

Implementasi Metode Realtime, Live Data Dan Parsing JSON Berbasis Mobile Dengan Menggunakan Android Studio Dan PHP Native

Ahmad Roihan, Aditya Agus Wisanto, Yudi Sulaeman, Fuad Muhammad Nur, Sofian Williandi, Wahyu Pribadi

Universitas Raharja

ahmad.roihan@raharja.info, aditya.agus@raharja.info, yudi.sulaeman@raharja.info, fuad.muhammad@raharja.info, sofian.williandi@raharja.info, wahyu.pribadi@raharja.info

ABSTRAK

Pengolahan data secara konvensional dirasa kurang efektif dalam penerapannya dalam penelitian ini kami membuat sebuah aplikasi yang dapat mengelola data mahasiswa secara mobile sehingga mengelola data menjadi mudah. Dalam penelitian ini kami juga bertujuan untuk menguji secara empiris pengaruh pengalaman sistem terdistribusi data, integrasi data, data langsung dan data setiap waktu. Analisis ini menggunakan berbagai pendalaman pada materi yang dipelajari dan keahlian analisis data pada database. Sampel penelitian ini adalah aplikasi android yang digunakan untuk pendaftaran seminar. Sampel ini dilakukan dengan metode Restful HTTP data yang dimana data dapat dibuat, data dapat dibaca, data dapat dihapus, dan data dapat diperbaharui. Teknik pendistribusian data dilakukan oleh pemegang aplikasi tersebut (admin) lalu dikirimkan melalui web service PHP Rest FullAPI ke database localhost (hosting www.000webhost.com) untuk menyimpan data yang telah diinputkan melalui aplikasi tersebut. Hasil penelitian ini menunjukkan bahwa pendistribusian data mempengaruhi kinerja proses pada pengiriman data dengan melewati beberapa metode – metode di belakang layar user interface penginputan data. Tetapi tugas pendistribusian tidak secara signifikan mempengaruhi kinerja aplikasi yang diolah datanya sedemikian rupa.

Kata kunci: API, Android, Sistem Terdistribusi, JSON

ABSTRACT

Conventional data processing is considered less effective in its implementation in this study we made an application that can manage student data in a mobile so that managing data becomes easy. In this study we also aim to empirically examine the influence of data distributed system experience, data integration, direct data and data at any time. This analysis uses a variety of deepening of the material being studied and data analysis skills in the database. This research sample is an android application used for seminar registration. This sample is done with the Restful HTTP data method where data can be created, data can be read, data can be deleted, and data can be updated. Data distribution technique is carried out by the application holder (admin) and then sent via the PHP Rest FullAPI web service to the localhost database (hosting www.000webhost.com) to store the data that has been inputted through the application. The results of this study indicate that the distribution of data affects the performance of the process of sending data by passing several methods behind the scenes of the data input user interface. But the task of distributing does not significantly affect the performance of applications that are processed in such a way.

Keyword : API, android, Distributed System, JSON

PENDAHULUAN

Pembuatan aplikasi mobile yang berbasis android di jaman sekarang ini telah mencapai kemajuan yang sangat besar terutama aplikasi yang berhubungan dengan pendataan mahasiswa dalam sebuah seminar. Sistem aplikasi yang berhubungan dengan pendataan mahasiswa merupakan salah satu bagian yang sangat penting dalam pengelolaan data. Penggunaan sistem yang terkomputerisasi akan lebih banyak menghemat waktu, tidak banyak menyita tenaga, dan menghasilkan keakuratan penyajian data apalagi ditambah dengan sistem *database* sebagai media penyimpanan datanya.

Mebutuhkan kecepatan mentransfer data untuk melihat data yang masuk secara langsung dan dapat mencari suatu data dengan cepat dengan menggunakan konsep sistem terdistribusi. Mengelola data dari mahasiswa masih menggunakan cara manual, artinya dari segi pencatatan dan pengolahannya masih menggunakan selembar kertas berupa persediaan. Setiap pendaftaran data tersebut harus di update guna mengetahui berapa mahasiswa yang telah terdaftar. Betapa sulitnya jika penyimpanan data mahasiswa menggunakan cara konvensional seperti itu dan masih offline.

METODE

Metode eksperimental yang dilakukan pada penelitian ini adalah: Menurut Sugiyono metode observasi adalah merupakan teknik pengumpulan data mempunyai ciri spesifik bila dibandingkan dengan teknik yang lain, yaitu wawancara dan angket dan juga metode yang dilakukan dengan suatu pengamatan atau kegiatan yang sistematis terhadap objek yang dituju secara langsung dengan menggunakan indera mata.[1]

Upaya yang dilakukan oleh penulis untuk menghimpun segala informasi diperoleh dari buku-buku, laporan penelitian, karangan ilmiah, jurnal nasional dan

internasional disertai dan sumber-sumber lain. Selain itu, penulis juga perlu memanfaatkan hasil penelitian dan pemikiran yang relevan dengan masalah penelitian yang saat ini penulis lakukan untuk menghindari terjadinya pengulangan penelitian serupa atau duplikasi yang tidak diinginkan. Dalam hal ini peneliti mengambil dasar teori dari beberapa literasi jurnal, berikut dasar teori yang peneliti gunakan dalam penelitian ini :

Sistem Terdistribusi merupakan sekumpulan otonom yang terhubung ke jaringan, dimana bagi pengguna sistem terlihat seperti satu komputer. Sedangkan otonomi sendiri memungkinkan komputer dapat berjalan walaupun komputer tidak terhubung ke jaringan.[2]

Jaringan pembentuk didalam sistem terdistribusi dibangun dari berbagai macam media transmisi seperti transmisi kabel, transmisi fiber optik, maupun jaringan yang menggunakan sistem wireless termasuk didalamnya bagian hardware seperti *router, switch, hub, repeater* dan *network interface* lainnya dan bagian software seperti protokol stack, pengatur komunikasi dan driver.[3]

Karakteristik sistem terdistribusi adalah sebagai berikut :

Concurrency of components.

Mengakses sumber daya dengan bersamaan dan dapat tetap berjalan, dalam hal ini menambahkan bandwidth dalam sebuah jaringan untuk menambah kemampuan jaringan tersebut untuk menampung user secara bersama - sama. Sebagai contoh : User mengakses suatu web secara bersamaan dengan lancar lalu ada *No global clock*. Hal ini menyebabkan kesulitan dalam menyelaraskan waktu seluruh komputer/perangkat yang terlibat. Dapat berpengaruh pada pengiriman pesan/data, seperti saat beberapa proses berebut ingin masuk ke *critical session* serta yang terakhir *Independent failure of component*. Setiap perangkat dapat mengalami kegagalan namun perangkat lain tetap dapat berjalan dengan baik.[4]

Menurut Abdurahman Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti pada jurnal, pengertian aplikasi adalah program yang digunakan orang untuk menjalankan serta melakukan perintah pada sistem komputer.[5] Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.[6] Jadi dapat ditarik pengertian bahwa sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel, seperti telepon seluler dan PDA.

Sedangkan android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux.[7] Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak.

Real-time menurut Kang G. Shin dalam penjabaran adalah aplikasi yang terdiri atas satu set alat yang bekerja secara bersamaan pada jangka waktu yang sama pula. Seperti contoh aplikasinya dalam dunia penerbangan, dimana keterlambatan tidak dapat ditolerir sedikitpun karena beresiko membahayakan banyak nyawa penumpang. [8]

JSON adalah suatu format pertukaran data yang memiliki tipe data *light-weight key-value*. JSON sendiri adalah JavaScript Object Notation dimana Formatnya berbentuk teks serta dapat dibaca oleh manusia dan juga digunakan untuk menggambarkan struktur data sederhana

dan larik asosiatif (disebut objek). Format JSON acapkali digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi. [9]

PEMECAHAN MASALAH

Berdasarkan pengamatan sebelumnya terhadap masalah yang dihadapi pada sistem seminar konvensional, penelitian ini ingin memecah masalah ini dengan memberikan solusi berupa pembuatan aplikasi yang nantinya aplikasi tersebut dapat mempermudah pendataan para peserta seminar dengan memanfaatkan JSON untuk memarsing data sehingga data dapat dilihat secara *Real Time*. Nantinya, distribusi data akan disebar dengan menggunakan layanan *RestHTTP* yang sebelumnya akan disimpan dalam suatu database di localhost. Aplikasi juga dapat dibuka di banyak device dan data juga dapat dicari bersamaan dengan penginputan data atau *Live Data*.

Alur jalannya aplikasi sesuai dengan pendataan seminar konvensional pada umumnya. Mahasiswa diminta untuk memasukkan data berupa NIM (Nomor Induk Mahasiswa), Nama, Jurusan, Gender dan Alamat. Setelah itu, data akan dikirim ke database localhost namun sebelumnya data di decode terlebih dahulu dalam bentuk JSON karena di dalam pembuatan aplikasinya di encode setiap data yang dikirim melalui *Request HTTP*. NIM pada *database* diatur menjadi *Primary Key* untuk mencegah redudansi data pada *database*. Sistem ini dirancang mampu dibuka secara bersamaan dari perangkat yang berbeda dalam memenuhi kriteria sistem terdistribusi. Dalam pembuatan aplikasi ini, peneliti membuat aplikasi berbasis *Android* dengan tujuan untuk mobilitas sistem. Peneliti menggunakan Android Studio sebagai IDE utama untuk menulis kode program dan men-*debug* aplikasi ini. Langkah pertama dalam pembuatan aplikasi ini dalam Android Studio IDE adalah dengan mengimpor beberapa API seperti Retrofit dan Converter Gson dan

dalam sisi design harus mengimpor Android design dan support seperti pada gambar di bawah ini:

```
//design
implementation 'com.android.support:design:29.0.0'
implementation 'com.android.support:support-v4:29.0.0'

//api
implementation 'com.squareup.retrofit2:retrofit:2.6.0'
implementation 'com.squareup.retrofit2:converter-gson:2.6.0'
```

Gambar 1. Import API dan Design

Fungsi peneliti mengaktifkan desig adalah untuk mengaktifkan RecyclerView. Untuk penggunaan API Converter Gson sendiri adalah untuk mengkonversikan file JSON ke service android studio dalam pembacaan data berformat JSON karena Android Studio tidak bisa membaca file langsung dari database, harus dikonversikan terlebih dahulu dengan format JSON.

Karena nantinya peneliti akan memanfaatkan layanan Web Service, maka langkah selanjut kan adalah mengatur *permission* untuk menyetujui penggunaan internet. Nantinya, permission ini akan diletakkan pada `app/manifest/AndroidManifest.xml`.

```
<uses-permission android:name="android.permission.INTERNET" />
```

Gambar 2. Set Permission untuk terkoneksi dengan internet

Tahap Selanjutnya adalah membuat sebuah class bernama Mahasiswa. Class ini nantinya akan memiliki informasi yang menampung tentang NIM, Nama, Jurusan, Gender, Value dan Result. Seperti pada gambar di bawah ini:

The screenshot shows a Java class named `Mahasiswa` with the following code:

```
private String alamat;

@SerializedName("value")
private String value;

@SerializedName("message")
private String message;

@SerializedName("result")
List<Mahasiswa> dataMahasiswa;

//alt + insert Num LK lalu pilih
//Getter and Setter
```

Overriding the `toString` method:

```
public String toString() {
    return "Mahasiswa{" +
        "alamat=" + alamat +
        ", value=" + value +
        ", message=" + message +
        ", dataMahasiswa=" + dataMahasiswa +
        "}";
}
```

On the right, a dialog titled "Select Fields to Generate Getters and Setters" is open, showing a tree view of the class fields: `nim:String`, `nama:String`, `jurusan:String`, `gender:String`, `alamat:String`, `value:String`, `message:String`, and `dataMahasiswa>List<Mahasiswa>`. The "value" field is selected.

Gambar 3. Class Mahasiswa

Fungsi dari `@SerializedName (" ")` adalah untuk data pada file `lihatMahasiswa.php` ketika di parsing datanya dalam bentuk JSON. `lihatMahasiswa.php` nantinya akan dipergunakan untuk menangkap data yang sudah dikirimkan melalui *Request HTTP* di dalam method Android Studio nya. Kemudian, fungsi `private` dalam class digunakan agar isi dari class tidak bisa dipanggil dari class lain untuk menjaga kerahasiaan data. Sementara fungsi `List<mahasiswa>` adalah bentuk array object dimana object resultnya adalah `[data mahasiswa]`.

Pada tahap selanjutnya, peneliti membuat class bernama `ApiClient`. Class ini berisi library `Retrofit` yang fungsinya adalah untuk memanggil *link address* agar bisa diakses di server hosting. Dalam aplikasi ini, peneliti menggunakan `000webhost` sebagai web hosting nya.

```

public class ApiClient {

    private static final String BASE_URL = "https://seminarhimpifapp.000webhostapp.com/";

    private static Retrofit retrofit;

    public static Retrofit getApiClient() {
        if (retrofit == null) {
            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}
    
```

Gambar 3. Pemanggilan API Retrofit

Langkah berikutnya adalah membuat class baru bernama ApiInterface. Class ini berfungsi untuk penamaan file php di belakang tautan web hosting yaitu <https://seminarhimpifapp.000webhostapp.com/>

HASIL

```

public interface ApiInterface {
    @FormUrlEncoded
    @POST("addMahasiswa.php")
    Call<Mahasiswa> addMahasiswa(@Field("nim") String nim,
        @Field("nama") String nama,
        @Field("jurusan") String jurusan,
        @Field("gender") String gender,
        @Field("alamat") String alamat
    );

    @FormUrlEncoded
    @POST("editMahasiswa.php")
    Call<Mahasiswa> editMahasiswa(@Field("nim") String nim,
        @Field("nama") String nama,
        @Field("jurusan") String jurusan,
        @Field("alamat") String alamat
    );

    @FormUrlEncoded
    @POST("hapusMahasiswa.php")
    Call<Mahasiswa> hapusMahasiswa(@Field("nim") String nim);

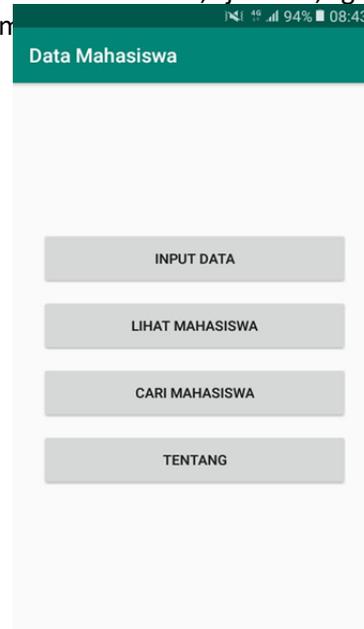
    @GET("lihatMahasiswa.php")
    Call<Mahasiswa> lihatMahasiswa();

    @FormUrlEncoded
    @POST("cariMahasiswa.php")
    Call<Mahasiswa> cariMahasiswa(@Field("search") String search);
}
    
```

Gambar 4. Script menu UI

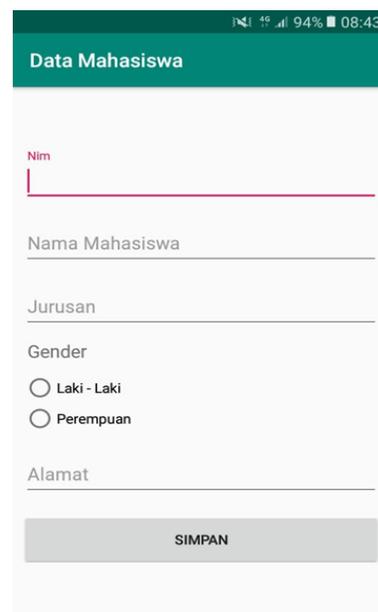
Dalam desainnya, aplikasi ini mempunyai *User Interface* yang sederhana dengan 4 menu utama yaitu input data, lihat mahasiswa, cari mahasiswa, dan tentang. Hal ini merupakan tahapan pengujian akhir aplikasi karena beberapa pengguna akan menggunakan aplikasi ini dengan menginput data sebagai data mahasiswa

yang akan disimpan. Gambar 5 di bawah ini adalah ikhtisar dari tampilan halaman utama dari aplikasi pada menu data mahasiswa, menu ini digunakan untuk mengisi data mahasiswa yang terdiri dari nim, nama mahasiswa, jurusan, gender, dan alamat

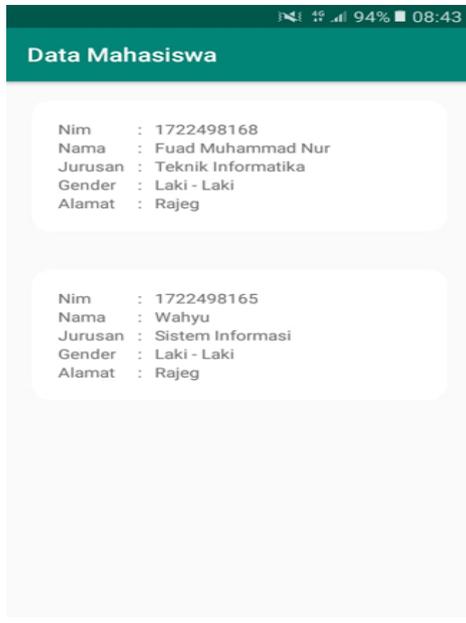


Gambar 5. Halaman utama Data Mahasiswa

Hasil selanjutnya adalah gambar 6 yang menunjukkan tampilan input mahasiswa yang menampilkan form data mahasiswa.

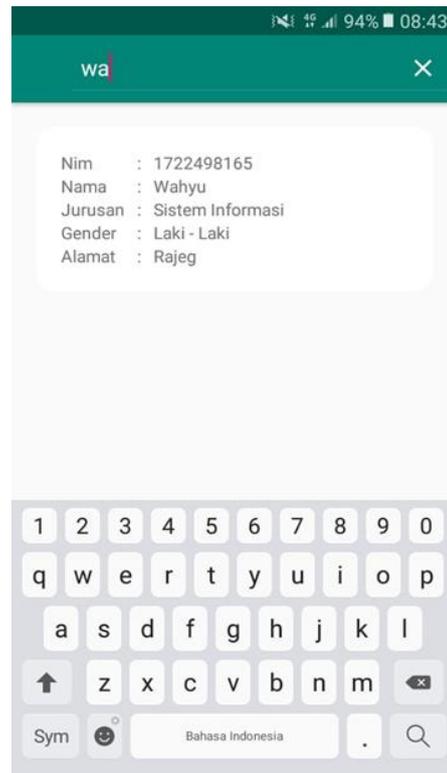
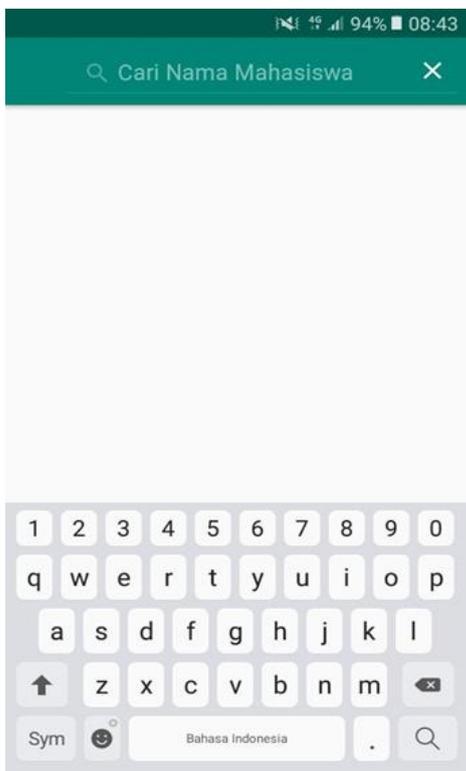


Gambar 6. Tampilan menu lihat mahasiswa
Berikutnya pada gambar 7
menampilkan data mahasiswa yang telah
berhasil diinput



Gambar 7. Tampilan data mahasiswa

Sedangkan pada Gambar 8
menunjukkan tampilan untuk mencari data
mahasiswa yang telah disimpan.



Gambar 8. Tampilan menu cari mahasiswa

Agar aplikasi mendapatkan hasil maksimal dan berkualitas dalam proses implementasi, proses pengujian yang dapat dilakukan dengan menggunakan metode *performance testing*. Dalam hal ini integration dan usability test yang menentukan apakah system or subsystem dapat memenuhi kriteria kinerja berbasis waktu seperti response time atau throughput. Response time menentukan batas waktu maksimum yang diijinkan dari respon software untuk query dan update. Throughput menentukan jumlah minimum query dan transaksi yang harus diproses per menit atau per jam.

Dalam memasukkan data mahasiswa, user hanya harus membuka aplikasi dan langsung input. Hal ini hanya dilakukan user yang berperan sebagai admin bagian pendataan. Hingga selanjutnya data dapat dilihat oleh petugas atau panitia lain dan dapat menampilkan list mahasiswa yang telah diinput sebelumnya oleh admin. Sistem

menggunakan database sehingga pembacaan akan akurat dan tepat.

KESIMPULAN DAN SARAN

Kesimpulan

- Berdasarkan hasil rancangan sistem dan implementasinya, dapat ditarik kesimpulan bahwa sistem ini :
- jauh lebih mudah untuk dioperasikan dibandingkan dengan cara sebelumnya yang masih konvensional.
- pengolahan data akan semakin mudah dalam pendokumentasiannya hingga dapat memudahkan semua pihak yang terlibat dengan sistem aplikasi mobile ini.
- Aplikasi berbasis mobile merupakan pilihan tepat yang digunakan oleh panitia atau suatu Himpunan Mahasiswa Jurusan yang akan mengadakan suatu event seperti seminar, workshop dan lain-lain.
- dokumentasi suatu event tidak mudah hilang.

DAFTAR PUSTAKA

- [1] Sugiyono. (2009). Metode Penelitian Kuantitatif, Kualitatif, dan R&D. Bandung: CV ALFABETA.
- [2] Niswatin, Ratih Kumalasari. Seminar Nasional Teknologi Informasi dan Multimedia 2013. (2013). *Sistem Informasi Terdistribusi Pada Manajemen Inventarisasi Peralatan Laboratorium*.
- [3] Prihantoro, Cahyo. & Witriyono, Harry. Journal of Technopreneurship and Information System. (2019). *Perancangan Client Server Three Tier pada Pembangunan Web Service Anggota Perpustakaan Universitas Muhammadiyah Bengkulu*.
- [4] G. Coulouris, J. Dollimore, Tim Kindberg, Gordon Blair, "Distributed System Concept and Design Fifth Edition". Boston : Addison Welsey. 2012
- [5] Abdurahman, Hasan. Riswaya, Ririh Asep. *Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti*. Jurnal Computech & Bisnis : Vol. 8 No. 2, Desember 2014, 61-69
- [6] Suryani, Des. Yulianti, Ana. & Zulhelmi, Muhammad. (2018). *Aplikasi Legalitas Surat Izin Mengemudi (SIM) Berbasis Mobile (Studi Kasus : Polisi Resort Rengat)*. IT Journal Research and Development Vol.2, No.2, Maret 2018
- [7] Wei-Meng Lee, 2011, *Beginning Android Application Development*, English, Wrox.
- [8] Shin, Kang G., Fellow, & Ramanathan Parameswaran. (1994). *Real-time Computing: A New Discipline Of Computer Science And Engineering*. Proceedings Of The IEEE, Vol. 82, No. 1, January 1994
- [9] Peng, Dunlu. Cao, Lidong. & Xu, Wenjie. (2011). *Using Json For Data Exchanging In Web Service Applications*. Journal Of Computational Information Systems 7: 16 (2011) 5883-5890