

RANCANG BANGUN ACTIVEX UNTUK DETEKSI PENGGUNA SISTEM MENGGUNAKAN FINGER PRINT

Banni Satria Andoko¹, Cahya Rahmad²

Manajemen Informatika, Teknik Elektro, Politeknik Negeri Malang
ando@polinema.ac.id¹, cahya_rahmad@yahoo.com²

Abstrak

Mesin pembaca sidik jari memiliki kegunaan untuk membaca identitas pengguna yang lebih baik dan akurat dikarenakan mempergunakan sidik jari sebagai identifikasinya. Dengan mempergunakan mesin tersebut, pengguna yang tidak berwenang bisa diminimalisir. Selama ini penggunaan mesin sidik jari sudah banyak dipergunakan untuk aplikasi berbasis desktop.

Penggunaannya mesin ini dalam banyak aplikasi merupakan alat ukur bahwa mesin deteksi ini sangat dibutuhkan, tetapi memiliki keterbatasan dikarenakan mesin ini hanya bisa bekerja diaplikasi lokal.

Kelemahan dari aplikasi berbasis web adalah, aplikasi ini memiliki keterbatasan untuk mengakses sumber daya client, termasuk perangkat-perangkat yang diakses oleh client. Agar aplikasi berbasis web bisa mengakses sumberdaya disisi client, maka dibutuhkan suatu aplikasi khusus untuk menjembatani komunikasi tersebut

Kata kunci : *webservice, activeX, wdsl, SOAP*.

1. Pendahuluan

Mesin pembaca sidik jari memiliki kegunaan untuk membaca identitas pengguna yang lebih baik dan akurat dikarenakan mempergunakan sidik jari sebagai identifikasinya. Dengan mempergunakan mesin tersebut, pengguna yang tidak berwenang bisa diminimalisir. Selama ini penggunaan mesin sidik jari sudah banyak dipergunakan untuk aplikasi berbasis desktop.

Penggunaannya mesin ini dalam banyak aplikasi merupakan alat ukur bahwa mesin deteksi ini sangat dibutuhkan, tetapi memiliki keterbatasan dikarenakan mesin ini hanya bisa bekerja diaplikasi lokal.

Keterbatasan tersebut bisa dijawab dengan membuat aplikasi khusus yang dipergunakan untuk berkomunikasi dengan aplikasi utama yang berbasis web. Aplikasi berbasis web memiliki keleluasaan dalam segi implementasi dikarenakan client tidak memerlukan proses instalasi dan client cukup memanggil aplikasi dari sisi server.

Kelemahan dari aplikasi berbasis web adalah, aplikasi ini memiliki keterbatasan untuk mengakses sumber daya client, termasuk perangkat-perangkat yang diakses oleh client. Agar aplikasi berbasis web bisa mengakses sumberdaya disisi client, maka dibutuhkan suatu aplikasi khusus untuk menjembatani komunikasi tersebut.

2. Landasan Teori

2.1 SOAP (*Simple Object Access Protocol*)

SOAP (*Simple Object Access Protocol*), adalah protokol komunikasi client server yang digunakan untuk mengirim dan menerima informasi "di atas HTTP". Data yang dikirim dan diterima dalam format XML. SOAP cocok digunakan untuk data kompleks yang dikirim antar client-server.

SOAP merupakan mekanisme lain yang memungkinkan penggunaan remote procedure call. SOAP bersifat netral platform, netral bahasa dan tidak bergantung pada suatu objek model. Sehingga SOAP-enabled distributed application dapat menjangkau beragam operating sistem, dimana terdiri dari objek yang berasal dari vendor yang berbeda, ditulis pada bahasa yang berbeda, dan didasarkan pada objek model yang berbeda.

SOAP dibangun dengan menggunakan protocol komunikasi HTTP. Karena HTTP didukung oleh semua browser dan server, maka SOAP dapat berkomunikasi dengan berbagai aplikasi meskipun terdapat perbedaan sistem operasi, teknologi, dan bahasa pemrogramannya, dikarenakan protocol ini memanfaatkan berbagai teknologi yang sudah ada sebelumnya dan sudah banyak digunakan. Misalnya untuk protocol transport, yang paling banyak digunakan adalah HTTP, walaupun dimungkinkan untuk menggunakan protocol transport lainnya. Sedangkan untuk format data atau message digunakan XML yang tidak diragukan lagi manfaat dan perannya di dalam pertukaran data.



Gambar 7. Arsitektur SOAP

Pesan SOAP berbentuk seperti sebuah envelope yang berisi header (optional) dan body (required). Header berisi blok informasi yang berhubungan dengan bagaimana pesan tersebut diproses. Hal ini meliputi pe-routing dan delivery setting, authentication atau authorization assertions, and transaction contexts. Body berisi pesan sebenarnya yang dikirim dan diproses. Semua yang dapat ditampilkan dengan sintaks XML dapat dimasukkan dalam pesan body.

Setiap elemen Envelope harus berisi tepat satu elemen Body. Elemen Body dapat berisi sebanyak mungkin node anak yang diperlukan. Isi dari elemen Body adalah pesan. Elemen Body ditentukan dalam suatu cara dimana dapat berisi valid dan wellformed XML yang telah dibatasi oleh suatu namespace (qualified).

Jika sebuah Envelope berisi elemen Header, harus berisi tidak lebih dari satu, dan harus tampak pada first child dari Envelope, sebelum elemen Body. Header dapat berisi valid, well-formed, dan dibatasi dengan namespace XML dimana hendak dimasukkan oleh pencipta pesan SOAP.

Setiap elemen yang berada dalam Header disebut blok header. Tujuan dari blok header adalah untuk memberitahukan informasi yang berhubungan dengan pemrosesan pesan SOAP

2.2 WSDL (Web Services Description Language)

WSDL adalah bahasa berbasis XML untuk mendeskripsikan Web services dan bagaimana untuk mengaksesnya. WSDL membantu konsumen web service untuk memakai/menggunakan layanan web service. Tanpa WSDL web service akan terkunci dan tidak berguna. WSDL menspesifikasikan lokasi service dan operasi (methods) yang disediakan oleh web service

Sebagai protokol komunikasi dan format pesan yang distandarkan pada komunitas web, WSDL menjadi lebih tepat dan penting untuk mendeskripsikan model komunikasi di beberapa langkah yang terstruktur (dalam memprogram web

service). WSDL menyediakan kebutuhan ini dengan mendefinisikan bentuk XML untuk mendeskripsikan layanan jaringan sebagai kumpulan-kumpulan titik akhir (endpoints) komunikasi yang mampu melakukan pertukaran pesan. Hal tersebut sangat membantu komunikasi antara dua teknologi pada platform-platform secara umum. Definisi layanan WSDL menyediakan dokumentasi untuk sistem terdistribusi dan berfungsi sebagai panduan untuk mengotomatisasi rincian yang terlibat dalam komunikasi antaraplikasi

WSDL sering digunakan pada kombinasi SOAP dan XML Schema untuk menyediakan Web service di internet. Aplikasi client yang menghubungkan ke sebuah Web service dapat membaca file WSDL untuk menentukan operasi apa saja yang tersedia pada server. Tipe-tipe data khusus yang digunakan di-embed pada file WSDL dalam bentuk XML Schema. Client kemudian dapat menggunakan SOAP untuk memanggil operasi-operasi yang terdaftar pada file WSDL secara aktual menggunakan XML atau HTTP.

2.3 ActiveX

ActiveX merupakan teknologi yang dibuat oleh Microsoft office yang memungkinkan aplikasi software berbeda untuk berbagi informasi dan fungsi. ActiveX hanya bekerja pada aplikasi Microsoft seperti Microsoft excel, Microsoft office, Microsoft Power Point dan internet explorer. ActiveX juga hanya berjalan pada operasi sitem windows.

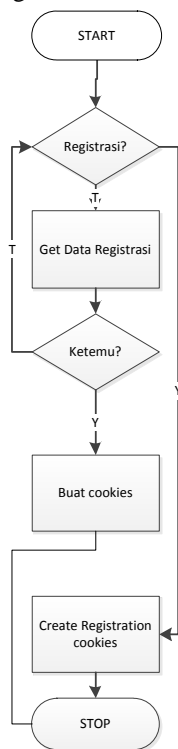
Activex atu sering disebut dengan kontrol activex merupakan kontrol menggunakan Microsoft ActiveX teknologi. Kontrol ActiveX dapat secara otomatis didownload dan dieksekusi oleh Web browser. ActiveX, bukan bahasa pemrograman melainkan seperangkat aturan untuk bagaimana aplikasi harus berbagi informasi. Pemrogram dapat mengembangkan kontrol ActiveX dalam berbagai bahasa, termasuk C, C ++, VisualBasic, dan Java.

Kontrol ActiveX mirip dengan Java applet. Tidak seperti applet Java, kontrol ActiveX memiliki akses penuh ke sistem operasi Windows. Ini memberi mereka kekuatan lebih dari applet Java, tetapi dari kekuatan ini timbul resiko baru tertentu yakni applet dapat merusak perangkat lunak atau data pada mesin. Untuk mengendalikan risiko ini, Microsoft mengembangkan sistem pendaftaran sehingga browser dapat mengidentifikasi dan mengotentikasi kontrol ActiveX sebelum didownload. Perbedaan lain antara applet Java dan kontrol ActiveX adalah Java applet dapat ditulis untuk berjalan pada semua platform, sedangkan kontrol ActiveX saat ini terbatas pada lingkungan Windows.

Selain itu ActiveX merupakan bahasa scripting yang disebut VB Script yang memungkinkan penulis Web untuk menanamkan elemen interaktif dalam dokumen HTML

3. Desain Sistem Aplikasi activeX

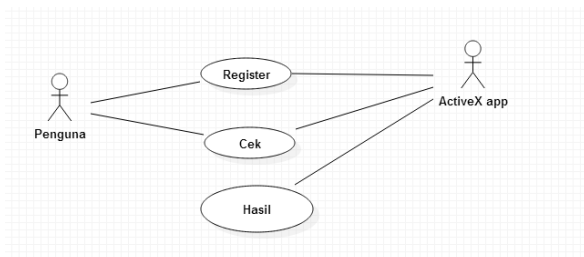
Aplikasi activeX memiliki perbedaan dengan aplikasi desktop, aplikasi ini dapat dipanggil dengan menggunakan command windows dan seolah-olah berjalan dibelakang sistem. Tugas dari aplikasi ini adalah untuk menangkap event-event yang diberikan oleh web aplikasi dari sisi server. Setelah menangkap event-event tersebut, selanjutnya aplikasi ini mengirimkan hasil validasi ke server.



Gambar 8. flowchart activeX

4. Desain Sistem Aplikasi web

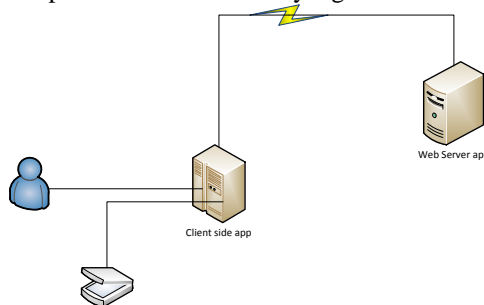
Aplikasi ini nantinya akan berkomunikasi dengan aplikasi activeX disisi client. Aplikasi client dipergunakan untuk menangkap event dari peralatan scanner sidik jari. Nilai dari gambar tersebut diproses dan dijadikan suatu nilai unik yang nantinya akan dipergunakan sebagai verifikator data sidik jari yang berada di sisi server.



Gambar 3. Usecase Diagram Sistem

5. Desain arsitektur sistem

Dengan mempergunakan arsitektur ini, maka untuk bisa mendapatkan sumberdaya lokal client dibutuhkan suatu aplikasi khusus sebagai jembatan dan pintu masuk ke sistem yang berada disisi server.



Gambar 3. Diagram Sistem

6. Kesimpulan dan Saran

6.1 Kesimpulan

Dengan mengembangkan sebuah aplikasi activeX maka :

- Aplikasi berbasis web bisa seolah-olah berjalan layaknya sebuah aplikasi.
- Memiliki tingkat keamanan lebih tinggi dikarenakan mempergunakan sidik biometri sebagai identifikasi pengguna.
- Komunikasi data melewati aplikasi menjamin keamanan data.

6.2 Saran

Untuk mengembangkan aplikasi ini maka disarankan untuk:

- Membuat aplikasi berbasis java, sehingga memiliki keleluasaan untuk penerapan pada banyak Sistem Operasai
- Pengenalan pola bisa diterapkan dengan konsep AJAX sehingga tidak perlu melakukan refresh untuk seluruh halaman aktif
- Mengembangkan konsep baru sehingga tidak diperluakan aplikasi tambahan dan kebergantungan pada komponen tertentu

Daftar Pustaka:

Christensen, E. F. (2001). Web Service Definition Language(WSDL). <http://www.w3.org/TR/wsdl>.

Graham, S. (2002). Building Web Services with Java: Making Sense. Sams Publishing.

Yates R dan Neto B. (1999) : *Modern Information Retrieval*. Addison Wesley Longman Limited, 1999.