

BAB II

LANDASAN TEORI

2.1 Model Tingkah Laku Pengunjung

Model tingkah laku pengunjung suatu situs Web mengambil elemen-elemen tingkah laku pengunjung dalam arti pola navigasi, fungsi-fungsi yang digunakan dan waktu antara pengaksesan terhadap fungsi-fungsi yang ditawarkan oleh suatu situs Web. Model tingkah laku pengunjung ini diperoleh berdasarkan interaksi yang dilakukan pada setiap *session* dari pengunjung situs Web. Data mengenai interaksi masing-masing pengunjung pada setiap *session* dapat diperoleh dari *log* akses terhadap situs Web yang tersimpan pada Web server.

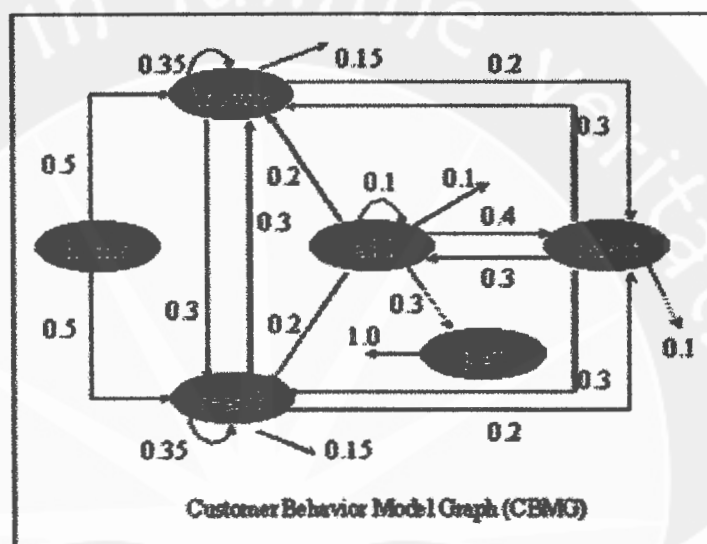
Selama satu *session*, pengunjung dapat diklasifikasikan dalam suatu keadaan yang berbeda menurut tipe fungsi-fungsi yang diminta. Kondisi perbedaan ini disebabkan oleh banyaknya pilihan navigasi yang dimiliki oleh situs web yang memungkinkan kombinasi yang banyak. Transisi yang mungkin terjadi di antara suatu *graph* yang disebut dengan *Customer Behavior Model Graph* (CBMG).

2.1.1. Customer Behavior Model Graph (CBMG)

Customer Behavior Model Graph (CBMG) adalah pemodelan secara *graph* berdasarkan karakteristik *session* suatu Web. CBMG digunakan untuk memodelkan pola navigasi dari user yang mengunjungi suatu situs Web. CBMG didasarkan pada probabilitas transisi secara matrik yang memodelkan bagaimana suatu pengunjung berpindah dari satu *state* ke *state* berikutnya (misalnya

dalam fungsi-fungsi *e-bussiness*).

Untuk membangun suatu CBMG, harus didefinisikan secara jelas fungsi-fungsi yang disediakan situs *Web* untuk pengunjungnya, kemudian kita menyaring fungsi-fungsi tersebut berdasarkan penggunaan *resource*-nya. Terakhir perlu dilakukan penetapan transisi antar *state*.



Gambar 1. Keadaan dan Transisi pada CBMG
(Sumber: <http://eceb.gmu.edu/projects.htm>)

Dalam CBMG tidak diberikan deskripsi yang jelas tentang "waktu berpikir pengunjung" jadi waktu berpikir dianggap terdistribusi secara merata ke tiap perpindahan fungsi dari situs web yang menjadi salah satu kekurangan dari CBMG, Contoh : jika Andi ingin membeli hadiah untuk seseorang, Andi membutuhkan waktu yang lebih lama untuk memilih untuk membeli apa, tetapi setelah pembelian mungkin adalah pembungkusan kado yang seharusnya membutuhkan waktu pikir yang lebih sedikit, oleh sebab itu waktu berpikir aksi ke dua bergantung dari aksi pertama.

Setiap situs web memiliki halaman *home* yang merupakan *state* awal untuk memulai pemodelan CBMG, jadi CBMG mungkin dimulai dengan *state* awal *home* menuju ke *state* lain yang merepresentasikan fungsi yang dimiliki oleh situs web.

2.2 Log File

Pengolahan Web server secara efektif membutuhkan *feedback* mengenai aktivitas dan kinerja server, serta permasalahan-permasalahan yang mungkin terjadi. *Feedback* ini dapat berupa informasi mengenai pengguna-pengguna yang mengakses situs Web, apa yang diakses dan statusnya, serta waktu pengaksesan. Informasi tersebut tersimpan pada log Web server terutama file *access_log*. *Common Log Format* (CLF) adalah format dari konfigurasi informasi yang akan dituliskan ke file *access_log*. Format standar ini dapat dihasilkan oleh berbagai Web server dan dapat dibaca oleh berbagai program penganalisa file log. Format log ini dapat dimodifikasi agar menampilkan informasi yang benar-benar dibutuhkan oleh administrator server.

Common Log Format (CLF) untuk setiap baris data pada file *access_log* adalah sebagai berikut (R. Harol, 1997):

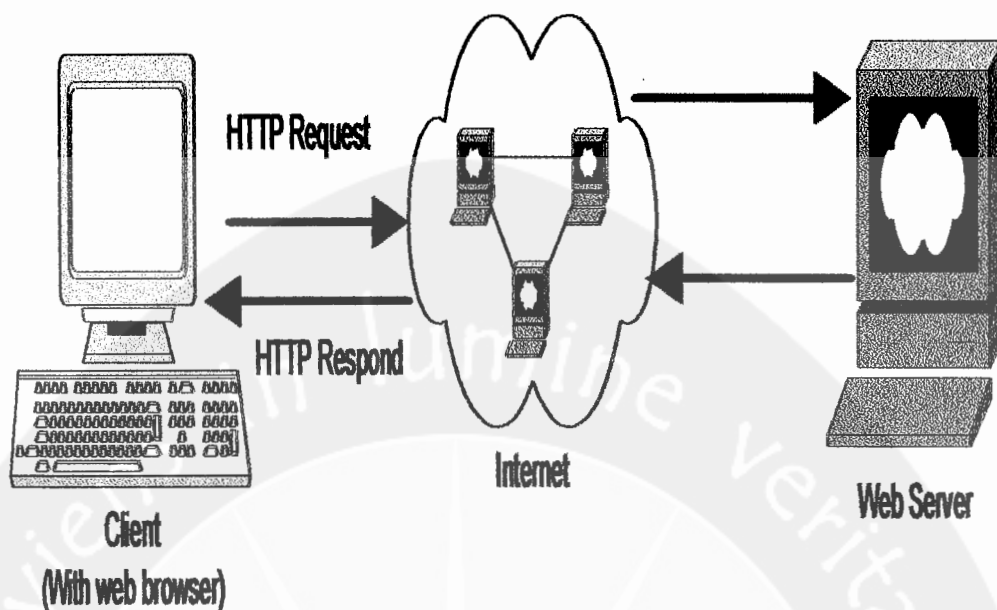
```
Remotehost rfc931 authuser [date] "request" status  
bytes
```

Tabel 1. Common Log Format File access_log

Field	Description
Remotehost	Nama host atau alamat IP dari pengguna yang mengakses situs Web.
Rfc931	<i>Username</i> dari pengguna pada <i>remote system</i> seperti yang dispesifikasikan pada RFC931.
Authuser	<i>Username</i> dari pengguna pada <i>remote system</i> seperti yang dispesifikasikan pada RFC931.
[date]	Informasi tanggal dan waktu pada saat melakukan <i>request</i> HTTP dan merupakan sistem pewaktuan lokal.
"request"	Informasi HTTP <i>request</i> dari pengguna.
Status	Angka numeric yang menyatakan status kode dari HTTP yang dikirimkan ke pengguna.
Bytes	Panjang <i>bytes</i> dari data yang dikirimkan ke pengguna.

2.3 Hyper Text Transfer Protocol (HTTP)

Protokol adalah aturan yang ditetapkan dalam melakukan komunikasi data. HTTP merupakan protokol level aplikasi dengan kemudahan dan kecepatan yang diperlukan untuk sistem informasi terdistribusi dan *hypermedia*. HTTP digunakan untuk mendukung komunikasi di antara *Web Browser* dan *Web Server (HTTP Server)*. HTTP dapat mendukung pertukaran banyak *request & reply* melalui koneksi TCP tunggal dengan nomor port standar 80.



Gambar 2. Model HTTP Request - Respond

2.4 Identifikasi Session

Web browser menggunakan protokol HTTP untuk berkomunikasi dengan *Web server*. Protokol HTTP bersifat *connectionless*, ini berarti setiap permintaan untuk halaman web ditangani langsung oleh *Web server*, kemudian koneksi antara *Web server* dan *browser*-nya akan ditutup. Proses ketika meminta suatu halaman ke situs web dapat dijabarkan sebagai berikut :

1. *Browser* akan membuka koneksi HTTP ke *Web server*.
2. *Browser* akan mengirim HTTP *request* untuk informasi.
3. *Web server* akan merespon permintaan tersebut dan mengirimkan balasannya.
4. Koneksi antara *browser* dan *web server* akan ditutup.

Ketika meminta halaman lain pada situs web yang sama, walau hanya beberapa detik berlalu, sebuah

koneksi akan dibangun dan ditutup seperti pada langkah diatas. Karena koneksi baru harus dibangun setiap kali melakukan permintaan ke *Web server*, *Web server* tidak akan mengetahui apakah yang melakukan permintaan adalah pengunjung yang sama dengan yang melakukan permintaan beberapa detik yang lalu. Dalam bahasa *Web*, ini berarti bahwa *Web server* tidak mengenal *state* atau *session*.

Karena pada sebagian aplikasi bisnis, mengetahui *session* sangat penting dan tidak disediakan HTTP protocol maka muncullah identifikasi *session*. Identifikasi *session* adalah mekanisme yang memungkinkan *Web server* melacak data pengunjung atau *session* yang berjalan dari satu fungsi ke fungsi yang lain dalam situs *Web*.

Ada beberapa kesulitan dalam menentukan *session* dan lamanya waktu pengaksesan terhadap halaman-halaman *Web* secara akurat. Hal ini dikarenakan hal-hal sebagai berikut (Fu, Y. et al, 2002):

- a. *Client* atau *proxy* melakukan *cache* dari halaman *Web* sehingga log pada *Web server* mungkin tidak mendeteksi adanya *request* dari pengguna, karena *client* mengambil dari *cache*.
- b. Pengguna diidentifikasi dengan alamat IP yang digunakan sehingga sulit untuk mendeteksi pengguna-pengguna yang berbagi-pakai alamat IP melalui *proxy*.

Kondisi ini menyebabkan beberapa penelitian sebelumnya menggunakan asumsi dalam menentukan sebuah *session*. Identifikasi suatu *session* pada file log *Web server* dapat dilakukan dengan suatu algoritma tertentu yang mengelompokkan *request-request* dari alamat IP yang

sama dalam suatu rentang waktu tertentu menjadi sebuah *session* (Fu, Y. et al, 2000).

Ada 3 cara untuk mengidentifikasi *session*, yaitu :

1. *Cookies*

Cookie adalah informasi yang disimpan di sisi *client*. Ketika *browser* melakukan permintaan pertama ke sebuah *Web server*, *Web server* akan menciptakan sebuah *session* dengan mengirimkan kembali ID *session* ke *browser* sebagai *cookie*. Untuk permintaan-permintaan berikutnya dari *browser*, *Web server* akan menggunakan informasi dari *cookie* tersebut.

Kelemahan dari mekanisme ini adalah sebagian *browser* lama tidak mendukung *cookies* dan pengguna dapat menon-aktifkan fitur ini pada *browser* jika dirasa tidak nyaman untuk digunakan.

2. *URL rewriting*

Web server juga dapat mengidentifikasi *session* dengan memastikan setiap URL dan setiap *link* yang dikirimkan *server* ke *browser* mempunyai pengidentifikasi *session*.

Contoh:

<http://myexchange.com/offers>

dapat ditulis kembali sebagai:

[http://myexchange.com/offers;\\$sessionid\\$gLDKgAvtdvAJ33k](http://myexchange.com/offers;$sessionid$gLDKgAvtdvAJ33k)

Kelemahan dari mekanisme ini adalah setiap URL harus diberi informasi *session* padanya, jika tidak mendapat informasi *session* menyebabkan pengguna kehilangan *session*-nya.

3. *Hidden form field*

Mekanisme lain adalah dengan menggunakan *Hidden field* pada form HTML. Formatnya dapat ditulis sebagai berikut:

```
<INPUT      TYPE="HIDDEN"      NAME="sessionInfo"
VALUE="username">
```

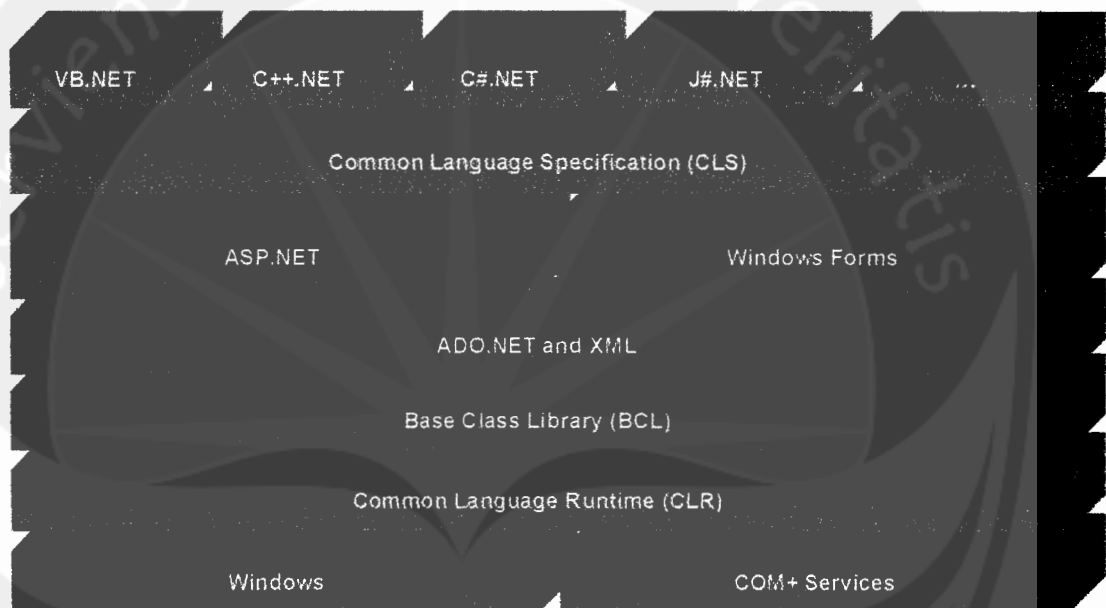
Ketika form di-request, variabel "*sessionInfo*" dan nilai lainnya seperti "*username*" akan dikirimkan ke Web server bersama informasi lain dari form seperti *text field*, *radio button*, *check box* dan lainnya. Jika setiap halaman yang dikunjungi pengguna mengandung *hidden field* dengan informasi *session* didalamnya, Web server dapat dengan mudah mengakses informasi *session* dari pengguna saat pengguna berpindah dari satu halaman ke halaman yang lain.

Kelemahan dari mekanisme ini adalah bahwa setiap halaman harus memiliki *hidden field* yang mengandung informasi *session* didalamnya.

2.5 .Net Framework

Microsoft .NET ialah model pemrograman dari platform .NET untuk membangun, menyebarkan dan menjalankan XML *Web Service* dan aplikasi. .NET *Framework* menyediakan lingkungan berbasis standar produksi yang tinggi untuk memadukan investasi yang ada dengan aplikasi dan service generasi yang akan datang. .NET *Framework* terdiri atas dua bagian utama, yaitu CLR dan gabungan kelas *library* termasuk ASP.NET untuk aplikasi web dan XML *Web Services*, Windows *forms* untuk aplikasi klien dan ADO.NET.

Visual Studio.NET dibangun menggunakan fondasi .NET Framework. .NET Framework menyediakan lingkungan yang cerdas, mudah dikembangkan untuk membangun, menyebarkan dan menjalankan XML *Web Services* yang terdistribusi serta aplikasi. Dalam istilah yang mudah, .NET memisahkan platform Sistem Operasi menjadi dua layer, yaitu sebuah layer pemrograman dan layer eksekusi.



Gambar 3. Arsitektur .Net Framework
(Sumber: <http://www.microsoft.com>)