunakan bertambah. Bagaimana untuk melakukan pengulangan dengan pengulangan decrement atau nilai variabelnya berkurang? Pengulangan For pun memiliki cara agar dapat melakukan pengulangan

decrement ini yaitu dengan memberi nilai negatif pada step. Contoh berikut adalah pengulangan decrement :

```
For I : 10 to 1 step -1
Stateamen 1
Stateamen 2
Next
```

2.4.7.2 For Each ...Next

Keyword ini akan mengadakan pengulangan untuk mendaftar semua anggota dari sebuah array atau collection. Sintaks dari pengulangan ini adalah :

```
For each elemen in array_atau_collection
    Perintah 1
    Perintah 2
Next
```

Pada pengulangan ini dapat pula menggunakan keyword exit for untuk keluar dari pengulangan. Di bawah ini adalah contoh potongan program dengan menggunakan pengulangan For each...Next

```
Dim myarray(10) As string
For each elemen in myarray
    Perintah 1
    Perintah 2
Next
```

2.4.7.3 Do ... Loop

Do...Loop digunakan untuk melakukan pengulangan perintah. Model pengulangan Do...Loop ada dua yaitu dengan menggunakan While dan Until.

Dalam melakukan pengulangan dengan kondisi While terdapat dua cara. Cara pertama kondisi While terletak pada bagian awal pengulangan. Sedangkan cara kedua kondisi While terletak pada bagian akhir pengulangan. Pengulangan akan berlangsung sampai kondisi While bernilai true. Sintaksnya adalah :

Pengecekan dilakukan pada bagian awal pengulangan :

```
Do While kondisi
    Perintah 1
    Perintah 2
Loop
```

Pengecekan dilakukan pada bagian akhir pengulangan

```
Do
    Perintah 1
    Perintah 2
Loop While kondisi
```

Seperti pada kondisi While, pengulangan dengan kondisi Until juga terdapat dua cara pengecekan yaitu pada bagian awal dan akhir pengulangan. Pengulangan akan berlangsung sampai kondisi Until bernilai true. Sintaksnya adalah :

Pengecekan dilakukan pada bagian awal pengulangan :

```
Do Until kondisi
    Perintah 1
    Perintah 2
Loop
```

Pengecekan dilakukan pada bagian akhir pengulangan :

```
Do
    Perintah 1
    Perintah 2
Loop Until kondisi
```

2.4.7.4 While ... Wend

Keyword ini akan mengulangi sekumpulan perintah selama kondisi yang diperiksa bernilai TRUE dan akan selesai sampai kondisi menjadi False (terpenuhi). Hal ini juga bisa dilakukan dengan menggunakan keyword Do Loop dan While.

Walaupun keyword While-Wend setara dengan keyword Do-While loop, tetapi penggunaan While wend telah diperkenalkan sejak lama. Sintaks keyword ini adalah :

```
While kondisi
    Perintah 1
    Perintah 2
Wend
```

2.4.8 Prosedur

Visual Basic .Net memiliki prosedur dan fungsi. Perbedaan keduanya adalah terletak pada nilai kembaliannya. Pada prosedur tidak mengembalikan nilai kembalian (non return value). Sedangkan pada function memberikan nilai kembalian (return value).

2.5. Web Forms

Web Forms adalah model pemrograman Common Language Runtime (CLR) yang dapat digunakan oleh server untuk memproses dan menghasilkan web pages yang dinamik.

Web Forms mengizinkan untuk membuat halaman web yang berbasis forms. Dalam membuat halaman tersebut, dapat dipergunakan Server Control untuk membuat elemen UI dan memrogramnya sesuai keinginan. Hal ini jelas akan memudahkan dalam memrogram.

Keuntungan dari Web Forms ini adalah memiliki banyak control UI (User Interface). Dimana control ini memiliki fungsi-fungsi umum dan dapat langsung dipergunakan. Hal ini tentunya dapat mengurangi jumlah kode yang harus dibuat serta memberikan kemudahan untuk membuat program yang bersih / teratur.

Dalam membuat Web Forms terdapat beberapa sintaks yang perlu diperhatikan. Berikut adalah 8 jenis sintaks yang berguna dalam membuat Web Forms :

- Sintaks Kode Render
- Sintaks Pendeklarasian Kode
- Sintaks Server Control
- Sintaks HTML Server Control
- Sintaks Databinding
- Sintaks Tag Objek
- Sintaks Server Side Comment
- Sintaks Server Side Include

2.5.1 Sintaks Kode Render

Sintaks kode render ini sebenarnya sama dengan sintaks ASP yaitu dengan menggunakan elemen `<%...%>`. Di mana untuk memulai pemrograman harus dimulai dengan elemen `<%` dan diakhiri dengan `%>`.

2.5.2. Sintaks Pendeklarasian Kode

Sintaks kedua yang akan dibahas adalah sintaks pendeklarasian kode. Pada ASP.Net perlu untuk mendeklarasikan kumpulan kode. Yaitu pada bagian script, sintaks pendeklarasiannya adalah : `<script runat="server">`

Ini berarti kode dikompilasi dan dijalankan di server.

2.5.3. Sintaks Server Control

Pada ASP.Net terdapat puluhan Server Control yang telah jadi dan siap dipakai. Server Control ini seperti komponen pada bahasa visual yang dapat digunakan, tanpa perlu mengetahui bagaimana cara kerja dari komponen tersebut. Di bawah ini adalah tabel Server Control yang telah build-in pada ASP.Net.

Tabel 2.2. Server Control

AdRotator	Button	Calendar
CheckBox	CheckBoxList	CompareValidator
CustomValidator	DataGrid	DataList
DropDownList	HyperLink	ImageButton
Image	Label	LinkButton
ListBox	Panel	Placeholder
RadioButton	RadioButtonList	RangeValidator
RegularExpression Validator	Repeater	RequiredField Validator
Table	TableCell	TableRow
TextBox	ValidationSummary	XML

Sintaks dari Server Control ini adalah :

```
<asp:nama_ServerControl runat="server" />
```

Untuk dapat menggunakan, berikut contoh sintaks Server Control label:

```
<asp:label id="Message" font-size=24 runat="server" />
```

2.5.4. Sintaks HTML Server Control

Selain Server Control di atas ASP.Net memiliki HTML Server Control. HTML Server Control dipergunakan untuk memanipulasi elemen HTML.

HTML Server Control adalah komponen pada ASP.Net yang berjalan di server. Sedangkan tag HTML biasa tidak berjalan atau dieksekusi oleh server melainkan langsung diberikan ke client. Untuk membedakannya pada HTML Control server mempergunakan atribut :

```
<span runat=server>
```

Di bawah ini adalah HTML Server Control yang telah build-in pada ASP.Net.

Tabel 2.3 HTML Server Control

HtmlAnchor	HtmlButton	HtmlForm
HtmlGenericControl	HtmlImage	HtmlInputButton (Button)
HtmlInputButton (Reset)	HtmlInputButton (Submit)	HtmlInputCheckBox
HtmlInputFile	HtmlInputHidden	HtmlInputImage
HtmlInputRadioButton	HtmlInputText (Password)	HtmlInputText (Text)
HtmlSelect	HtmlTable	HtmlTableCell
HtmlTableRow	HtmlTextArea	

2.5.5. Sintaks Databinding

Pengaksesan data (data binding) merupakan hal penting yang perlu dilakukan dalam membuat suatu aplikasi web. ASP.Net menyediakan cara untuk mengakses data, memfilter data, dan memanipulasi data. Untuk mengakses data pada sintaks ini mempergunakan tanda <%# pada awal kode dan tanda %> pada akhir kode.

Berikut adalah contoh pengaksesan data dengan menggunakan datalist Server Control.

```
<asp:datalist runat=server>
```

Bila menggunakan datalist, maka harus menspesifikasikan tiap-tiap ItemTemplate. Pengaksesan data terjadi pada ItemTemplate tersebut. Contoh :

```
<asp:datalist id="MyList" runat=server>
  <ItemTemplate>
    Here is a value: <%# Container.DataItem %>
  </ItemTemplate>
</asp:datalist>
```

Container adalah lokasi penyimpanan data dari datalist.

2.5.6. Sintaks Tag Objek

Untuk dapat menggunakan array dapat dilakukan dengan menggunakan ArrayList dan dideklarasikan seperti berikut ini :

```
Dim myarray As New ArrayList
```

Pendeklarasian array dapat dilakukan dengan model lain yaitu dengan menggunakan sintaks Tag Objek. Tag Objek dapat berfungsi untuk mendeklarasikan dan membuat instant / turunan dari variabel.

Cara pendeklarasian array dengan sintaks Tag Objek :

```
<object id="myarray" class="System.Collections. ArrayList"
  runat="server"/>
```

Maka tidak perlu menulis “Dim myarray As New ArrayList” lagi. Tetapi cukup digantikan dengan Tag Objek di atas. Objek myarray akan dibuat secara otomatis pada waktu program dijalankan. Setelah itu myarray siap dipergunakan.

2.5.7. Sintaks Server Side Comment

Berikut adalah sintaks untuk dapat memberikan catatan / komentar didalam program. Di mana catatan / komentar tersebut tidak akan dieksekusi oleh program.

Untuk dapat menuliskan catatan / komentar, dapat menggunakan tanda <%-- dan --%> pada awal dan akhir catatan atau komentar. Block bagian dengan tanda <%-- sampai dengan --%> tidak dijalankan oleh server.

2.5.8. Sintaks Server Side Include

Untuk menggabungkan dua atau lebih file pada satu halaman ASP.Net dapat dilakukan dengan menggunakan metode Server-Side Include. Sintaksnya adalah :

```
<!-- #Include File="NamaFile" -->
```

NamaFile : diisi dengan nama file yang hendak digabungkan seperti misal :

```
<!-- #Include File="Header.inc" -->
```

```
...
```

```
<!-- #Include File="Footer.inc" -->
```

Pada bagian berikut ini akan dibahas mengenai perbedaan sintaks kode render di ASP dengan sintaks di ASP.Net.

2.6. Akses Database

Mengakses database adalah salah satu bagian yang sangat penting di setiap aplikasi ASP.Net. Dan yang paling menyenangkan adalah .Net Framework memiliki sangat banyak class dan control yang berkerja dengan database di ASP.Net.

Teknologi untuk mengakses data dapat mempergunakan ADO.Net yang terdapat pada .Net Framework.

2.6.1 ADO.Net

ADO.Net adalah model *disconnected database*. Artinya sekali mengirim data dari database, maka hubungan koneksinya akan putus. Hal ini berbeda dengan model ADO terdahulu yang lebih mempertahankan koneksi tetap terhubung atau disebut juga dengan *open connection database*.

Sebenarnya untuk mengakses data dari database, ADO.Net memiliki 2 macam model yaitu Direct Access (langsung akses Database) dan model DataSet.

Model Direct Access adalah model untuk dapat mengakses ke dalam database secara langsung. Model ini menggunakan Objek Data Command yang didalamnya terdapat perintah SQL atau *reference stored procedure*. Berikut ini adalah beberapa kelebihan menggunakan Direct Access :

- Memiliki control fungsi yang lebih banyak dari DataSet. Dengan menggunakan objek data Command maka Anda dapat melakukan kontrol

lebih pada perintah SQL atau stored procedure dalam mengeksekusi dan menyimpan data.

- Lebih meringankan kerja komputer. Karena pada Direct Access dalam membaca dan menulis dilakukan secara langsung ke database tanpa perlu menyimpan data sementara ke dalam DataSet. Dimana DataSet mempergunakan memori untuk menyimpan data. Sehingga pemakaian Direct Access lebih mengurangi pemakaian memori dan mengurangi beban kerja dibandingkan menggunakan DataSet.

Model akses database yang kedua adalah dengan menggunakan DataSet. Data akan disimpan sementara ke dalam DataSet dan untuk dapat melakukannya diperlukan Data Adapter untuk membaca dan menulis data ke database. Berikut ini adalah keuntungan bila mempergunakan DataSet :

- Berkerja dengan banyak Tabel. Apabila Anda mempergunakan banyak tabel, maka mempergunakan DataSet dapat meningkatkan efisiensi dari program aplikasi Anda.
- Berkerja dengan banyak data dari banyak sumber seperti banyak database, file XML, spreadsheet. Dengan menyimpan menjadi satu pada DataSet, hal ini memudahkan dan mempercepat perkerjaan Anda.
- Memindahkan data diantara tier pada suatu aplikasi yang terdistribusi. Semisal database Anda terletak pada komputer lain (data tier). Untuk memudahkan, data yang diperlukan oleh aplikasi akan disimpan ke dalam DataSet. Sehingga hal ini dapat mempercepat pengaksesan dan manipulasi data.
- Apabila aplikasi Anda mempergunakan pertukaran data (data exchange) dengan aplikasi lain. DataSet sangat efektif dalam melakukan pertukaran antar data ini.
- Apabila suatu record data diakses berulang-ulang pada halaman web yang berbeda maka sangat efisien bila disimpan ke dalam DataSet. Karena record data tersebut berada pada memori sehingga pengaksesannya sangat cepat dibanding query kembali ke dalam database. Selain itu DataSet menyediakan fasilitas untuk mem-filter, ataupun mengurutkan record di dalamnya.

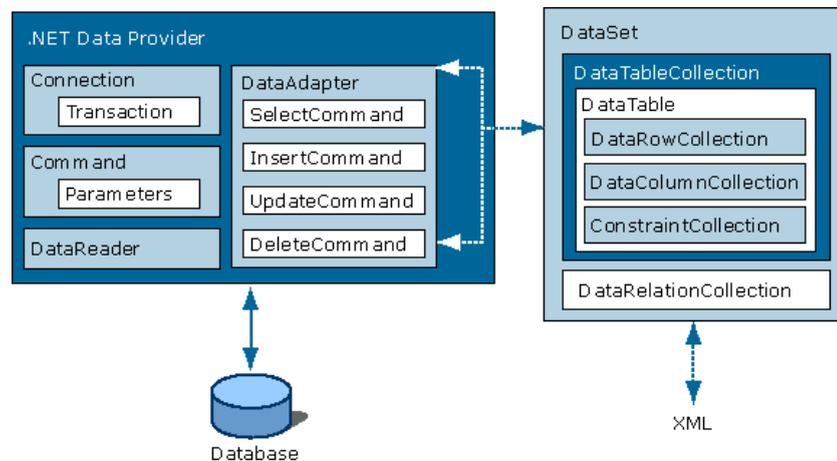
Pada ADO.Net sama dengan ADO memiliki objek dasar. Pada ADO memiliki dua macam objek yaitu Connection dan Command, sedangkan pada

ADO.Net selain memiliki kedua objek di atas terdapat objek tambahan yaitu DataSet, DataReader dan DataAdapter.

Berikut adalah penjelasan singkat dari masing-masing objek ADO.Net :

- Connection : digunakan untuk menghubungkan dan mengatur transaksi terhadap database.
- Command : digunakan untuk melakukan akses langsung ke dalam database melalui perintah SQL.
- DataReader : digunakan untuk membaca record data dari database.
- DataSet : digunakan untuk menyimpan, memindahkan data dari banyak tabel dan banyak sumber ke dalam memori.
- DataAdapter : digunakan untuk melakukan query data dari database serta menulis perubahan yang terjadi di DataSet ke dalam database.

Di bawah ini adalah ilustrasi untuk memberikan gambaran bagaimana hubungan antar kelima objek diatas :



Gambar 2.2 Hubungan Objek di ADO.Net

Pada kelima objek di atas dapat dibedakan menjadi 2 jenis yaitu objek untuk mengakses database SQL Server atau SQL Server.Net Data Provider dan objek untuk mengakses OleDb layer atau OleDb.Net Data Provider.

SQL Server.Net Data Provider digunakan sebagai perantara agar dapat mengakses SQL Server 7.0 atau SQL Server versi terbaru. Sedangkan OleDb.Net Data Provider digunakan sebagai perantara agar dapat mengakses macam-macam database seperti Microsoft Access, Oracle, DB2, Adaptive Server Anywhere, dsb.

2.6.2 Membuat Koneksi ke Database

Langkah awal untuk dapat melakukan aktivitas yang berhubungan dengan database adalah dengan membuat koneksi ke database yang akan dipergunakan.

Untuk dapat membuat koneksi ke dalam database perlu menambahkan namespace pada awal bagian program. Tujuan dari namespace ini untuk menciptakan instan / turunan dari class .Net Framework. Sehingga objek ADO.Net dapat dipergunakan.

Jika hendak menggunakan database SQL Server 7.0 atau terbaru, perlu menambahkan namespace seperti di bawah ini pada awal program :

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>
```

Sedangkan jika tidak menggunakan database SQL Server 7.0 atau terbaru melainkan menggunakan database lain seperti Microsoft Access, Oracle, atau Microsoft SQL Server 6.5 maka perlu ditambahkan namespace berikut ini :

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>
```

Untuk dapat mengakses data ke database dibutuhkan objek Connection. Pada ADO.Net dapat membuat dan mengatur koneksi dengan terdapat dua macam cara yaitu :

- **SqlConnection** : Objek Connection yang mengatur koneksi yang menghubungkan ke database SQL Server versi 7.0 atau terbaru.
- **OleDbConnection** : Objek Connection yang mengatur koneksi ke berbagai database via OLE DB. Seperti database Access, DB2, Oracle, dsb.

Karena database yang akan dipergunakan adalah Microsoft Access maka akan lebih banyak membahas tentang OleDb.Net Data Provider. Yaitu OleDbConnection yang digunakan untuk menghubungkan ke database Microsoft Access.

Berikut ini adalah *connection string* untuk menghubungkan ke database Microsoft Access :

```
Provider=Microsoft.Jet.OLEDB.4.0; Data Source=
c:\bin\LocalAccess40.mdb;
```

- Provider : digunakan untuk mendefinisikan nama dari engine database yang digunakan.
- Data Source : digunakan sebagai petunjuk tempat database berada.
- User Id dan Password : apabila mempergunakan login maka perlu disebutkan User id serta passwordnya.

Berikut contoh membuat koneksi ke database Microsoft Access:

```
<%@ Import Namespace="System.Data" %>
<%@ Import NameSpace="System.Data.OleDb" %>
<%
Dim myConnection As OleDbConnection
myConnection = New OleDbConnection( "PROVIDER=
Microsoft.Jet.OLEDB.4.0; DATA Source=c:\authors.mdb")
myConnection.Open()
%>
Connection Opened!
```

Setelah menuliskan namespace pada awal program, berikut adalah mendeklarasikan myConnection sebagai turunan objek OleDbConnection. Kemudian untuk mengaktifkan objek agar merujuk pada database, diperlukan keyword new OleDbConnection(“ “). Dan yang terakhir adalah menambahkan perintah myConnection. Open(). Barulah koneksi telah siap dipergunakan.

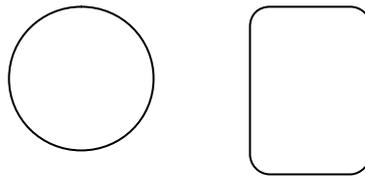
2.7. Data Flow Diagram

Data Flow Diagram yang selanjutnya di singkat dengan DFD adalah representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses-proses suatu sistem.

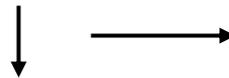
DFD menunjukkan perpindahan dan perubahan data dalam suatu sistem. Meskipun diberi nama DATA FD, namun penekanan pada DFD lebih pada prosesnya, bahkan DFD merupakan salah satu alat pemodelan proses dari sistem yang paling sering digunakan.

Simbol-simbol yang digunakan dalam DFD hanya terdiri dari 4 macam yaitu proses, arus data, simpanan data, dan kesatuan luar (external entity).

- Proses** adalah simbol yang mengubah suatu data dari suatu bentuk menjadi bentuk yang lain. Atau dengan kata lain, proses menerima input data dan mengeluarkan output data lain yang telah diproses.
- Simbol dari proses dalam DFD ada 2 macam:



- c. **Arus data** atau data flow adalah aliran yang menunjukkan perpindahan data dari satu bagian ke bagian yang lain dalam sebuah sistem. Data flow dalam DFD disimbolkan dengan tanda panah dan diberi nama atau keterangan di sampingnya yang menunjukkan data apa yang mengalir. Contoh:



- d. **Simpanan data** adalah tempat penyimpanan data dalam suatu sistem, baik secara manual maupun secara elektronik. Simpanan data digunakan jika suatu proses perlu menggunakan data tersebut lagi kemudian. Simbol dari simpanan data dalam DFD juga ada 2 macam :



- e. **Kesatuan Luar** (*external entity*) adalah seseorang, sekelompok orang, sebuah departemen di dalam maupun di luar organisasi, atau sebuah sistem yang lain yang memberikan input untuk sistem yang ada atau menerima output dari sistem yang ada. *External entity* juga disebut **terminator**, karena merupakan batas dari sebuah sistem. Dalam DFD kesatuan luar disimbolkan dengan sebuah kotak persegi panjang seperti berikut:

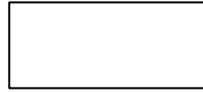


2.8. Entity Relationship Diagram

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Contohnya adalah siswa, buku, pembayaran, nilai, dsb. Entity memiliki atribut dimana atribut dapat menjelaskan karakteristik dari entity. Hubungan antara entity satu dengan entity yang lain disebut sebagai Relationship Entity.

Sedangkan gambaran hubungan antar entity disebut juga sebagai Entity Relationship Diagram, yang selanjutnya dapat disingkat dengan ERD.

Entity dapat digambarkan dengan :



Sedangkan relasi antar entity satu dengan yang lain digambarkan dengan tanda garis yang diikuti dengan keterangan akan hubungan antar entity yang diletakan didalam gambar tanda segitiga. Contoh:



Simpanan data adalah tempat penyimpanan data dalam suatu sistem, baik secara manual maupun secara elektronik. Simpanan data digunakan jika suatu proses perlu menggunakan data tersebut lagi kemudian. Simbol dari simpanan data dalam ERD juga ada 2 macam :

