BAB II

LANDASAN TEORI



2.1. Definisi ASP (*Active Server Pages*)

Microsoft Active Server Page (ASP) merupakan suatu skrip yang bersifat *server-side* yang ditambahkan pada HTML untuk membuat sebuah *web* menjadi lebih menarik, dinamis, dan interaktif. Dengan ASP seorang *developer web* dapat membuat informasi berupa *database* dengan suatu *form* dan menampilkannya dalam suatu tabel yang interaktif atau bahkan membuat aplikasi-aplikasi tertentu dalam sebuah *web*.

ASP bersifat *server-side*, yang berarti adalah bahwa proses pengerjaan script berlangsung diserver, bukan di *browser klien*. Dengan kata lain jika seorang *klien* (*user*) menggunakan sebuah browser untuk memangil sebuah *file* ASP, maka *browser* tersebut mengirimkan permintaan ke *web server*, kemudian *server* tersebut mengeksekusi setiap script yang ada dan hasilnya dikirimkan kembali ke *browser klien*. Karena bersifat *server-side* maka untuk dapat dijalankan pada sebuah PC bias yang berbasis windows, PC tersebut perlu disimulasikan menjadi sebuah *web server* dengan menginstal *Microsoft Personal Web Server (PWS)* atau *Microsoft Information Services (IIS)*.

Bahasa skrip standar yang digunakan oleh ASP adalah *Microsoft Vbscript* dan *Microsoft Jscript*.

2.2. Struktur skrip ASP

Sebuah *file* ASP merupakan *file text only* yang didalamnya berisi teks, HTML tag dan skrip ASP. Jadi dapat dikatakan bahwa *file* ASP sebenarnya merupakan fie HTML biasa yang didalamnya ditambahkan skrip ASP. Jika *file* HTML mempunyai ekstensi **.html* atau **.htm*, maka jika sudah diberi skrip ASP, ekstensi tersebut tinggal diubah menjadi **.asp*.

Skrip ASP bisa diletakkan dimana saja seperti halnya dalam HTML tag. Untuk membedakan antara text, HTML tag, dan skrip ASP, maka digunakan suatu tanda yang disebut delimeter. Delimeter adalah suatu karakter atau kumpulan karakter yang mengawali dan mengakhiri suatu tag atau script. Untuk HTML tag delimeter yang digunakan adalah < dan >, sedangkan untuk script ASP *delimeter* yang digunakan adalah kumpulan karakter < % dan % >.

Contoh HTML sederhana yang mengandung skrip ASP seperti dibawah ini :

```
<HTML>
<BODY>
Halaman ini dipanggil pada <% =Now %>
</BODY>
</HTML>
```

pernyataan "*Now*" berfungsi untuk menampilkan waktu sekarang. Jadi jika HTML diatas di eksekusi di *browser*, maka akan muncul hasil kurang lebih seperti ini:

```
Halaman ini dipanggil pada 1/8/04 2:20:00 PM
```

Waktu yang tertera pada hasil diatas tentu tergantung waktu pada saat skrip ASP ini dieksekusi.

Skrip ASP bahkan dapat dikombinasikan dengan HTML tag seperti ditunjukkan oleh contoh berikut :

```
<HTML>
<BODY>
<% FOR I = 1 to 5 %>
<FONT SIZE = <% = I %>>
Selamat Pagi <BR>
</FONT>
<% NEXT %>
</BODY>
</HTML>
```

2.3. Bahasa Skrip ASP

Diatas telah dikatakan bahwa skrip ASP dapat mengunakan *Microsoft VBScript* dan *Microsof Jscript*. Jika tidak dinyatakan tersendiri, maka secara default skrip yang digunakan adalah Microsoft VBScript. Untuk mengubah default itu dilakukan dengan menambahkan perintah sebagai berikut :

```
<%@ LANGUAGE = Bahasaskrip %>
```

dimana *BahasaSkrip* adalah bahasa skrip yang dipilih sebagai bahasa primer. *BahasaSkrip* dapat diisi dengan VBScript ataupun Jscript. Bahkan jika *web server* mendukung perl, ASPpun dapat menggunakan skrip tersebut. Perintah diatas disebut Asp *directives* dan harus diletakkan pada baris pertama pada sebuah file ASP.

Selain bahasa primer, ASP juga mengizinkan untuk menggunakan bahasa lain didalam suatu file yang sama. Jika ingin menggunakan bahasa pemograman yang lain selain bahasa primer, maka perlu ditambahkan bahasa skrip seperti berikut :

```
<SCRIPT [RUNAT = SERVER] LANGUAGE=BAHASASKRIP>
...script...
</SCRIPT>
```

Pernyataan RUNAT = SERVER menyatakan bahwa skrip tersebut harus diproses di server. Dapat dilihat diatas bahwa *delimeter* untuk bahasa skrip bukan primer sama seperti HTML tag, dan bukan *delimeter* ASP.

Selain bahasa-bahasa skrip diatas , ASP juga mendukung bahasa SQL (*Structure Query Language*). Bahasa SQL ini digunakan untuk hal-hal yang berkaitan dengan pengolahan *database*.

2.4. VBScript

VBScript merupakan salah satu anggota keluarga dari bahasa pemograman Visual Basic. Perbedaan mendasar antara Visual Basic dengan VBScript adalah bahwa Visual Basic memiliki lingkungan desain kerja (Design-time environmen), sedangkan VBScript tidak. Dalam Visual Basic seorang programmer dapat membuat *form* dan menulis kode program dengan menggunakan shell yang interaktif. Sedang *VBScript* menumpang didalam HTML ataupun didalam lingkungan pemrograman lain, misalnya untuk pembuatan makro pada *Microsoft word* dan *Excel*.

2.4.1 Tipe Data

Tipe data yang dikenal oleh VBScript hanya ada satu yaitu *variant*. *Variant* ini dapat dipergunakan untuk menampung berbagai jenis data, seperti bilangan, karakter, string dan lain-lain. Sekalipun demikian, *Variant* memiliki beberapa subtipe yang dikategorikan berdasarkan jenis data yang dipergunakan. Subtipe *variant* ditunjukkan pada table berikut ini :

SUBTIPE	DESKRIPSI
Empti	*Variant* tidak diberi nilai awal. Ini berarti nilai 0 untuk bilangan atau *string* kosong (“”).
Null	*Variant* tidak berisi data apapun (termasuk bilangan 0 atau *string* kosong).
Boolean	Berisi nilai *true* atau *false*.
Byte	Merupakan bilangan integer dengan rentang 0 hingga 225.
Integer	Merupakan bilangan integer dengan rentang –32.768 hingga 32.768.
Currency	Merupakan bilangan riil dengan rentang –922.337.203.685.477,5808 hingga 922.337.203.685.477,5808.
Long	Merupakan bilangan integer dengan rentang –2.147.483.648 hingga 2.147.483.648
Single	Merupakan bilangan riil presisi tunggal dengan rentang –3.402823E38 hingga –1.401298E-45 untuk bilangan negatif, dan 1.401298E-45 hingga 3.402823E38 untuk bilangan positif.
Double	Merupakan bilangan riil presisi ganda dengan rentang –1.79769313486232E308 hingga –4.94065645841247E-324 untuk bilangan negatif dan 4.94065645841247E-324 hingga 1.79769313486232E308 untuk bilangan positif
String	Merupakan data berupa *String* (teks) yang dapat mencapai 2 juta karakter.
Object	Berisi sebuah objek
Error	Berisi sebuah nomor keterangan *error*

Tabel 2.1 Subtipe Variant

2.4.2 Deklarasi Variabel

VBScript memiliki beberapa pernyataan yang digunakan untuk pendeklarasian variable, yaitu Dim, Private, dan Public, contoh :

```
Dim Var1, Var2
Public nama, alamat, telpon
Private Suhu
```

Perbedaan pernyataan ini terletak pada ruang lingkup variable yang digunakan. Jika menggunakan Dim atau Private, maka variabelnya bersifat local, sedangkan jika menggunakan publik, maka variabelnya bersifat global.

VBScript memiliki kemampuan untuk mengenal variable sekalipun tidak dideklarasikan, namuan sebaiknya deklarasi tetap dilakukan untuk menghindari kesalahan menggunakan variable. Untuk "memaksa" agar deklarasi harus dilakukan maka pada bagian atas script harus ditambahkan pernyataan *Option Explicit.*

2.4.3 Aturan Penamaan Variabel

Variable adalah suatu tempat dimemori yang dialokasikan sebagai wadah untuk menyimpan informasi yang dibutuhkan program dan nilainya dapat berubah-ubah sepanjang jalannya program. Ada beberapa aturan yang harus dipenuhi dalam menamai sebuah variable, yaitu :

1. Harus dimulai dengan huruf
2. Tidak boleh ada nama variable yang sama pada ruang lingkup yang sama
3. Dapat mencapai 255 karakter
4. Tidak boleh menggunakan karakter-karakter tertentu, seperti karakter yang digunakan untuk operasi matematika (misalnya +, -, /, *, <, >), atau kata-

kata kunci yang telah dipergunakan untuk menulis kode program, misalnya if, then, else, exit, dan lain-lain.

5. Tidak boleh menggunakan spasi. Untuk menggantikan spasi,digunakan karakter underscore (_).

Penggunaan variable dalam VBScript tidak bersifat *case sensitive* artinya huruf kapital dan huruf kecil tidak dibedakan.

2.4.4 Variabel Array

Array merupakan sekumpulan nilai data yang dikelompokkan dalam sebuah variable. Array berguna jika terdapat beberapa kumpulan nilai data yang sejenis dan akan mendapat perlakuan yang sama pula, misalnya saja data sepuluh nilai data yang sejenis dan akan diolah dengan cara yang sama, maka akan lebih mudah untuk memasukkannya dalam sebuah array darai pada masukannya kedalam sepuluh variable yang berbeda. Kumpulan nilai tersebut satu sama lainnya dibedakan dengan indeks dan masing-masing disebut elemen array. Untuk mendeklarasikan sebuah array digunakan pernyataan seperti berikut ini :

Dim nana array (elemen-1)

Contoh :

```
Dim A(10), NamaSiswa (25)
```

Jumlah elemen array pada contoh pertama adalah 11, sedangkan jumlah elemen array pada contoh kedua adalah 26. Untuk mendeklarasikan sebuah variable array yang dimulai dari 0.

2.4.5 Operator

Operator merupakan suatu alat yang dipergunakan untuk mengolah dan memanipulasi data. VBScrip mengenal beberapa tipe operator, yaitu operator penugasan, operator aritmatika, operator pembandingan, dan operator logika.

1. Operator Penugasan

   Operator penugasan berfungsi untuk memasukkan data kedalam satu variable. Operator ini dilambangkan dengan tanda sama dengan (=).

2. Operator Aritmetika

   Operator aritmatika digunakan untuk operesi matematis. Jenis operator matematika diperlihatkan pada tabel berikut :

OPERATOR	DESKRIPSI
^	Digunakan untuk pemangkatan
*	Digunakan untuk perkalian
/	Digunakan untuk pembagian
\	Digunakan untuk pembagian, namun hasil yang diperoleh adalah integer saja
Mod	Digunakan untuk pembagian, namun hasil yang diperoleh adalah sisa sisa dari pembagian tersebut
+	Digunakan untuk penambahan
-	Digunakan untuk pengurangan
&	Digunakan untuk penggabungan string

Tabel 2.2 Operator Aritmetika

3. Operator pembandingan

Operator pembandingan adalah operator yang membandingkan antara satu nilai atau operasi matematis dengan nilai atau operasi yang menghasilkan satu nilai yang bersubtipe Boolean, yaitu True atau False.

OPERATOR	DESKRIPSI
<	Lebih Kecil
>	Lebih Besar
<=	Lebih Kecil atau Sama Dengan
>=	Lebih Besar atau Sama dengan
=	Sama Dengan
<>	Tidak sama dengan

Tabel 2.3 Operator Pembandingan

4. Operator Logika

Operator logika digunakan untuk membandingkan dua ekspresi perbandingan dan menghasilkan nilai yang bersifat Boolean. Jenis operator logika ini ditunjukkan oleh tabel dibawah ini.

OPERATOR	DESKRIPSI
And	Dan
Or	Atau
Xor	Atau yang ekslusif
Not	Tidak
Eqv	Sebanding
Imp	Implikasi

Tabel 2.4 Operator Logika

2.5. Dasar Pemograman ASP

2.5.1 Membuat File ASP

File ASP merupakan file teks dengan ekstensi *.asp dan berisi kombinasi dari item-item dibawah ini :

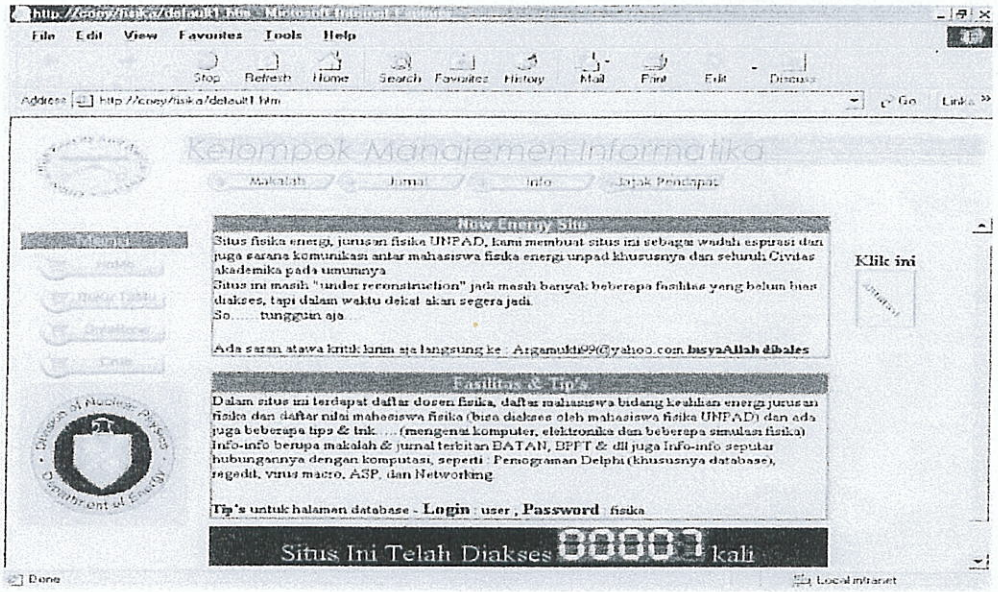
1. Teks
2. HTML tag
3. Perintah – perintah ASP

Karena pada dasarnya file ASP merupakan file HTML yang disisipi skrip ASP, maka pembuatan file ASP dapat dilakukan dengan menggunakan teks editor, misalnya Noteped atau Edit. File ASP juga dapat digunakan dengan menggunakan HTML editor yang memiliki fasilitas *insert script*, misalnya *fontpage* atau *macromedia dreamweaver*.

Untuk dapat mengeksekusi file ASP, maka file tersebut harus diletakkan pada direktori virtual yang telah diberi akses *execute*. File tersebut dieksekusi

menggunakan sebuah browser (missal : Internet Explorer atau Netscape), dan pada bagian address diisikan alamat file ASP tersebut sebagai berikut :

http : //namakomputer/namavirtual/namafile.asp



Gambar 2.1 Contoh pengaksesan file ASP pada IE 5.0

2.5.2 ASP Directive

ASP Directive merupakan pernyataan yang menunjukkan ke server cara untuk menangani sebuah file ASP. Pernyataan ini harus dituliskan pada baris pertama suatu file ASP dan diapit oleh delimeter <% dan %>.

Direktive yang didukung oleh ASP adalah :

1. @CODEPAGE
2. @ENABLESESSIONSTATE
3. @LANGUAGE
4. @LCID
5. @TRANSACTION

2.5.3 Menggunakan Prosedur/Fungsi

Prosedur/fungsi yang berbahasa skrip sama dengan bahasa primer menggunakan delimeter <% dan %>, sedangkan prosedur/fungsi dengan menggunakan bahasa yang lain menggunakan tag <SCRIPT>.

Contoh :

```
<% @Language = VBScript %>
<% Option Explicit %>
<HTML>
<HEAD>
<%
Sub garis ()
      Response.write "<HR>"
End Sub
<Script Language = Jscript Runat = Server>
function Tanggal ()
{
      var x
      x = new data()
      response.write(x.tostring())
}
</SCRIPT>
</HEAD>
<BODY>
<H2>hallo, selamat datang di homepage saya </H2><BR>
<% Call garis() %>
Waktu Sekarang Adalah :
<% Call Tanggal %>
</BODY>
</HTML>
```

2.5.4 Komponen Dan Objek ASP

Komponen yang digunakan dalam ASP sebagian besar berupa komponen ActiveX. Untuk dapat digunakan, mula-mula sebuah komputer harus diset sebagai sebuah objek. Untuk mereferensikan sebuah komponen menjadi sebuah objek, pernyataan yang digunakan adalah sebagai berikut :

<% Set namaobjek = Server.CreateObject("PROGID") %>

namaobjek merupakan nama variable yang menyimpan objek tersebut. Sedangkan PROGID adalah identitas diri komponen ActiveX. Identitas dari komponen ActiveX ini tersimpan dalam registry *windows*. Setiap komputer akan mempunyai komponen AktiveX yang berbeda-beda sekalipun menggunakan versi *windows* yang sama. Hal ini tergantung dari perangkat lunak akan menyertakan komponen ActiveX yang dimilikinya pada saat intalasi.

2.5.5 File Global.asa

File Global.asa merupakan file optional yang dapat digunakan untuk mendefinisikan skrip suatu event atau mendefenisikan objek dengan ruang lingkup *session* atau *application*. File harus terletak pada root direktori suatu aplikasi ASP, dan setiap aplikasi hanya dapat mempunyai sebuah file global.asa. sebagai perbandingan, penggunaan file ini mirip dengan konfig.sys pada DOS atau win pada *windows*.

2.6. Objek Built-In ASP

Aktif Server Pages menyediakan objek-objek built-in yang digunakan untuk mempermudah pemograman. Objek-objek tersebut dapat digunakan untuk anatara lain mengumpulkan data-data permintaan *browser*, menanggapai

permintaan *browser*, menyimpan informasi *user*, dan lain-lain. Objek built-in yang dimiliki oleh ASP adalah :

1. Objek Application
2. Objek Request
3. Objek Response
4. Objek Server
5. Objek Session
6. Objek ObjectContext

Secara umum penggunaan objek-objek tersebut adalah sebagai berikut :

```
Object.[koleksi|metode|property|event] [("variable")]
```

Koleksi adalah tempat meyimpan string, bilangan, objek, ataupun nilai-nilai lain yang dimiliki oleh objek.

Metode adalah tugas-tugas tertentu yang dapat dikerjakan oleh objek yang bersangkutan.

Properti adalah sifat dari objek yang bersangkutan.

Event merupakan suatu sekumpulan skrip akan dijalankan bila pengguna menjalankan pemicu event tersebut.

2.7 Penggunaan SQL (Structured Query Language)

SQL adalah singkatan dari structured Query Language dan sering disebut *sequel* saja. SQL merupakan bahasa standar untuk pengolahan *database*. SQL mulai dikembangkan pada akhir tahun 70-an di laboratorium IBM San Josse – California. Beberapa sintaks penggunaan SQL yang paling sering digunakan diantaranya adalah :

a. Select

Perintah select digunakan untuk mengambil data dari suatu table. Sintaksnya adalah sebagai berikut :

SELECT {*|namafiled} FROM namatabel [INTO tabeltujuan] [WHERE kondisi]

Tanda bintang (*) menunjukan bahwa semua filed yang ada akan dipilih.

b. Insert

Perintah insert digunakan untuk menyisipkan data kedalam table.

Sintaksnya adalah sebagai berikut :

INSERT INTO namatabel[(filed1[,filed2,...])] VALUE (ekspresi1[,ekspresi2,...])

Atau : INSERT INTO namatabel FROM ARRAY namaarray

c. Delete

Perintah delete digunakan untuk menghapus sebuah record dari table.

Sintaksnya adalah sebagai berikut :

DELETE FROM namatabel WHERE kondisi

d. Update

Perintah update digunakan untuk memperbaharui nilai suatu data.

Sintaksnya adalah sebagai berikut :

UPDATE namatabel

SET criteria

WHERE kondisi


UNIVERSITAS KOMPUTER INDONESIA
UNIKOM