

Dukungan Session ASP dalam Aplikasi Web E-Commerce

Oleh: Andi Sunyoto

Pendahuluan

e-commerce adalah aplikasi yang digunakan untuk melakukan bisnis di web. Perusahaan akan mendapat banyak keuntungan, yang pertama adalah citra perusahaan akan terangkat. Jika perusahaan memiliki citra yang baik maka otomatis akan mudah bagi untuk membangun suatu relasi, peningkatan *Customer Service*, dengan menampilkan informasi barang yang akan di jual, konsumen tidak harus bersusah payah menelpon atau datang langsung ke lokasi. Selain itu juga merupakan ekspansi yang tak terbatas, karena web tidak dibatasi oleh batas-batas geografis dan waktu, maka perusahaan dapat membentangkan sayap untuk menjangkau konsumen seluruh penjuru dunia.

Salah satu tantangan yang dihadapi oleh para perancang aplikasi web adalah bagaimana mengatur informasi user pada suatu rangkaian "perjalanan" ke halaman web tersebut dimana terdapat kemungkinan user akan berpindah-pindah dari suatu halaman ke halaman lainnya dengan nyaman. HTTP yang merupakan protokol yang tidak pernah "mencatat" (stateless protokol) suatu permintaan browser jika browser mengirim permintaan ke server, maka permintaan tersebut akan berdiri sendiri, dan server tidak mempunyai informasi terhadap permintaan-permintaan sebelumnya.

Ketidak mampuan HTTP untuk menyimpan informasi tersebut akan menimbulkan kesulitan dalam perancangan suatu aplikasi web dan itu akan membuat user kurang nyaman dalam menjelajah halaman web anda. Kekurangan tersebut dapat ditangani oleh ASP (*Active Server Pages*) sebagai salah satu program *side-serve* (program yang bekeja di server).

Apa itu Session ASP ?

Session di ASP (*Active Server Pages*) adalah obyek *Built-in* yang ada di ASP yang dapat langsung diakses diberbagai halaman web. Session tersebut dapat menyimpan data yang di miliki user di dalam memori server. Session dapat menyimpan berbagai bentuk tipe data yaitu:

- variabel sederhana, seperti *string*, *numeric*
- variasi array dari satu dimensi maupun banyak dimensi
- obyek-obyek yang dapat dikenal oleh ASP

Dengan kemampuan untuk menerima nilai tersebut membuat ASP sangat berguna. Setiap user di beri identitas sendiri untuk membedakan dengan user lainnya, sehingga tiap user tidak bisa membaca data orang lain.

Samakah Session dengan Cookie ?

Dilihat dari letak penyimpanan data antara Session dan cookie sudah berbeda, jika Session disimpan di server tapi kalau cookie datanya disimpan di komputer klien. Cookie digunakan oleh Session hanya untuk menyimpan identitas Session (*SessionID*), sekali lagi Cookie di klien tidak berisi data, hanya *SessionID*. Tapi Jika anda memerlukan Session tanpa Cookie klien, anda bisa install filter ISAPI yang bernama "Cookie Mungler". Hal ini perlu di perhatikan karena ada sebagian user walaupun sangat jarang *men-disable*-kan (me-non-aktif-kan) cookie pada browser mereka.

Bagaimana Session bekerja ?

Dengan kemampuannya menyimpan informasi data, Session juga dapat digunakan untuk membatasi durasi kunjungan user di suatu halaman web. Ketika masing-masing user datang ke halaman web site, web server mengalokasikan memori untuk membentuk obyek Session user tersebut yang masing-masing user berbeda. Selanjutnya masing user diberi identitas yang disebut dengan *SessionID*. *SessionID* berupa bilangan *numeric* yang selalu bertambah, untuk melihat *SessionID* dari user dapat dilihat dengan perintah:

```
<%  
'melihat SessionID  
Session.SessionID  
  
'menampilkan SessionID ke layar  
Response.Write "SessionID Anda adalah : " & Session.SessionID  
%>
```

Alokasi memori tersebut di batasi dengan periode waktu 10 menit sebagai defaultnya, ini bertujuan untuk meringankan kerja server, karena dengan hilangnya Session pada user, maka server akan membebaskan alokasi memori yang sebelumnya di berikan untuk user yang bersangkutan. Tapi kita dapat di set lebih singkat atau lebih lama dari waktu defaultnya. Session dialokasikan ke dua lokasi yaitu; Web Server dan klien. Di Web server berisi *SessionID* dan data, tapi di klien hanya hanya berisi *SessionID* yang di tuliskan di cookie yang tidak berisi data, jadi di klien data Session tidak dapat di baca. Di dalam program Session dapat diolah sesuka kita. Berikut potongan kode program ASP yang memanfaatkan Session berikut:

```
<%  
'memberi nilai Session nama dengan Andi  
Session("nama")="Andi"
```

```
*memberi nilai Session dengan variabel  
nama="Andi"  
Session("nama")=nama
```

```
*memberi nilai variabel dengan Session  
nama=Session("nama")
```

```
*menuliskan nilai Session ke layar  
Response.write Session("nama")  
%>
```

Keunggulan session dalam menentukan waktu aktif ini juga sangat berguna untuk mengatur berapa user dapat mengunjungi halaman tersebut. Berapa lama user di alaman tersebut dalam di set dengan memasukkan nilai numeris pada kode programnya. Atau bisa dilihat pada kode program berikut:

```
<%  
*Session akan aktif selama 5 menit  
Session.Timeout=5  
%>
```

Atau user dapat menghapus session mereka jika sudah mengunjungi halaman tersebut dengan menekan tombol Log Out kalau kita bayang sedang browsing. Dengan demikian data Session kita jika itu penting tidak akan terbaca oleh user lain. Jika dilihat di dalam kode programnya dengan perintahnya adalah:

```
<%  
*Menghapus Session  
Session.Abandon  
%>
```

Kapan Session di pakai E-Commerce ?

Kalau kita tinjau, kita sebagai pihak penyelenggara ataupun pembuat aplikasi e-commerce kita harus memperhatikan paling tidak 2 bagian yang penting yaitu, bagian klien (pembeli) dan bagian pengelola (*back Office*). Di sisi pembeli terdapat halaman katalog, keranjang belanja dan sisi pengelola di situ terdapat administrasi pengolahan datanya.

Di sisi klien biasanya dibagi atas beberapa halaman yang saling terintegrasi, yaitu:

- a. Halaman selamat datang yang berisi identitas toko *online* yang bersangkutan, serta daftar kategori barang yang di sediakan toko, serta form pencarian barang
- b. Halaman kategori barang yang berisi daftar barang untuk kategori yang dipilih pada halaman selamat datang
- c. Halaman detail yang berisi detail barang yang di pilih pada halaman kategori
- d. Halaman *shopping chart* atau keranjang belanja. Ini merupakan halaman yang penting. Pada halaman ini akan ditunjuk item-item barang yang telah dibeli dan fasilitas untuk menghapus dan mengupdate item-item tersebut.
- e. Halaman *cek out* yang berisi formulir data konsumen tersebut juga data kartu kredit yang digunakan untuk transaksi.
- f. Halaman terima kasih yang berisi konfirmasi bahwa transaksi telah diproses pada halaman ini jуда terdapat proses memasukkan data ke dalam database.

Dengan melihat bagian-bagian tersebut maka besar kemungkinan user untuk berpindah dari halaman satu ke halamn lain. Sebagai contoh pembeli masuk ke halaman kategori setelah itu halaman detail dan memasukkan barang yang dia beli ke keranjang belanja, untuk melihat barang lain dia harus kembali ke halaman kategori dan kembali ke halaman detail barang yang di pilih untuk di masukkan ke keranjang belanja. Agar pembeli nyaman dalam berbelanja data yang lama di keranjang belanja harus masing ada. Dengan kemampuan yang dimiliki HTTP tidak dapat mengenali variabel jika sudah pindah halaman, hal tersebut diatas tidak dapat terselesaikan.

Dengan kemampuan yang dimiliki Session yang dapat di kenali oleh setiap halaman sebelum waktunya habis membantu menyelesaikan masalah diatas. Denga kombinasi Session yang diisi dengan array yang dinamis sangat membantu untuk menampung daftar belanja pada *shopping chart* meskipun kita berpindah-pindah untuk menelusuri semua halaman.

Sebernarnya kita bisa menggunakan koneksi ke database untuk mengisi *shopping chart* kita tapi itu akan memperlambat proses, karena untuk koneksi ke database membutuhkan waktu yang lebih lama.

Di sisi pengelola juga terdapat beberapa halaman. Mulai dari halaman login, entri barang, editing yang kesemuanya harus di cek user yang masuk. Ini berguna untuk membatasi user-user yang masuk untuk hal keamanan. Dengan hanya login sekali pada halaman login yang seterusnya user kita dicatat di dalam Session pengelola tidak harus login pada setiap halaman. Ini sangat membantu dalam pengelolaan sebuah aplikasi. Sehingga efisieansi waktu dan kemudahan dalam suatu aplikasi web e-commerce terwujud.

Kesimpulan :

Session sangat membantu dalam membangun suatu aplikasi web e-commerce, sehingga harapan dalam melakukan browsing kita dapat dengan nyaman, cepat dan aman dapat terwujud khususnya di dalam aplikasi keranjang belanja (shopping chart). Kita tidak harus mengulang memasukkan barang belanjaan kita dari awal jika akan cek out. Tidak hanya itu, session sangat membantu dalam proses (back office) pengelolaan suatu aplikasi web e-commerce jika kita bertindak sebagai pihak penyelenggara situs tersebut.

Referensi :

Alex Homer, Dave Sussman, Brian Francis, **Profesional Active Server Pages 3.0**
Wrox Press Inc, 29 Slasalle St, Suit 520 Chicago, Illinois 60603 USA.

Riyeke Ustadiyanto, SE dan Silvia Ratna Ariani, SE,
Strategi Serangan Internet Marketing, Andi Offset
Jl. Beo 38-40, Yogyakarta.

Yahya Kurniawan, ST, **Aplikasi Web Database dengan ASP** (seri STJ), PT. Elex
Media Komputindo, Jl. Palmerah Selatan 22 Jakarta 10270

Situs:

www.microsoft.com

www.learnasp.com/learn/security.asp

www.msdn.microsoft.com/workshop/server/toolbox/cookie.asp