Membangun Webservice dengan Simple Object Acces Protocol (SOAP)

Jazi Eko Istiyanto,[1] Siti Imsyawati Maulidya[2]
jazi@ugm.ac.id
Departement of Physics Faculty Mathematics and Natural Science, Gadjah Mada University Yogyakarta 55281
lia@akprind.ac.id
Departement of Informatics, Institut of Sains & Technology AKPRIND Yogyakarta

Summary

The web utilization in industries at this time has become a powerful transaction tool. As a distributed system, the web interactively offers more advantages to attract the consumer to buy their products. Basically, the foundation of web service is a SOAP (Simple Object Access Protocol) protocol and XML (eXtensible Markup Language). The web service it self are using HTTP (Hyper-Text Transfer Protocol) and SOAP to produce data exists in web. One time when user doing a data request via internet (it is not limited to web application, it is also support windows based or mobile application connected to the internet as a major condition), SOAP will call a function resident in a server and return a readable XML format to the user.

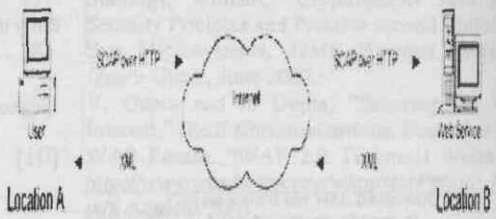
Keywords
Web Service, SOAP (Simple Object Access Protocol), RPC, XML, HTTP.

1. Pendahuluan

We service adalah antar muka dari suatu fungsi aplikasi yang dapat diakses melalui jaringan, dimana dibangun dengan standard teknologi internet. Dengan kata lain, bila suatu aplikasi dapat diakses melalui jaringan dengan menggunakan kombinasi protokol seperti HTTP, XML, SMTP dan SOAP, maka itulah layanan web (*web service*).

1.1. Arsitektur WEB Service

Arsitektur WEB Service merupakan arsitektur sistem yang lahir dari pendekatan ini. Arsitektur ini merupakan evolusi dari analisis dan desain berorientasi obyek (OOA & OOD). Seperti diketahui, dalam orientasi obyek dikenal terminologi seperti: encapsulation, message passing, dynamic binding, service description and querying. Dasar dari orientasi webservice adalah *notion* bahwa *everything is service*.



Gambar 1. Arsitektur Web Service

2. Simple Object Acces Protocol (SOAP)

SOAP merupakan singkatan dari Simple Object Access Protocol. Dimana maksud dari protokol adalah sekumpulan aturan yang mengatur komunikasi antar komputer. Ide dasarnya adalah dua aplikasi, dengan tidak memperdulikan operating system, bahasa pemrograman, atau detil implementasi teknis yang lain, dapat menggunakan informasi secara bersama-sama. SOAP adalah XML, dengan kata lain, SOAP adalah aplikasi dari spesifikasi XML, dapat menggunakan standard XML seperti XML Schema dan XML Namespaces untuk pendefinisian dan fungsi/penggunaan. SOAP dengan berbasis pada XML serta protokol transport HTTP, dimana telah digunakan berbagai platform sebagai standard web, memungkinkan dapat diterapkan dan digunakan pada berbagai platform dan bahasa pemrograman. SOAP memiliki kompatibilitas, fleksibilitas, dan interoperabilitas (kemampuan dua tipe komputer berbeda untuk saling bekerja sama dan berkomunikasi) yang tinggi. File-file web service mempunyai ekstension .asmx yang mempunyai sifat berbeda dengan halaman web biasa. Aplikasi web service tidak mempunyai sebuah "*User Interface*" atau tampilan web pada umumnya. Web service hanya berupa class dan method dari sebuah fungsi dan mempunyai output dalam format XML. Hal ini penting untuk diingat, bahwa "konsumen" dari web services adalah bukan "manusia" melainkain

"aplikasi lain" sehingga tampilan web dianggap tidak terlalu penting.

3. Membangun Web Service

Membangun sebuah aplikasi web Service sederhana. Banyak tools/editor HTML yang bisa digunakan untuk membuat aplikasi web Service termasuk notepad. Bila membangun web service sederhana dapat menggunakan notepad sebagai . Tetapi web Service yang lebih kompleks, dianjurkan menggunakan tools Visual Studi .NET.Membuat Web Service menggunakan Visual Studio. NET, berarti membangun web service dalam Microsoft NET. Visual Sudio.NET adalah development tool yang dapat dipakai untuk membuat webservices, dan aplikasi yang memanfaatkan webservices. Aplikasi ini dapat berbentuk salah satu di bawah ini:

1. WEB Form Application
2. Win Form Application
3. Console Application

WEBServices pada hakikatnya adalah classes dengan methods yang dapat dipanggil secara remote. Mendefinisikan sebuah WEBService mirip dengan mendefinisikan sebuah class. Suatu WEBService dapat diinvoke menggunakan Microsoft.NET

WEB Services dapat dibuat menggunakan salah satu dari bahasa-bahasa pemrograman seperti visual basic, visual C# (baca C sharp), dan extended visual C++, yang menggunakan common runtime language yang sama. Source code dalam salah satu bahasa tersebut dikompilasi ke kode dalam bahasa intermediari, disebut Microsoft Intermediary Language (MSIL), Kode tersebut belum menjadi "native" dari mesin ekesekutor. Konsep just in time dikenal dalam Microsof.NET, yaitu bahwa kode intermediari hasil kompilasi dieksekusi oleh mesin menggunakan just in time execution. Terdapatnya CRL memungkinkan konsep inheritansi klas/ obyek dilakukan secara cross language.

3.1. Membuat Web Services

Persiapan untuk mencoba membuat web services ini

1. Install web server , dan pastikan Web Server local anda jalan (IIS).
2. Install Microsoft Visual Studio .NET beserta NET FrameWorknya.

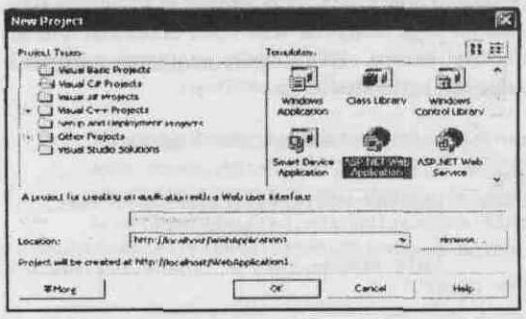
Berikut langkah – langkah dalam membuat Web Services sederhana dan konsep parameters dan data yang diperlukan untuk membangun web services ini.

3.2. Sistem Layanan Web Validasi Kartu Kredit.

Disini kita akan membuat web service system validasi kartu kredit. Logika pengesahan kartu kredit adalah sebagai sumber yang dapat dipakai bersama diantara berbagai aplikasi. Logika ini perlu memiliki infrastruktur dan tingkat persyaratan yang berbeda-beda diantara aplikasi komersial, jadi dalam hal ini dibutuhkan satu aplikasi terpisah yaitu aplikasi layanan web Visual Studio.NET yang akan dikerjakan pada langkah berikut ini.

Langkah 1. Layanan *WebForm*

Membuka Visual Studio.NET. Selanjutnya membuat proyek *Web Application* baru. Mengganti tipe proyek dengan dengan pilihan bahasa, lalu memilih template *Web Application*. Proyek diberi nama commerce , lokasi URL diarahkan menuju ke *server Web Target*: http//localhost/Commerce.



Gambar 2. Web Application

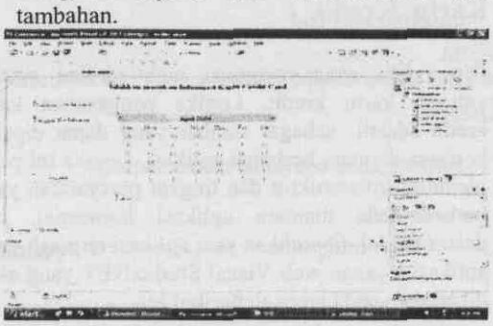
Langkah – 2

Setelah itu maka pada visual studio.NET akan membuat beberapa folder penting dan file-file pendukung web services. Pada Gambar 3. dibawah dapat dilihat Explorer dari project web services tersebut

Visual Studio.NET membuat proyek pada server Web yang ditentukan dan meletakkan template file aplikasi web kedalam direktori tersebut. Pada file ini terdapat satu halaman *WebForm1.aspx*, namanya diganti dengan Order.aspx. Lalu ganti nama kelas yang didefinisikan dalam Order.aspx dari WebForm1 menjadi Order. Langkah ini terdiri dari :

- Membuka file *code-behind Order*.aspx dengan mengklik-kanan nama file dalam *Solution Explorer* kemudian pilih view-code.

- Mengganti nama kelas *WebForm1* beserta konstruktornya dengan nama Order.
- Pada panel Properties, ganti nama labelnya dengan status, teks untuk label tersebut; *Silahkan masukkan informasi kartu kredit*, nama tombolnya dengan *Place Order* setelah mengatur layout dan memasukkan teks tambahan.



Gambar 3. Tampilan panel properti

Langkah 3.

Menyiapkan pelaksanaan untuk tombol *Place Order*. Untuk menambahkan *even-handler* tombol pesanan itu, klik dua kali tombol itu dan masukkan kode (terdapat pada lampiran) , supaya saat pengguna mengklik tombol tersebut, panel *validate* akan dipanggil untuk menentukan apakah informasi kartu kredit yang dimasukkan adalah sah. Kode ini juga mengatur teks label untukenampilkan informasi berupa teks kepada pengguna mengenai keabsahan kartu kreditnya.

```
private void placeOrder_Click(object
sender,System.EventArgs  e)
 {
if(cc.Validate(this.creditCardNumber.Text,
 this.expirationDate.SelectedDate))
        {
        this.status.Text = "Thank you for
your order.";
        }
        else
                {
        this.status.Text = "Credit card
invalid.";
 }
```

Langkah 4.

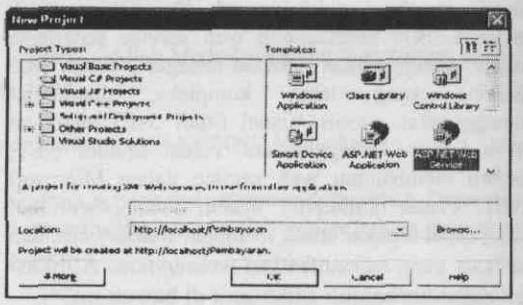
Membuat metode validate dibawah metode *PlaceOrder_Click*. Metode ini untuk menentukan apakan tanggal kadaluwarsanya sudah terlewati, jika belum, metode validate akan menggunakan algoritma mod10 standar untuk melakukan pengesahan kartu kredit.

Langkah 5.

Mengkompilasikan dan mencoba aplikasi web untuk menguji apakah sudah dapat dijalankan.

Langkah 6. Membuat Layanan *Web Pembayaran*

- Membuka Visual Studio.NET, lalu pilih *Create New Project*, ganti tipe proyeknya, kemudian pilih template web service. Proyek ini diberi nama *Pembayaran*, selanjutnya mengganti lokasi URL agar menuju ke server Web target : http//Localhost/Pembayaran.



Gambar 4. Membuat *new project* Pembayaran.

- Mengganti nama *file Service1.asmx* menjadi *CreditCard.asmx*. Sebaiknya nama kelas disesuaikan dengan nama file .asmx, ganti nama kelas *Service1* menjadi *CreditCard*. Visual Studio.NET akan membuat file *code-behind* untuk file .asmx sama dengan yang dibuat untuk file .aspx.
- Memotong metode *validate* dari *Web Form Order* dan meletakkannya kedalam kelas *CreditCard*. Untuk memperindah tampilan kelas tersebut, dapat menggunakan WebMethod lalu dikompilasikan pada aplikasi ini. Atribut *WebMethod* adalah satu-satunya kode yang dibutuhkan untuk memunculkan metode *validate* pada web.

```
[WebMethod]
public bool Validate(string cardNumber,
DateTime expDate)
{
if(expDate >= DateTime.Today)
   {
        int total = 0;
        int temp = 0;
        char [] ccDigits =
cardNumber.ToCharArray();
        for(int i = 0; i < cardNumber.Length;
i++)
   {
        if(((i+1)%2) == 0)
                {
total += int.Parse(ccDigits[i].ToString());
                }
                else
                {
temp = int.Parse(ccDigits[i].ToString())* 2;
        if(temp > 9)
                {
                temp = (int)temp - 9;
                }
        total += temp;
                }
                }
```

publik bagi web service menggunakan format XML. Interface publik ini bisa mengandung informasi pada seluruh function, informasi tipe data, informasi binding, informasi alamat dari suatu service. WSDL juga memiliki built-in extention untuk menyajikan service berbasis SOAP [11],[7].

3.4 UDDI

Universal Description, Discovery and Integration (UDDI) menyajikan komponen *service discovery* di dalam arsitektur web service. UDDI mengandung dua komponen utama yaitu , **pertama**, UDDI adalah *spesifikasi teknis untuk membangun direktori* bisnis terdistribusi dan web service. Data disimpan dalam format XML, dan spesifiaksi UDDI mengandung API untuk pencarian data dan publikasi data baru. **Kedua**, UDDI Business Registry adalah implemetasi penuh dari spesifikasi UDDI. Dengan UDDI dimungkinkan sembarang perusahaan mendaftarkan diri beserta layanan-layanannya [7].

4. Aspek Keamanan Web Service

Pertukaran informasi antar perusahaan melalui internet mutlak dibutuhkan tetapi akan membawa implikasi keamanan. Isu keamanan di internet cukup penting karena merupakan infrastruktur jaringan publik yang *insecure* dan *untrustable*, mudah diserang oleh para intruder baik yang amatir maupun yang professional.

Informasi-informasi yang bisa diakses melalui internet memiliki tingkat kerahasiaan yang beragam. Contohnya, dalam bisnis perhotelan, *informasi* penyewaan kamar bisa diakses oleh setiap orang, tetapi ada layanan pemesanan khusus dengan harga dan pembayaran khusus bagi rekanan bisnis, misalnya biro wisata yang memiliki hubungan bisnis *dengan hotel*.

Web service bekerja menggunakan teknologi XML dan SOAP. Dengan kedua teknologi tersebut berarti web service memiliki karakteristik **netral bahasa** dan **netral platform,** tetapi sekaligus memiliki lubang keamanan yang besar [8], [3], [12].

Siddique dalam [13] mengatakan bahwa standar keamanan berbasis XML yang dipublikasikan oleh W3C dan OASIS adalah **XML Signatures, XML Encryption dan Web Services Security (WSS).** Ketiga standard keamanan berbasis XML tersebut, dalam komunikasi SOAP menawarkan fitur keamaman berupa :

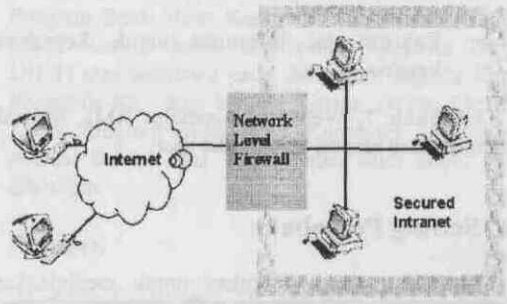
- otentikasi user
- integritas message dan
- kerahasiaan

Salz dalam [14] menuliskan bahwa beberapa kriteria keamanan web service adalah : ,

- **Integritas**, message diterima dalam kondisi asli tidak mengalami perubahan.
- **Rahasia**, message diterima *hanya oleh pihak* yang sah.
- **Otentikasi**, pengirim dapat diketahui secara pasti
- **Otorisasi**, pengirim adalah pemakai yang resmi, *memiliki hak pakai terhadap service tersebut.*

Pengamanan jaringan sering menggunakan firewall. Firewall tradisional bekerja dengan melakukan bloking terhadap port tertentu, nomor IP tertentu atau menutup jalur berdasarkan sifat penggunaannya. Firewall ini, packet-level firewall [3] / network-level firewall [14], tidak mampu membedakan isi message SOAP dan XML yang lewat, sehingga message dapat lewat begitu saja [1]. Network-level firewall digambarkan seperti gambar 2.

Mekanisma yang mungkin untuk melakukan pengamanan web service adalah menggunakan XML Firewall, yaitu modul keamanan yang merupakan gabungan protokol-protokol keamanan yang direkomendasikan oleh W3C dan OASIS[13].



Gambar 2. Network Level Firewall

XML firewall bekerja pada content-level yaitu dengan melakukan pemeriksaan terhadap message yang diterima. XML firewall mampu memeriksa SOAP header dan XML tags, sehingga mampu membedakan message yang *unauthorized* milik pengguna yang tidak sah. [3], [1].

5. Cara Kerja XML Firewall

Bilal Siddiqui, dalam [15] menjelaskan bahwa cara kerja XML Firewall dapat dijelaskan berdasarkan gambar 3.

Langkah 1, User mengirimkan request terhadap suatu service.

Langkah 2, XML firewall menerima request berisi pemanggilan suatu method. Sebelum request diteruskan ke SOAP Server, XML firewall memeriksa dua hal :

hanya berupa `class` dan `method` dari sebuah fungsi dan mempunyai output dalam format XML. Hal ini penting untuk diingat, bahwa "konsumen" dari web services adalah **bukan** "manusia" melainkan "aplikasi lain" sehingga tampilan web dianggap tidak terlalu penting.

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa :

1. Keuntungan dari penggunaan SOAP didalam HTTP yaitu mampu mengimplentasikan fleksibilitas SOAP dengan semua fitur atau kelengkapan HTTP. Penggunaan SOAP tidak berarti bahwa SOAP mengesampingkan HTTP tetapi SOAP mengikuti kegunaan dari HTTP itu sendiri.
2. SOAP secara alami mengikuti model permintaan/ respon dari HTTP, SOAP menyediakan parameter permintaan/ request dan juga parameter respon didalam respon HTTP. Tetapi perantara SOAP tidak sama dengan perantara HTTP. Hal itu adalah suatu perantara HTTP yang dialamatkan dengan koneksi HTTP tidak bisa memproses SOAP yang dibawa oleh request HTTP.

References

[1] Simple Object Access Protocol (SOAP) 1.1 W3C Note 08 May 2000.

[2] Scott Short, 2003, *Building XML Web Service For The Microsoft Net Platform*, first Edition, Microsoft Press, Microsoft Corporation.

[3] The Open Group, *1997, Remote Procedure Call Mode CDE 1.*1: Remote Procedure Call Copyright ©.

[4]http://www.w3.org/TR/2000/NOTE-SOAP-20000508, 2003

[5]http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[6]http://www.w3.org/TR/SO

Jazi Eko Istiyanto received a B.Sc in Physics from Gadjah Mada University (1986) , a postgraduate Diploma in Computer Programming and Microprocessor Applications (1997), an M.Sc in Computer Science (1998), and Ph.D in Electronic Systems Engineering (1995), all from Essex University, UK. He is now head of a BSc. Scheme in Electronics and instrumentations at The Physic Department and a faculty member of the M.Sc program in Computer Science, Gadjah Mada University. His research interest covers information security, web service, and embedded systems.

Siti Imsyawati Maulidya received the B.S. degrees in Information Technology from Institute of Science and Technology "AKPRIND" Yogyakarta in 1996. Currently, she is puswing an MSc in Computer Science, Gadjah Mada University, She research interest covers to Computer Network, Distribute System, Internet and Image Processing.