Implementasi Protokol OAuth 1.0 Sebagai Autentikasi pada Aplikasi SMS Blast Berbasis Android

Kurnia Saputra*, Khalid Farhan
Jurusan Informatika, Fakultas MIPA, Universitas Syiah Kuala
Darussalam – Banda Aceh, 23111
*Email: kurnia.saputra@unsyiah.ac.id

Abstract

Keamanan dalam pertukaran data pada aplikasi mobile berbasis Android merupakan sebuah hal penting yang perlu dilakukan. Pada penelitian ini telah dikembangkan sebuah aplikasi SMS Blast berbasis Android yang menggunakan protokol OAuth 1.0 sebagai sebuah model autentikasi client-server untuk mengamankan pengambilan data nomor telepon melalui web service berbasis RESTful. Data nomor telepon yang digunakan adalah nomor telepon alumni Unsyiah yang diperoleh dari database Exit Survey CDC Unsyiah. Untuk pengambilan data disiapkan sebuah Application Programming Interface (API) berbasis RESTful yang selanjutnya proses pertukaran data diamankan dengan menggunakan protokol OAuth 1.0. Dengan mengimplementasikan protokol autentikasi OAuth diharapkan dapat mengurangi kemungkinan serangan dan pencurian data.

Keywords: Android, Web Service, OAuth, RESTful, SMS Blast.

I. PENDAHULUAN

Android merupakan sistem operasi yang dirancang untuk perangkat *mobile*. Android berisi *kernel* bebasis Linux OS, memiliki banyak fitur seperti *user interface, end-user applications, code libraries,* aplikasi *frameworks,* dukungan terhadap multi media, dan masih banyak lagi. Disamping itu fungsi telepon juga disertakan dalam sistem operasi Android. Beberapa komponen pada system operasi Android ditulis menggunakan bahasa pemrograman C atau C++, akan tetapi aplikasi yang digunakan oleh pengguna ditulis dengan bahasa pemrograman Java yang menggunakan Android *Software Development Kit* (*SDK*)[1].

Pada penelitian ini aplikasi yang akan dikembangkan adalah aplikasi *SMS Blast,* dimana aplikasi ini memiliki fitur utama yaitu mengirim pesan singkat atau *Short Message Service* (SMS) ke banyak nomor sekaligus. Data nomor telepon didapat dari basis data*Exit Survey* Career Development Centre (CDC) Universitas Syiah Kuala. Pengembangan aplikasi *SMS Blast* berbasis Android dipilih karena mengingat banyaknya pengguna perangkat mobile berbasis sistem operasi Android, disamping itu Android juga merupakan sebuah sistem operasi *open source* yang tentunya membuat pengembangan aplikasi akan menjadi lebih mudah.

Protokol yang digunakan pada penelitian ini adalah OAuth 1.0. Dalam model autentikasi *client-server* tradisional, *client* menggunakan hak akses mereka sendiri untuk mengakses *resources* yang tersimpan di *server*. Dengan meningkatnya penggunaan *distributed web service* dan penggunaan *cloud computing,* maka diperlukan protokol yang memungkinkan aplikasi pihak ketiga *(third-party aplications)* untuk bisa mengakses *resources* yang disimpan di *server.* OAuth hadir dengan memperkenalkan metode baru, yaitu *resource owner.* Pada protokol OAuth, *client* (bukan *resource owner*, tetapi bertindak atas nama *resource owner*) meminta akses *resource* dimana proses permintaan ini dikontrol oleh *resource owner*[2].

Pada penelitian ini, aplikasi SMS Blast berperan sebagai aplikasi pihak ketiga *(third-party aplications)*, dimana aplikasi ini akan melakukan permintaan *resource*. *Resource owner* disini adalah CDC Unsyiah. Proses permintaan *resource* oleh aplikasi sebelumnya harus disetujui oleh CDC Unsyiah sebagai *resource owner*.

Proses permintaan *request* dari aplikasi akan dilakukan melalui *web service* RESTful.

Dengan *web service*memungkinkan aplikasi saling bertukar data meski aplikasi ditulis menggunakan bahasa pemrograman yang berbeda. *Web service* diletakkan pada suatu tempat yang bisa diakses melalui *standard-based Internet protocols* seperti HTTP atau SMTP [3].

RESTful adalah sebuah *Application Programming Interface* (API) yang mengikuti *style* dari *Representational State Transfer* (REST). REST adalah *stateless* dan *resource-oriented*. Semua yang ada pada REST arsitektur berhubungan dengan sumber daya *(resource).* Setiap *request* adalah *independent,* server tidak menyimpan sesi untuk setiap request yang ada.
RESTful API menggunakan *Uniform Resource Identifier* (URI) untuk merepresentasikan *resource*[4].

II. METODELOGI

Penelitian ini dilakukan di Kantor Career Development Centre Universitas Syiah Kuala (CDC Unsyiah), yang dimulai dari bulan Juni sampai dengan Agustus 2016. Alat dan bahan yang digunakan meliputi *hardware* dan *software*. *Hardware* yang digunakan antara lain 1 unit laptop dan 1 unit *smartphone* dengan sistem operasi Android. Sedangkan *software*pendukung yang digunakan antara lain aplikasi XAMPP (untuk *web server* lokal), IDE Eclipse (untuk membuat *web service*), aplikasi Postman (untuk menguji *web service*), dan IDE Android Studio (editor untuk membuat aplikasi berbasis Android).

A. Pembuatan Web Service

Web service dibuat menggunakan *Integrated Develepment Environment* (IDE) Eclipse yang merupakan sebuah *tool open-source*. Sebelum membuat *web service* terlebih dahulu dilakukan pencarian *dependencies* apa saja yang akan digunakan untuk mempermudah proses pembuatan *web service*. Pencarian ini dilakukan pada website https://mvnrepository.com/ yang bisa secara otomatis diunduh dari aplikasi Eclipse. *Dependencies* yang diperlukan diletakkan dalam file pom.xml.

```
<dependencies>
  <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-server</artifactId>
      <version>1.9</version>
  </dependency>
  <dependency>
          <groupId>mysql</groupId>
          <artifactId>mysql-connector-java</artifactId>
          <version>5.1.35</version>
      </dependency>
      <dependency>
      <groupId>org.glassfish</groupId>
      <artifactId>javax.json</artifactId>
      <version>1.0.2</version>
  </dependency>
```

Gambar 1. Potongan kode *pom.xml*

Setelah semua *dependencies* diletakkan di dalam file *pom.xml*, maka Eclipse akan secara otomatis mengunduh semua file *dependencies* yang diperlukan untuk membuat *web service*.

Ada beberapa komponen dari OAuth 1.0, antara lain:

Tabel 1. Komponen dari OAuth 1.0

Komponen	Penjelasan
Consumer	Aplikasi pihak ketiga yang dapat melakukan *request*
Service provider	Server
User	*Resource owner* (pihak yang memiliki hak atas *protected resources*)
Consumer Key and Secret	Consumer credentials
Request and secret token	temporary credentials
Access and secret token	Hak akses berupa token rahasia

Dalam penelitian ini yang berperan sebagai *Consumer* adalah aplikasi SMS Blast yang melakukan *request* nomor telepon alumni yang merupakan sebuah *protected resources*. Data alumni itu berada pada *data base* CDC Unsyiah *(Service Provider)*. Pihak CDC Unsyiah merupakan *User* yang memiliki hak atas *protected resoruces*.

Gambar 2. Alur proses kerja aplikasi*SMS Blast*

Dari Gambar 2 dapat dilihat sebelum *consumer* melakukan *request protected resources*, terlebih dahulu *consumer* melakukan *request temporrary credentials (request token)*.

```
@POST
@Path("/requestToken")
@Consumes(MediaType.APPLICATION_JSON)
@Produces("aplication/json")
public OauthParamResponse requestToken(OauthParamReq param
```

Gambar 3. Potongan kode web service untuk menerima request token dari consumer

Gambar 3 merupakan potongan kode untuk menerima permintaan *request token* dari *consumer*. Untuk melakukan request token, consumer harus mengirimkan informasi *Consumer Credential (Consumer Key and Secret)* beserta time stamp dan nonce. Informasi ini harus dikirimkan dalam format json. Selanjutnya web service akan mengembalikan informasi *temporrary credentials (request token)* juga dalam format json.

```
@POST
@Path("/accessToken")
@Consumes(MediaType.APPLICATION_JSON)
@Produces("application/json")
public OauthParamResponse accessToken(OauthParamAccess paramAccess)
```

Gambar 4. Potongan kode web service untuk menerima permintaan access token dari consumer

Gambar 4 merupakan potongan kode *web service* yang menerima permintaan *access token* dari *consumer*. Untuk melakukan permintaan access token consumer harus mengirimkan informasi *temporrary credentials (request token)* yang telah diverifikasi, arti telah diverifikasi adalah *consumer* telah diizinkan oleh *user* untuk mengakses *protected resources*. Sama halnya seperti pada proses *request token consumer* juga harus menyertakan *timestamp* dan *nonce*. Informasi ini dikirim dalam format json. Selanjutnya web service akan memberikan respon berupa *access* dan *secret token* juga dalam format json.

Access dan *secret token* dapat digunakan oleh *consumer* untuk mengakses *protected resources.*

B. Pembuatan Aplikasi

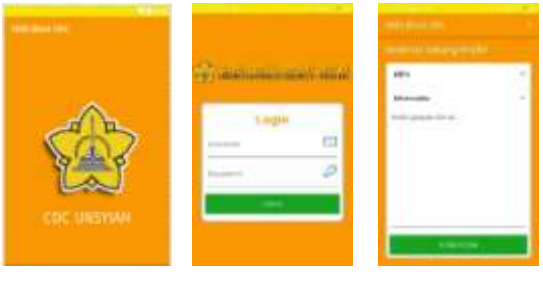
Aplikasi SMS Blast akan dibuat menggunakan IDE Android Studio. Sama halnya dengan proses pembuatan *web service*, sebelum membuat aplikasi terlebih dahulu akan dilakukan pencarian *dependencies* apa saja yang bisa memudahkan dalam proses pembuatan aplikasi SMS Blast. Dependencies ini akan diletakkan dalam file build.gradle.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:22.4.0'
    compile group: 'org.apache.commons', name: 'commons-lang3', version: '3.1'
    compile 'com.squareup.retrofit2:retrofit:2.0.2'
    compile 'com.squareup.retrofit:retrofit:1.9.0'
    compile 'com.google.code.gson:gson:2.6.2'
    compile 'com.squareup.retrofit2:converter-gson:2.0.2'
    compile group: 'joda-time', name: 'joda-time', version: '2.3'
}
```

Gambar 5. Potongan kode file build.gradle

Aplikasi SMS Blast memiliki 3 activity, yaitu Launcher Activity, Main Activity, dan Home Activity.

LauncherActivity merupakan halaman awal berupa *splash screen*. MainActivity adalah halaman untuk login operator CDC. Sedangkan HomeActivity merupakan halaman utama dari aplikasi untuk mengambil nomor telepon alumni dari *web service* dan mengirimkan SMS ke nomor-nomor tersebut.



Gambar 6. Tampilan aplikasi SMS Blast, dalam penelitian ini aplikasi SMS Blast berperan sebagai user

Pada Home Activity terdapat *drop down* untuk memilih fakultas dan program study. Ketika operator memilih fakultas, aplikasi akan proses sesuai dengan yang ditunjukkan oleh Gambar 2. Pada proses ini aplikasi akan mendapatkan *protected resources* berupa daftar nama dan kode fakultas yang ada di lingkungan Universitas Syiah Kuala. Kemudian operator akan memilih program studi, aplikasi juga melakukan proses seperti yang ditunjukkan pada Gambar 2. Pada proses ini aplikasi akan mendapatkan *protected resources* berupa nomor telepon alumni yang memiliki program studi sesuai dengan yang dipilih oleh operator.

III. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah sebuah aplikasi SMS Blast berbasis Android dan sebuah *web service* yang digunakan oleh aplikasi untuk melakukan *request* terhadap data nomor telepon alumni sesuai dengan program studi yang ada pada lingkungan Universitas Syiah Kuala.

Tampilan dari aplikasi SMS Blast dapat dilihat pada Gambar 6.

OAuth menggunakan *token* pada setiap *request*. Web service akan membangkitkan *token* yang berbeda pada setiap *request* dari *consumer*. Penggunaan *token* ini dapat meminimalkan kemungkinan terjadinya serangan *Man in the Middle Attack* dan *Hijacking Attack*[4].

Biasanya, untuk mencegah kemungkinan pencurian data selama pertukaran data, direkomendasikan untuk menggunakan TLS/SSL. Pada OAuth adanya penggunaan *signature* yang memungkinkan pertukaran data menjadi lebih aman padaprotokol berbasis non-HTTPS.

Selanjutnya, *resource provider* dapat membatasi kemungkinan serangan balik yang bersifat merusak dengan menerapkan protokol atribut *nonce* dan *timestamp*. Nilai dari *oauth_nonce* dibuat secara acak untuk menandai *request* dari *client*, dan *oauth_timestamp* mendefenisikan rentang waktu dari *nonce*.

IV. KESIMPULAN

Dari hasil pembahasan di atas dapat diambil kesimpulan:

1. Penelitian menghasilkan sebuah aplikasi SMS Blast berbasis Android dan sebuah web service yang akan melayani permintaan data oleh aplikasi SMS Blast.
2. Protokol OAuth dapat mengurangi kemungkinan serangan-serangan, seperti MITM, Hijack attacking.

3. Dengan menggunakan protokol OAuth *resource provider* dapat membatasi kemungkinan serangan balik yang bersifat merusak dengan menerapkan atribut *nonce* dan *timestamp*.

Aplikasi SMS Blast diharapkan dapat membantu CDC Unsyiah untuk menginformasikan informasi penting kepada alumni Unsyiah tanpa harus melakukan SMS ke satu-satu nomor telepon. Dengan aplikasi SMS Blast, CDC Unsyiah tidak perlu lagi memberikan daftar nomor telepon alumni kepada operator SMS. Nomor telepon alumni dapat secara langsung melakukan *request* kepada *web service* melalui aplikasi SMS Blast.

DAFTAR PUSTAKA

[1] Ableson, W. F., Sen, R., King C. dan Ortiz, C. E. 2012, *Android in Action*, Manning Publications Co.,

[2] Hammer-Lahav,E., The OAuth 1.0 Protocol. 2010, *Internet Engineering Task Force*.

[3] Chappell,D., dan Jewell, T. 2002. *Java Web Services*, OREILLY.

[4] Huang, X-W., Hsieh, C-Y., Wu, C-H.,dan Cheng, Y. C., 2015, *A Token-Based User Authentication Mechanism for Data Exchange in RESTful API. Network-Based Information Systems (NBiS)*, 2015 18th International Conference on.*IEEE*.