

Implementasi Session Tracking Pada Mobile Application Dengan Menggunakan Java 2 Platform Microedition

Teguh Susyanto ²⁾

Abstrak

Dalam protokol HTTP, setiap koneksi data yang terjadi antara aplikasi client dengan server tidak ada keterkaitan koneksi antara halaman satu dengan halaman lainnya. Hal ini sangat bertolak belakang dengan model koneksi pada aplikasi desktop, yang memiliki kecenderungan koneksi yang terjadi bersifat persistent, dalam artian koneksi data antar client dengan server akan terus terpelihara sampai salah satunya client atau server yang memutuskan koneksi. Dengan latar permasalahan ini, pengelolaan session dalam protokol HTTP perlu penanganan khusus, terutama untuk memelihara koneksi yang terjadi antara client dengan server. Mekanisme pengelolaan sesi pada aplikasi yang memanfaatkan protokol HTTP terutama dalam aplikasi berbasis web maupun aplikasi mobile dapat dimanfaatkan untuk mengimplementasikan fasilitas shopping-cart dalam e-commerce dan juga fasilitas otentikasi user untuk memvalidasi user yang berhak mengakses aplikasi terkait privasi aplikasi. Session ini dimanfaatkan untuk memberikan kemampuan pada aplikasi untuk 'mengingat' data transaksi yang telah dilakukan sebelumnya, meskipun user telah berpindah-pindah mengakses menu-menu aplikasi lainnya. Pada aplikasi mobile untuk mengelola koneksi data via protokol HTTP diperlukan penanganan khusus, sehingga ini yang dijadikan permasalahan yang diangkat penulis terutama penerapan mekanisme session pada aplikasi mobile berbasis Java 2 Microedition kali ini. Adapun metode yang disajikan dalam penelitian ini adalah menggunakan cookie fetch, untuk membangun fasilitas session.

I. PENDAHULUAN

Protokol HTTP bersifat stateless, yang berarti setiap koneksi data yang terjadi antara aplikasi server dengan aplikasi client melalui HTTP dianggap tidak ada hubungan koneksi antara satu halaman dengan koneksi halaman lain dan tidak bersifat persistent. Sehingga

²⁾ Staf Pengajar STMIK Sinar Nusantara Surakarta

pemrosesan nilai data yang terjadi pada halaman yang satu tidak akan di-'ingat' pada halaman lain. Karena request/permintaan layanan oleh suatu client ke server akan membentuk satu koneksi dan koneksi yang terbangun akan terputus manakala server telah selesai memberi respon kepada client. Request ini terbangun melalui alamat-alamat file (dalam bentuk URL) yang terdapat pada server.

Hal ini akan memerlukan penanganan secara khusus apabila sebuah aplikasi menghendaki mekanisme sesi koneksi yang terpelihara dalam jangka waktu tertentu, yang membiarkan aplikasi client men-request alamat yang berpindah-pindah untuk keperluan dalam transaksi aplikasi dan server tetap dapat mengingat nilai atau kejadian yang telah diproses/dilakukan oleh client. Mekanisme session ini terutama diperlukan untuk membuat layanan aplikasi keranjang belanja dalam aplikasi e-commerce, validasi user untuk membuat fasilitas login-logout, dan lain-lain.

Dalam aplikasi J2ME terutama untuk mengimplementasikan aplikasi database, protokol HTTP memegang peranan yang cukup signifikan dalam membangun aplikasi multi-tier. Karena aplikasi J2ME dalam ponsel fasilitas programnya sangat terbatas. Sehingga pengendalian pemrosesan ditimpakan pada aplikasi server, seperti pengaksesan record-record tabel dalam DBMS, di sini aplikasi yang dibangun dengan J2ME hanya difokuskan pada pemicuan-pemicuan pemrosesan oleh server. Seperti pemicuan validasi user yang melakukan login dalam aplikasi client, sementara data user-user yang berhak akses tersimpan dalam komputer server.

II. PERUMUSAN MASALAH

Pada proyek penelitian ini fokus permasalahan yang diangkat yaitu bagaimana mengimplementasikan mekanisme session dalam client-server dengan menggunakan java 2 microedition.

III. TUJUAN

Tujuan dari penelitian ini adalah menciptakan sebuah prototipe aplikasi *wireless mobile* untuk mengimplementasikan mekanisme session pada aplikasi ponsel dengan menggunakan teknik *cookie fetch*.

IV. METODE PENELITIAN

4.1. Obyek Penelitian

Dalam penelitian ini mencoba menerapkan mekanisme session pada aplikasi mobile dengan menggunakan teknik *cookie*

fetch. Aplikasi dibuat dalam bentuk simulasi validasi user login-logout yang dijalankan dengan menggunakan emulator dalam PC Desktop, sedangkan database dan skrip pengelola datanya dibuat masing-masing menggunakan software *MySQL*, *PHP* dan *Apache Webserver*. Respon data yang dikirimkan balik oleh server ke client midlet adalah berupa format *XML*.

4.2. Sumber Data

Sumber data untuk mendapatkan pedoman pembuatan program dan perancangan sistem digunakan metode studi pustaka dari buku-buku maupun dengan browsing lewat internet yang membahas tentang penggunaan teknologi aplikasi *java mobile*.

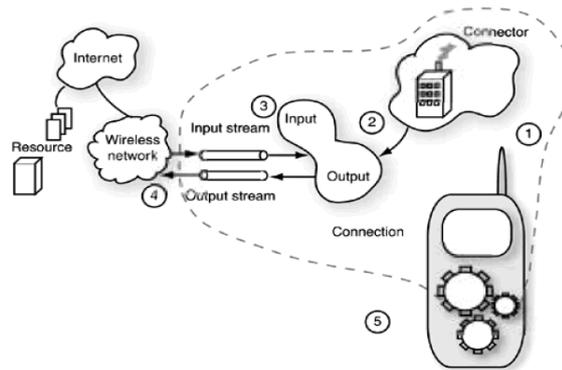
4.3. Java 2 Microedition (J2ME)

Merupakan perangkat software yang dikembangkan oleh Sun Microsystems Corporation, yang memungkinkan perangkat micro seperti Handphone, PDA, maupun perangkat kecil lain, mampu menjalankan aplikasi-aplikasi khusus berbasis java pada perangkat tersebut. Sebagian besar vendor-vendor produk seluler telah menyertakan lingkungan software java (*Java Virtual Machine*), sehingga perangkat seluler tersebut dapat menjalankan aplikasi-aplikasi yang dibuat dengan *platform java*. Vendor-vendor seperti Nokia, Siemen, Motorola, RIM, Palm telah banyak menyematkan dukungan perangkatnya pada *platform java*.

J2ME memiliki komponen-komponen yang terdiri dari Java Virtual Machine (JVM), Java API (Application Programming Interface) dan Paket-paket opsional lain untuk pengembangan aplikasi java. Bagian paling pokok pada J2ME adalah bagian *configuration*, *Profile* dan *Optional Packages*.

V. PEMBAHASAN

Dalam aplikasi java midlet yang memanfaatkan koenksi jaringan dengan menggunakan protokol HTTP, proses request dan response dari client ke server dan sebaliknya dapat digambarkan dalam bentuk diagram berikut ini :



Skema Arsitektur Koneksi Data Java ME

Menganut arsitektur koneksi data via HTTP di atas, implementasi pengelolaan session memanfaatkan bentuk aplikasi client-server tiga lapis (three-tier), dengan aplikasi midlet sebagai aplikasi client (user interface) sebagai lapis pertama, aplikasi server menggunakan PHP dan Apache Webserver sebagai aplikasi lapis kedua (*middle-tier*), yang berfungsi untuk mengakses database MySQL berdasarkan request dari aplikasi client, dan sebagai lapis terakhir adalah DBMS MySQL sebagai penyedia datanya.

Aplikasi server disini berfungsi untuk menentukan apakah sesi baru perlu dibentuk apa tidak, hal ini tergantung apakah client telah melakukan proses login dengan account yang valid. Jika account yang dimasukkan valid maka aplikasi server akan menciptakan sesi baru. Namun jika tidak valid maka sesi tidak akan diciptakan.

5.1. Aplikasi Server (Middle-Tier)

Merupakan aplikasi perantara yang mengotorisasi user dan mengidentifikasi pengenal sesi dari sebuah request yang dikirimkan oleh client.

```
<?php
... ..
$nim=addslashes($_POST['usrid']);
$psw=addslashes($_POST['paswd']);
if ( otentikasi($nim, $psw) ){
    // proses pembuatan sesi jika account user login valid
    $account=sprintf("%s|%s", $nim, $psw);
    setcookie("id", $account);
    $msg = "<status>TRUE</status>\n";
}
```

```

    $msg .= "<keterangan>$pesan</keterangan>\n";
}else{
    $msg = "<status>FALSE</status>\n";
    $msg .= "<keterangan>Login user gagal!</keterangan>\n";
}
... ..
?>

```

Kelompok perintah-perintah PHP di atas berfungsi untuk menciptakan sesi baru dengan memanfaatkan fasilitas cookie di PHP. Dengan memanfaatkan fungsi **setcookie()** untuk me-registrasi variable “**id**” sehingga nama variable ini akan tetap dipelihara selama periode waktu tertentu. Nilai variabel ”**id**” dibuat dalam bentuk format sebagai berikut :

id=userid | password

Setelah variabel account diregistrasikan menjadi variabel ”COOKIE”, server akan mengirimkan balik variabel tersebut ke client melalui HEADER responnya. Berikut ini merupakan contoh respon dari webserver :

```

HTTP/1.1 200 OK
Content-Type: text/xml
content-length: 159
date: Thu, 14 May 2009 07:36:54 GMT
server: Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9
OpenSSL/0.9.8i mod_autoindex_color PHP/5.2.6
set-cookie: id=04.1.00001%7Ccinta

```

berdasarkan bentuk header respon server di atas, apabila dilakukan proses login dengan menggunakan userid : **04.1.00001** dan password : **cinta**, sehingga didapatkan hasil respon dari server berupa variable dan nilainya

set-cookie: id=04.1.00001%7Ccinta

Simbol dalam “%7C” merupakan bentuk pengkodean dari symbol karakter pipa (”|”). Header respon dengan nama field ”set-cookie” inilah yang akan disimpan dalam program client, dan akan disertakan saat client melakukan request-request file/halaman lain dalam server. Demikian juga program server juga akan selalu menanyakan client yang meminta request, apakah telah memiliki atau melampirkan variable “**id**” dalam header request. Jika client tidak melampirkan variable **id** yang telah teregistrasi aplikasi server akan segera menolaknya. Perhatikan perintah-perintah aplikasi server berikut ini,

yang berfungsi untuk memvalidasi client yang memiliki otoritas meminta request.

```
<?php
if (isset($_COOKIE["id"])){
    $sub2=explode("|",$id);
    if(is_array($sub2)){
        $nim=$sub2[0];
        $psw=$sub2[1];
        if (otentikasi($nim,$psw)){
            switch($_POST['mod']){
                // menu aplikasi pertama
                case 101: $msg=mod_krs($nim, $_POST['thsms']); break;
                // menu aplikasi kedua
                case 102: $msg=mod_khs($nim, $_POST['thsms']); break;
                // menu logout
                case 103: unset($_COOKIE["id"]); break;
            }
        }
    }
}
}else{
    $msg = "<response>\n";
    $msg .= "<status>FALSE</status>\n";
    $msg .= "<keterangan>Anda belum login!</keterangan>\n";
    $msg .= "</response>\n";
}
header("Content-Type: text/xml");
echo("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
echo "<root>\n";
echo ($msg);
echo "</root>\n";
?>
```

File server di atas digunakan untuk memverifikasi variabel cookie yang bernama "id" dari client, apabila variabel tersebut tidak ada atau belum ditentukan nilai maka tidak akan diperkenankan untuk mengakses menu-menu yang ada di dalamnya.

5.2. Aplikasi Client (MIDlet)

Aplikasi midlet ini sebagai user interface berupa form LOGIN dan untuk menginputkan data login user. Dalam proyek penelitian ini telah diterapkan prosedur Login, prosedur ini berfungsi untuk membuat request validasi user dan password yang dikirimkan ke file server.

```
private String cookie = null;
... ..
public void login(String user, String pass)
```

```

throws IOException {
    InputStream iStrm=null;
    OutputStream oStrm=null;
    HttpURLConnection http=null;
    String url="http://localhost/mobile/login.php";

    try{
        http=(HttpURLConnection) Connector.open(url);
        http.setRequestMethod(HttpURLConnection.POST);
        http.setRequestProperty("Content-Type",
            "application/x-www-form-urlencoded");

        if (cookie!=null)
            http.setRequestProperty("cookie", cookie);
        String param="usrid="+user+"&paswd="+pass;
        oStrm=http.openOutputStream();
        oStrm.write(param.getBytes(param.getBytes()));
        oStrm.flush();

        if (http.getResponseCode()==HttpURLConnection.HTTP_OK){
            String tmpCookie=http.getHeaderField("set-cookie");
            if (tmpCookie!=null)
                cookie=tmpCookie;
            iStrm=http.openInputStream();
        }
    }finally{
        if (oStrm != null)
            oStrm.close();
        if (http != null)
            http.close();
    }
    return iStrm;
}

```

Pada kode perintah-perintah di atas dideklarasikan variabel global dalam kelas bernama **cookie**. Variabel ini yang akan dilampirkan setiap kali client midlet melakukan request ke server dengan menggunakan **setRequestProperty("cookie", cookie);** dengan demikian server akan mengetahui, bahwa client yang meminta request data adalah user yang telah diotorisasi. Adapun potongan kode perintah untuk mengidentifikasi cookie client adalah sebagai berikut ini :

```

if (isset($_COOKIE["id"])){
    $sub2=explode("|",$id);
    if(is_array($sub2)){
        $nim=$sub2[0];
        $psw=$sub2[1];
        if (otentikasi($nim,$psw)){

```

```

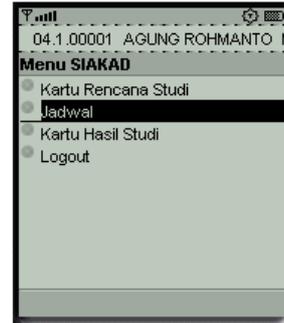
... ..
... ..
}
} }

```

Apabila nilai `$_COOKIE["id"]` belum bernilai (masih *null*) maka client dianggap belum melakukan login, namun sebaliknya jika telah ditentukan nilainya program akan memvalidasi cookie yang ada dan dicocokkan lagi sesuai dengan koleksi user dalam database yang memiliki akses dalam program. Berikut ini merupakan hasil implementasi tampilan midlet :



Form Login User



Tampilan Setelah Login

VI. KESIMPULAN

Dalam implementasi *tracking session* dalam aplikasi java mobile harus mematuhi aturan-aturan yang berlaku dalam protokol HTTP yang bersifat stateless, pengendalian session dengan menggunakan teknik *cookie fetch* diproses melalui mekanisme request dan response yang terjadi antar dua antarmuka, yaitu aplikasi midlet dan aplikasi server. Setiap request yang ingin di-‘anggap’ dalam satu periode sesi, wajib melampirkan pengenalan cookie yang dilampirkan pada header request.

DAFTAR PUSTAKA

1. Qusay Mahmoud, 2001, *Learning Wireless Java*, O'Reilly.
2. John W. Muchow, 2001, *Core J2ME™ Technology & MIDP*, Prentice Hall PTR.
3. Achmad Imam Kistijantoro, 2007, *Materi Kuliah : Java ME*, Fakultas Informatika ITB.
4. <http://kxml.sourceforge.net/kxml2/>
5. <http://java.sun.com/docs/books/tutorial>