

Indo-Tektips.com
Teknikal Tips dan
Online Forumnya
Orang Indonesia.....

Uang belanja bulanan
yang terus membengkak ?

Dapatkan Artikel-artikel menarik di bawah ini :

[_Artikel Java](#) | [Artikel VB](#) | [Artikel ASP](#) | [Artikel JSP](#) | [Artikel PHP](#) | [Artikel Cold Fusion](#) | [Artikel XML](#) | [Artikel Javascript](#) | [Artikel mySQL](#)
| [WAP](#) | [Oracle](#)

Teknikal Forum Diskusi :

[Java](#) | [VB](#) | [ASP](#) | [JSP](#) | [PHP](#) | [Cold Fusion](#) | [XML](#) | [JavaScript](#) | [mySQL](#) | [MS SQL](#) | [Oracle](#) | [Linux](#) | [Windows 2000](#) | [Windows 9x](#) | [FreeBSD](#) | [Unix](#)
| [RUP](#) | [UML](#) | [XP](#)

Login ID : Pwd : [Jadi Anggota Baru ?](#)

Ingin belajar teknologi
terbaru di Internet ?

[Belajar Java](#)
[Belajar VB](#)
[Belajar ASP](#)
[Belajar JSP](#)
[Belajar PHP](#)
[Belajar Cold Fusion](#)
[Belajar XML](#)
[Belajar Javascript](#)
[Belajar mySQL](#)
[Belajar Oracle](#)
[Belajar WAP](#)

Group Belajar Bersama
[Lewat Email](#)
[Lewat Forum Online](#)

Download :
[Freeware Indo-Tektips](#)
[Software-eBook-Source](#)
[Tutorial Indo \(PDF\)](#)

Lain-lain :
[Indo-Tektips.com Lelang](#)
[\(Auction\)](#)
[Indo-Tektips.com](#)
[Classified](#)
[Buku Tamu\(Guestbook\)](#)
[Saran buat Webmaster](#)



Proyek Freelance

Anda sedang
mencari proyek freelance /

Manajemen Sesi di ASP.NET

Kembali ke [Berita Terbaru](#)

Manajemen Sesi di ASP.NET

Dikirim oleh : Budi Kurniawan pada 10/8/2003 8:48:10 PM

Manajemen Sesi ASP.NET

Oleh Budi Kurniawan
(5Badut.com)

ASP.NET menyediakan metode baru utk manajemen sesi (*Session management*). Kali ini Microsoft menjagokan objek

Session

yang aman dipakai dan skalabel (*scalable*).

Kita semua tahu, HTTP adalah protokol yang *stateless*. Artinya, HTTP tidak bisa secara otomatis menentukan apakah orang yang masuk ke situs kita adalah orang yang baru datang pertama kali atau yang telah mengakses halaman-halaman yang lain. Pemrogram biasanya mengakali ini dengan menggunakan cookie, *hidden field* dalam form, atau metode lainnya. Semuanya dilakukan guna bisa mempertahankan informasi yang telah didapat dari kunjungan sebelumnya dari pemakai yang sama. Di ASP bahkan ada manajemen sesi tersendiri dengan menggunakan objek

Session

. Sayangnya objek

Session

ini dianggap tidak laik lantaran boros memori.

Belajar dari pengalaman, Microsoft kemudian memunculkan sistem baru untuk mengatur sesi di ASP.NET. Hebatnya, metode baru ini mampu mengatasi kelemahan di ASP lama dengan cara membiarkan pemrogram menentukan di mana informasi sesi disimpan. Ini dilakukan dengan menentukan salah satu dari tiga mode yang ada: *in-process*, *state server*, dan SQL Server. Tidak hanya itu, Anda juga bisa menghindari pemakaian cookie untuk menyimpan pengenalan atau pengidentifikasi sesi (*session identifier*).

Karena fitur (*feature*) ini termasuk baru dan cara kerjanya berbeda dari manajemen sesi yang lama di ASP, pertama-tama kita akan membahas mode dan menunda bahasan lainnya. Ini dimaksudkan agar pembaca tidak dibuat bingung.

sampingan ?

[Click a category!](#)

Gratis jadi anggota !

[Logo Design](#)

[Web Design](#)

[Complete Website](#)

[Web Programming](#)

[Flash Animation](#)

[Graphics](#)

[Banner Ads](#)

[Brochures](#)

[Illustration](#)

[Business Cards](#)

[Print Ads](#)

[More Categories](#)

[Bidding Info](#)

[Daftar Gratis Sekarang !](#)



Tiga Mode Manajemen Sesi

Ketiga mode manajemen sesi ASP.NET dibahas di bawah ini. Perhatikan bahwa mode state server dan SQL Server keduanya termasuk mode di-luar-proses (*out-of-process*).

Mode Di-Dalam-Proses (*In-Process*)

Mode ini pada prinsipnya sama dengan manajemen sesi lama di ASP. Mode ini boros memori dan tidak bisa diterapkan dalam sebuah *Web farm* tanpa memakai produk lain untuk memastikan bahwa pengunjung akan diarahkan ke server yang mengeluarkan pengenalan sesi.

Lalu, apa gunanya mode ini? Mode ini digunakan untuk mendapatkan waktu tanggap yang paling cepat karena objek Session disimpan di proses yang sama seperti proses pekerja (*worker process*). Gunakan mode ini bila aplikasi Anda hanya menggunakan satu server Web dan jumlah pemakai/pengunjung tidak terlalu banyak.

Mode State Server

Gunakan mode ini bila Anda menggunakan Web farm. Objek

Session

tidak disimpan di server yang memproses request, tapi di sebuah komputer yang bisa diakses oleh semua server di Web farm. Seperti mode di-dalam-proses, semua objek Session disimpan di memori komputer bersama tersebut. Mode ini juga bukan solusi yang skalabel karena penggunaan memori untuk menyimpan objek

Session

Untuk menggunakan mode ini, semua server Web di Web farm harus mengetahui alamat IP dari komputer bersama yang menyimpan objek Session dan komputer bersama tersebut harus menjalankan servis Windows NT yang namanya ASP State. ASP State ini terserta dalam SDK .NET dan akan diinstal secara otomatis waktu Anda menginstal SDK .NET. Tetapi, Anda harus menjalankannya secara manual dari Administrative Tools di Windows 2000 (lihat Gambar 1). Atau, Anda bisa mengetikkan perintah berikut dari prom perintah.

```
net start aspstate
```



Gambar 1: Servis ASPState

Mode SQL Server

Dalam mode ini, informasi sesi disimpan dalam database SQL Server yang bisa diakses semua server Web di Web farm. Semua server Web harus mengetahui alamat SQL Server ini termasuk *user name* dan *password* untuk mengakses databasenya. Bila Anda tidak memakai Web farm, Anda bisa memakai SQL Server di mesin yang sama dengan server Web.

Mode ini merupakan kompromi terbaik untuk skalabilitas. Mode ini tidak secepat dua mode sebelumnya karena informasi sesi disimpan di database, bukan di memori. Keunggulan lainnya: kalau server Web *hang*, informasi sesi tidak ikut lenyap.

Untuk menggunakan mode ini, Anda terlebih dahulu harus membuat database dalam SQL Server menggunakan file script bernama

```
state.sql
```

yang bisa ditemukan di direktori

```
[drive sistem]\winnt\Microsoft.NET\Framework\[version]\
```

```
.
```

File `state.sql` berisi script SQL untuk menciptakan database ASPState. Bila dijalankan, script ini juga akan menciptakan dua tabel dan beberapa prosedur tersimpan (*stored procedure*).

Untuk menjalankan script ini, masuklah ke prom perintah, lalu pindahlah ke direktori tempat file `state.sql`. Setelah itu, panggil `osql.exe`, yang merupakan salah satu perintah baris SQL Server, dengan `state.sql` sebagai parameternya.

```
osql ?S [server name] ?U [user] ?P [password] <state.sql
```

Misalnya, bisa SQL Server Anda bernama `boni` dan user name "`james`" dan password "`007.5`", ketiklah baris berikut.

```
osql ?S boni ?U james ?P 007.5 <state.sql
```

File config.web

Lalu, bagaimana kita menggunakan mode-mode tersebut? Jawabannya: dengan mengubah setting di file `config.web`.

Ada dua macam file `config.web`: file `config.web` mesin dan file `config.web` aplikasi. Keduanya serupa tapi file `config.web` mesin berlaku untuk seluruh aplikasi. Sebaliknya, file `config.web` aplikasi berlaku untuk aplikasi yang bersangkutan dan sifatnya opsional. Jika ada setting yang sama di kedua file, setting di `config.web` aplikasi yang akan dipakai.

Dalam SDK .NET versi Beta 1, file `config.web` mesin terletak di direktori berikut.

```
WinNT\Microsoft.NET\Framework\v1.0.2204
```

Sedangkan file `config.web` aplikasi ada di direktori aplikasi.

Setting di file config.web yang digunakan untuk manajemen sesi adalah session state. Perhatikan bahwa bila Anda mengedit file config.web, apa yang Anda ubah akan segera terpakai tanpa harus me-restart server Web.

Konfigurasi Manajemen Sesi

Dalam file config.web, session state bisa ditemukan dalam node

```
<sessionstate>
```

. Node ini sintaksnya sebagai berikut.

```
<sessionstate
  mode = "inproc" | "sqlserver" | "stateserver"
  cookieless = "true" | "false"
  timeout = <timeout sesi dalam menit >
  sqlconnectionstring = <string koneksi database>
  server = <nama server>
  port = <nomor port>
/>
```

Keenam setting di atas (mode, cookieless, timeout, sqlconnectionstring, server, dan port) digunakan untuk konfigurasi sesi dalam ASP.NET.

mode

Setting ini menentukan mode dan nilainya bisa berupa salah satu dari tiga nilai berikut: "inproc", "stateserver" dan "sqlserver". Gunakan "inproc" untuk mode di-dalam -proses, "stateserver" untuk servis Windows NT ASP State (di-luar -proses), dan "sqlserver" untuk mode SQL Server.

cookieless

Ini menentukan apakah pengenal sesi dikirimkan dari server Web ke browser dan dari browser ke server Web dengan menggunakan cookie. Nilainya kalau tidak "false" adalah "true". Jika setting ini diberi nilai "false", pengenal sesi dikirimkan dengan menggunakan cookie; sebaliknya, kalau nilai setting ini "true", ASP.NET akan menggunakan cara lain untuk mengirimkan pengenal sesi. Untuk jelasnya, lihat "Manajemen Sesi Tanpa Cookie".

timeout

Setting ini menentukan lama waktu (dalam menit) sebuah sesi dianggap valid. Untuk tiap request, periode timeout di-set ke waktu saat ini ditambah nilai timeout. Nilai setting ini haruslah bilangan bulat positif.

sqlconnectionstring

Setting ini berisi string koneksi database untuk mode sqlserver. Untuk mode lain, setting ini akan diabaikan.

server

Dalam mode "stateserver", setting ini berisi nama server yang menampung informasi sesi. Setting ini hanya digunakan untuk mode "stateserver". Untuk mode lainnya, setting ini akan diabaikan.

port

Dalam mode "stateserver", setting ini digunakan bersama setting server. Setting ini berisi nomor port yang terkait dengan setting server untuk mode "stateserver". Untuk mode lainnya, setting ini akan diabaikan.

Berikut adalah setting default keadaan sesi dalam file config.web mesin.

```
<sessionstate
  mode="inproc"
  cookieless="false"
  timeout="20"
  sqlconnectionstring="data source=127.0.0.1;user id=sa;password="
  server="127.0.0.1"
  port="42424"
/>
```

Untuk artikel ini, Anda sebaiknya membuat file config.web Anda sendiri di direktori aplikasi.

File config.web untuk tiap mode

Kode artikel ini, pada Listing 4 dan 5, menyimpan suatu nilai ke dalam objek Session dan kemudian menampilkan lagi nilai itu. Untuk bereksperimen dengan ketiga mode, Listing 1,2, dan 3 menampilkan contoh untuk tiap mode.

Listing 1: File config.web untuk mode dalam-proses

```
<configuration>
  <sessionstate
    mode="inproc"
    cookieless="false"
    timeout="20"
    sqlconnectionstring=""
    server=""
    port=""
  />
</configuration>
```

Listing 2: File config.web untuk mode stateserver

```
<configuration>
  <sessionstate
    mode="stateserver"
    cookieless="false"
    timeout="20"
    sqlconnectionstring=""
    server="127.0.0.1"
    port="42424"
  />
</configuration>
```

Listing 3: File config.web untuk mode sqlserver

```
<configuration>
  <sessionstate
    mode="sqlserver"
```

```

        cookieless="false"
        timeout="20"
        sqlconnectionstring="data source=127.0.0.1;user
id=sa;password="
        server=""
        port=""
    />
</configuration>

```

Menggunakan Manajemen Sesi ASP.NET

Cara menggunakan objek Session dalam ASP.NET sangat mirip ASP lama. Anda menciptakan pasangan kunci/nilai untuk tiap data. Jika kunci dalam pasangan tersebut sudah ada, nilai lama akan digantikan oleh nilai yang baru. Jika kunci tersebut belum ada, ASP.NET akan menciptakan pasangan kunci/nilai tersebut.

Listing 4 dan 5 berisi kode yang berfungsi sama tapi dalam bahasa yang berbeda: Visual Basic dan C#. Kode tersebut mendemonstrasikan bagaimana caranya memberikan nilai yang diketikkan pemakai ke elemen TEXT (Lihat Gambar 2) ke objek Session. Sewaktu pemakai mengklik tombol SUBMIT, metode WriteState di sisi server akan dijalankan.

Metode

WriteState

memeriksa objek Session untuk menemukan kunci bernama "MySession". Jika ketemu (yaitu, bila nilai kunci ini bukan null), variabel OldValue akan berisi nilai. Metode ini kemudian akan menuliskan nilai elemen TEXT ke objek Session dengan menggunakan kunci "MySession". Terakhir, metode ini menampilkan nilai lama dan baru.

Ingat bahwa untuk menguji kode di Listing 4 dan 5 dengan menggunakan file config.web di Listing 1, 2 atau 3, browser Web harus menerima cookie.

Listing 4: Menggunakan manajemen sesi ASP.NET dengan Visual Basic

```

<HTML>
<HEAD>
<TITLE>ASP.NET Session Management</TITLE>
<SCRIPT runat=server>
    Sub WriteState(sender As Object, e As EventArgs)

        Dim OldValue As String
        If (Len(Session("MySession"))) = 0) Then
            OldValue = "Old value: (empty)"
        Else
            OldValue = "Old value: " & _
                Session("MySession").ToString()
        End If

        Session("MySession") = text1.Value
        span1.InnerHtml = OldValue & _
            "<BR>Current value:" & _
                Session("MySession").ToString()

    End Sub
</SCRIPT>

```

```
</HEAD>
<BODY>
<FORM RUNAT=server>
  <INPUT ID=text1 TYPE=text RUNAT=server>
  <INPUT TYPE=submit RUNAT=server
    OnServerClick="WriteState"
    Value="Write State">
</FORM>
<FONT FACE=verdana SIZE=5>
<SPAN ID=span1 RUNAT=server/>
</FONT>
</BODY>
</HTML>
```

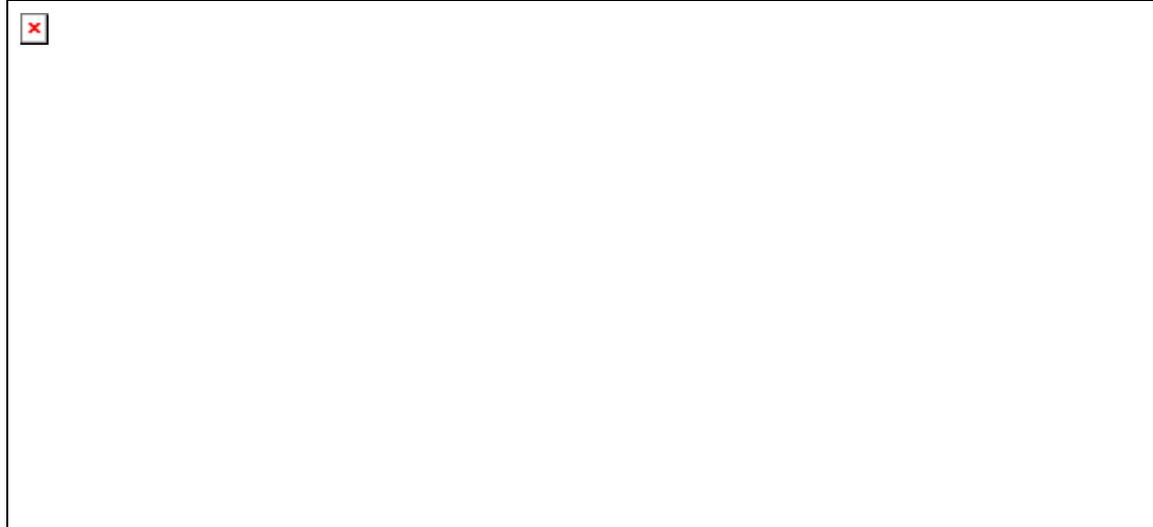
Listing 5: Menggunakan manajemen sesi ASP.NET dengan C#

```
<%@ Page Language="C#" %>
<HTML>
<HEAD>
<TITLE>ASP.NET Session Management</TITLE>
<SCRIPT runat=server>
  public void WriteState(Object sender, EventArgs e) {

    String OldValue = (String) Session["MySession"];
    if (OldValue==null || OldValue == "")
      OldValue = "Old value: (empty)";
    else
      OldValue = "Old value: " +
        Session["MySession"].ToString();
    Session["MySession"] = text1.Value;
    span1.InnerHtml = OldValue +
      "<BR>Current value: " +
      Session["MySession"].ToString();

  }
</SCRIPT>

</HEAD>
<BODY>
<FORM RUNAT=server>
  <INPUT ID=text1 TYPE=text RUNAT=server>
  <INPUT TYPE=submit RUNAT=server
    OnServerClick="WriteState" Value="Write State">
</FORM>
<FONT FACE=verdana SIZE=5>
<SPAN ID=span1 RUNAT=server/>
</FONT>
</BODY>
</HTML>
```



Gambar 2: Menguji manajemen sesi dengan menggunakan cookie

Manajemen Sesi Tanpa Cookie

Tiap sesi aktif dalam ASP.NET diidentifikasi dan dilacak dengan menggunakan string pengenal sesi yang panjangnya 120 bit. Nilai pengenal sesi dihasilkan dengan menggunakan algoritma yang menjamin keunikan (sehingga tidak ada sesi yang berbenturan) dan keacakan pengidentifikasi sesi (sehingga hacker tidak bisa menggunakan sebuah pengidentifikasi sesi untuk menebak pengidentifikasi sesi lainnya).

Bila Anda menggunakan kode di Listing 4 dan 5 dengan file config.web di Listing 1, 2 atau 3, pengenal sesi akan ditransfer dengan menggunakan cookie. Ini karena setting cookieless diset ke "false". Jika Anda tidak mau menggunakan cookie, Anda dapat mengeset setting cookieless ke "true", seperti berikut ini.

```
cookieless="true"
```

Bila Anda tidak menggunakan cookie, sistem manajemen sesi ASP.NET akan menyertakan informasi tentang pengenal sesi dengan memodifikasi URL. Ini ditunjukkan dalam Gambar 3. Perhatikan isi kotak Location browser.

Bila cookieless diset "true", browser tidak perlu diset untuk menerima cookie.



Gambar 3: Menguji manajemen sesi tanpa cookie

Kesimpulan

Dalam artikel ini Anda sudah belajar cara kerja manajemen sesi ASP.NET untuk mempertahankan informasi. Metode dalam ASP.NET berbeda dari ASP. Anda dapat menggunakan salah satu dari tiga mode: di-dalam-proses, state server dan SQL Server.

Anda juga bisa mengeset setting cookieless dalam file config.web ke true bila Anda tidak mau menggunakan cookie dalam manajemen sesi.

[Berikan tanggapan ?](#)