

DIKTAT MATA KULIAH XML

KONSEP DAN IMPLEMENTASI

XML WEB SERVICE

SOETAM RIZKY WICAKSONO

DAFTAR ISI

| | |
|---|-----------|
| KONSEP XML WEB SERVICE | 1 |
| APAKAH XML WEB SERVICE ITU ? | 2 |
| DASAR XML | 11 |
| APAKAH XML ITU ? | 12 |
| SEKILAS TENTANG FORMAT XML | 13 |
| XML DALAM ASP .NET 2.0 | 14 |
| KONSUMSI XML WEB SERVICE..... | 15 |
| XML WEB SERVICE DARI INTERNET | 16 |
| KONSUMSI DALAM APLIKASI WEB..... | 17 |
| KONSUMSI DALAM APLIKASI DESKTOP | 18 |
| IMPLEMENTASI XML WEB SERVICE | 19 |
| TIPE DATA PRIMITIF | 20 |
| <i>Integer</i> | 20 |
| <i>String</i> | 28 |
| <i>Array</i> | 32 |
| PUBLIKASI DAN UPLOAD WEB SERVER LOKAL..... | 35 |
| KONSUMSI XML WEB SERVICE | 43 |
| TIPE DATA KOMPLEKS | 48 |
| UPLOAD DAN PUBLIKASI ONLINE | 49 |
| XML WEB SERVICE DAN DATABASE | 50 |
| SEKILAS ADO .NET 2.0 | 51 |
| CONNECTED APPLICATION | 52 |
| <i>Datareader</i> | 52 |
| <i>Command</i> | 52 |
| DISCONNECTED APPLICATION..... | 53 |
| <i>Dataset</i> | 53 |
| XML WEB SERVICE DAN MICROSOFT OFFICE | 54 |
| APA YANG DIBUTUHKAN ? | 55 |
| XML WEB SERVICE DALAM WORD | 59 |

| | |
|-----------------------------------|----|
| XML WEB SERVICE DALAM EXCEL | 60 |
|-----------------------------------|----|

XML Web Service

Konsep dan Implementasi dengan ASP .NET 2.0 dan Visual Basic
2005

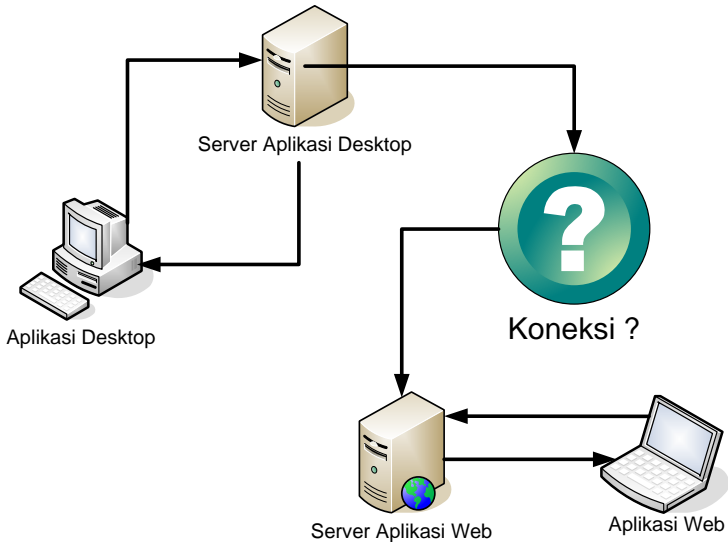
Konsep XML Web Service

Apakah XML Web Service itu ?

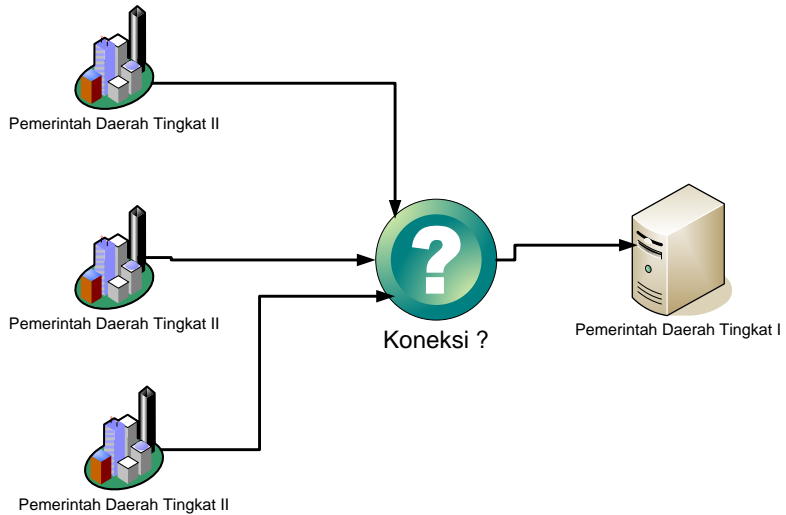
Sudah bukan menjadi rahasia lagi bahwa integrasi aplikasi sistem informasi di dalam sebuah perusahaan merupakan kendala klasik yang masih jarang terselesaikan saat ini. Begitu pula dengan transfer data lintas database, ataupun lintas bahasa pemrograman.

Misalkan saja, perusahaan X yang telah menjalankan sistem informasi berbasis desktop yang telah terbukti berjalan bertahun-tahun, kini ingin membuat situs untuk dapat mempublish datanya untuk kepentingan para customer. Karena bahasa pemrograman yang digunakan berbeda, dan juga karena transfer data dari aplikasi desktop ke aplikasi basis web dianggap tidak mungkin, maka yang terjadi adalah proses input ulang dari database di aplikasi desktop ke aplikasi web tiap hari !

Tentu saja, proses “primitif” tersebut sangatlah “melelahkan” serta inefisien, meski tujuan utama dari proses transfer data dapat dilakukan, tetapi proses tersebut sangatlah tidak membantu pihak perusahaan untuk dapat melakukan publikasi datanya secara up to date dengan sangat mudah dan cepat.



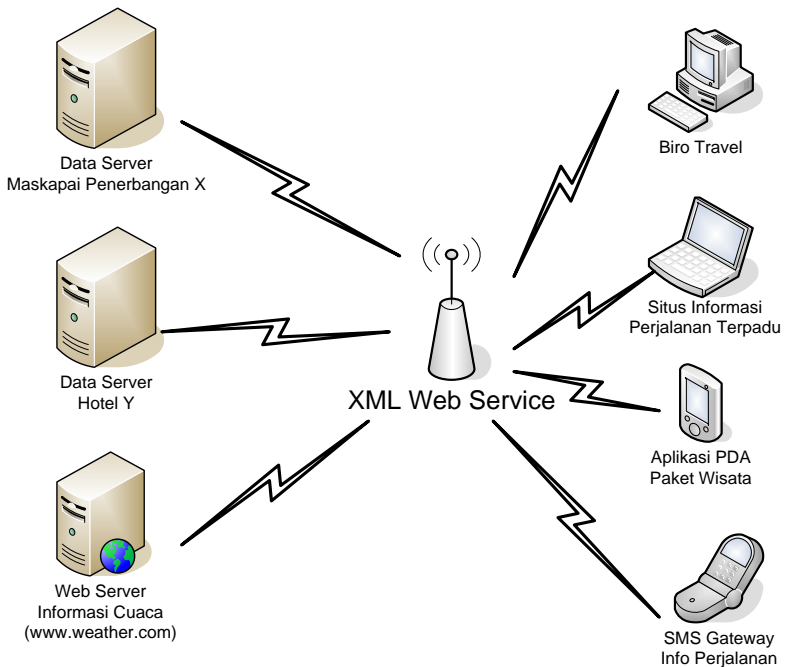
Begitu pula saat terjadi laporan konsolidasi yang membutuhkan transfer data dari sebuah sistem informasi kependudukan di beberapa pemerintahan daerah tingkat II (yang masing-masing menggunakan bahasa pemrograman serta database yang berbeda) ke dalam sebuah sistem informasi kependudukan di pemerintah daerah tingkat I, maka yang terjadi adalah melakukan transfer data terlebih dulu ke dalam Microsoft Excel, yang kemudian digabungkan satu per satu ke dalam sebuah workbook. Sungguh melelahkan, dan memungkinkan terjadinya kesalahan dengan probabilitas yang sangat besar !



Secara umum, masalah-masalah seperti pada ilustrasi tersebut dapat diatasi dengan cara yang jauh lebih mudah jika menggunakan XML Web Service. Transfer data dari aplikasi desktop ke dalam aplikasi web (bahkan juga ke aplikasi basis lain, misalkan berbasis PDA ataupun lintas platform), ataupun transfer data dari sistem informasi dengan bahasa pemrograman dan database yang berbeda ke dalam sistem informasi yang lain, seluruhnya dapat diselesaikan dengan menggunakan solusi XML Web Service.

Solusi lain yang ditawarkan misalnya, melakukan publish data dari sebuah situs (tanpa peduli bahasa pemrograman yang digunakan) agar dapat digunakan ke dalam situs yang lain, misalnya publish informasi cuaca, informasi jadwal penerbangan dan lainnya. Bahkan XML

Web Service mampu mengintegrasikan berbagai XML Web Service ke dalam sebuah aplikasi, dan sang programmer dari aplikasi tersebut tidak pernah perlu mengetahui apakah data yang diambil berasal dari database vendor tertentu dan dibuat dengan menggunakan bahasa pemrograman tertentu.



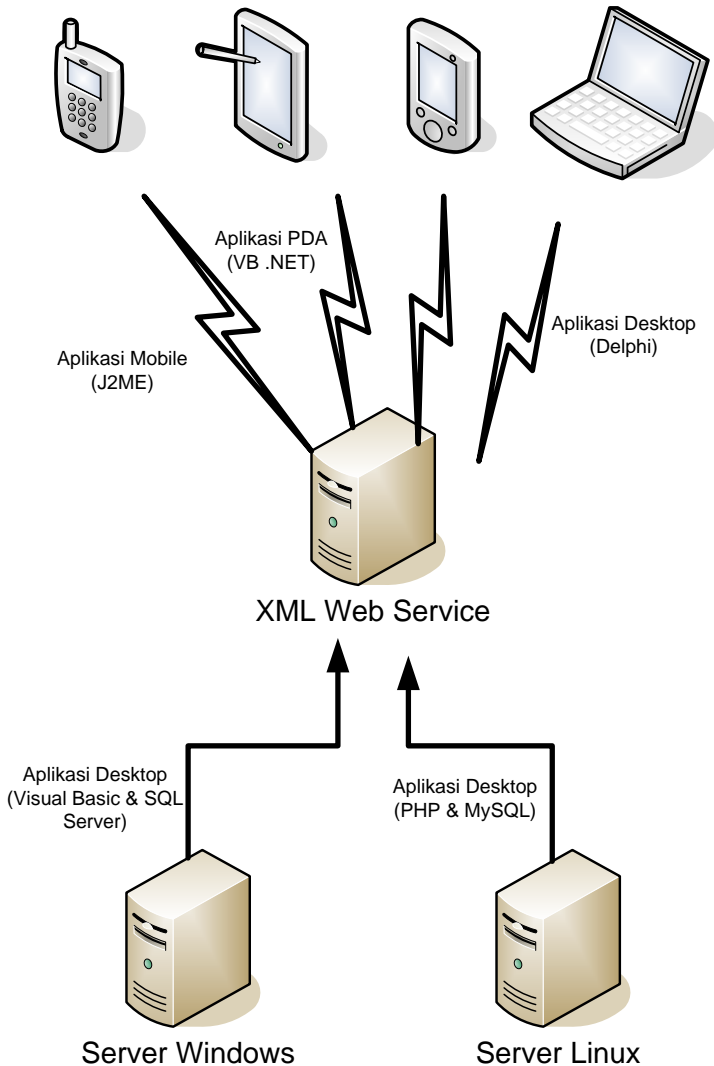
Dalam ilustrasi pada gambar terakhir, diandaikan jika sebuah biro travel ingin mendapatkan informasi terpadu dari maskapai penerbangan mengenai tiket yang tersedia pada kota X, sekaligus data dari server hotel di kota yang sama, sekaligus mengambil informasi cuaca dari situs www.weather.com, maka yang terjadi adalah

sebuah informasi terintegrasi bagi biro travel tersebut, sehingga mampu membuat sebuah penawaran paket wisata dalam waktu sekejap kepada pelanggannya yang ingin bepergian ke kota X, dan pelanggan akan merasa sangat “istimewa” !

Bahkan, aplikasi yang dibuat oleh biro travel tersebut nantinya juga dapat dipublikasi sebagai sebuah situs informasi perjalanan terpadu, dan dengan XML Web Service juga dapat menjadi sebuah aplikasi yang dapat diakses melalui PDA sebagai sebuah informasi paket wisata terpadu, dan juga mampu menjadi sebuah informasi pada aplikasi SMS Gateway untuk mengetahui informasi perjalanan secara cepat dan akurat bagi para pelanggannya. Maka, bisa dibayangkan *value added* yang dapat ditawarkan oleh biro travel tersebut dibandingkan para pesaingnya !

Dengan kata lain, fungsi utama dari XML Web Service jika disebutkan dalam satu kata adalah *integrasi* !! Jadi, apa sebenarnya XML Web Service itu ?

XML Web Service sendiri memiliki berbagai macam definisi jika dilihat dari literatur acuan yang sudah beredar. Secara umum, XML Web Service adalah fungsi logika yang dapat diakses dengan menggunakan protokol standard yaitu HTTP.



Salah satu hal yang perlu diperhatikan lagi bahwa XML Web Service tidak hanya dapat diimplementasikan di dalam aplikasi online (internet), tetapi juga dapat diimplementasikan di dalam aplikasi lokal (intranet).

Sebagai contoh, sebuah perusahaan retail yang ingin mengintegrasikan aplikasi POS (Point Of Sales) yang saat ini sedang berjalan, dan (umumnya) menggunakan bahasa pemrograman “jaman batu” yaitu Foxpro, ke dalam aplikasi back office yang telah menggunakan ERP dengan database Oracle. Maka dengan bantuan XML Web Service, kedua aplikasi tersebut dapat diintegrasikan datanya menjadi “satu”.

Jadi, sekali lagi, apakah XML Web Service hanya sekedar sebuah fungsi yang dapat mengintegrasikan berbagai sistem dari bermacam bahasa pemrograman serta database yang berbeda ? Jawabannya (sekali lagi) ya ! Dan perlu diingat pula, beberapa poin penting dalam XML Web Service, yaitu :

1. Standar dari XML Web Service adalah standar industri yang independen
2. XML Web Service tidak menyediakan interface (antar muka), tetapi hanya menyediakan fungsi
3. XML Web Service menggunakan HTTP sebagai protokol standar
4. XML Web Service menggunakan model *request-response* yang sama seperti pada aplikasi web
5. XML Web Service ditempatkan di dalam sebuah web server
6. XML Web Service lebih ditujukan untuk kepentingan dari programmer ke programmer, sehingga pengguna

XML Web Service yang utama bukanlah *end user* tetapi programmer yang menggunakan bahasa pemrograman/database/platform yang berbeda.

7. Hasil dari XML Web Service merupakan *plain text* yang berupa file XML
8. Fungsi dari sebuah XML Web Service bukanlah semacam sulap yang dapat langsung “muncul” dalam sebuah interface aplikasi, tapi perlu sebuah pengolahan lebih lanjut.
9. Secara umum, hampir semua bahasa pemrograman yang mampu melakukan koneksi ke internet dan membaca file XML, akan mampu pula “mengkonsumsi” XML Web Service, tapi tidak semua bahasa pemrograman tersebut mampu “memproduksi” XML Web Service.
10. Dalam perkembangannya, beberapa hasil implementasi dari XML Web Service tidak lagi dapat dikonsumsi secara bebas oleh banyak bahasa pemrograman, bergantung kepada model dan jenis nilai balik yang ada dalam XML Web Service tersebut misal : hasil XML Serialization, XML Schema dan lainnya.

Dasar XML

Apakah XML itu ?

Sekilas tentang format XML

XML dalam ASP .NET 2.0

Konsumsi XML Web Service

XML Web Service dari internet

Konsumsi dalam aplikasi web

Konsumsi dalam aplikasi desktop

Implementasi XML Web Service

Tipe data primitif

Integer

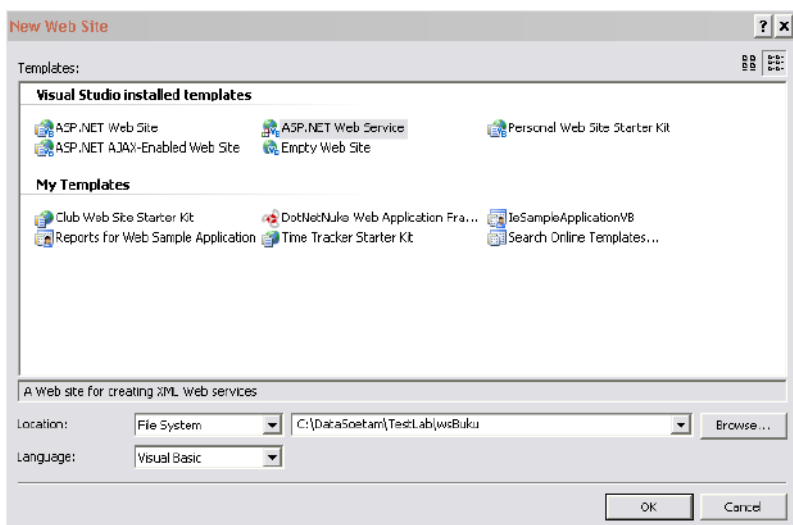
XML Web Service pertama yang akan kita buat adalah XML Web Service yang mengembalikan nilai balik primitif atau sederhana yaitu nilai balik tipe integer. Untuk memproduksi atau membuat sebuah XML Web Service dalam lingkup .NET Framework 2.0 dapat menggunakan Visual Studio 2005 (dengan lisensi berbayar) atau dengan menggunakan Visual Web Developer Express Edition yang gratis.

Perlu diingat ulang bahwa XML Web Service tidak menyediakan interface (antar muka) dari sebuah aplikasi, tetapi hanya sekedar menyediakan fungsi global yang (seharusnya) mampu dieksekusi di lintas bahasa pemrograman dan lintas sistem operasi. Karenanya, dalam proses pembuatan XML Web Service perlu diperhatikan beberapa jenis tipe nilai balik yang harus dipergunakan, dan perlu juga dirancang dengan sangat baik bagaimana nilai balik tersebut nantinya akan dipergunakan dalam proses selanjutnya.

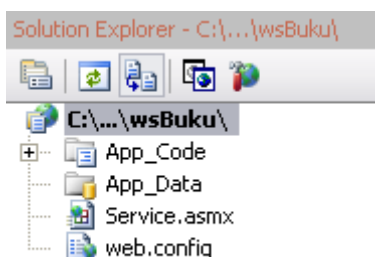
Berikut adalah langkah pembuatan dari XML Web Service dengan menggunakan Visual Web Developer Express Edition :

1. Buka Visual Web Developer Express, kemudian pilih template *ASP .NET Web Service*. Pilih *Location* → *File*

System dan beri nama solution *wsBuku* pada folder yang Anda tentukan sendiri



- Berikutnya, di dalam window *Solution Explorer*, akan terlihat struktur solution sebagai berikut :



Dalam solution explorer tersebut, terdapat folder *App_Code* yang nantinya akan berisi listing dari XML Web Service yang akan dibuat. Tetapi listing ini yang asalnya berupa file dengan ekstensi *asmx.vb* nantinya

akan menjadi file dengan ekstensi *dll* saat terjadi proses publikasi XML Web Service.

Sedangkan folder *App_Data* hanya akan memiliki isi jika pada saat pembuatan XML Web Service yang melibatkan database (akan dijelaskan pada bab selanjutnya).



3. Kini, berpindahlah ke window *code editor*, dan akan terlihat sebuah template XML Web Service pertama sebagai berikut :

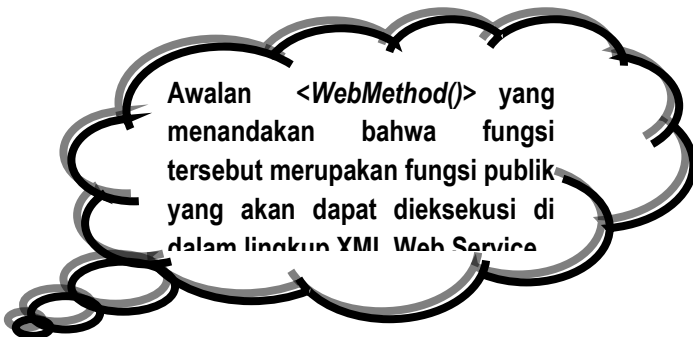
```

App_Code/Service.vb Start Page
(General)
1 Imports System.Web
2 Imports System.Web.Services
3 Imports System.Web.Services.Protocols
4
5 <WebService(Namespace:="http://tempuri.org/")> _
6 <WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
7 <Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
8 Public Class Service
9     Inherits System.Web.Services.WebService
10
11     <WebMethod()> _
12     Public Function HelloWorld() As String
13         Return "Hello World"
14     End Function
15
16 End Class
17

```

Pada template tersebut terdapat satu fungsi XML Web Service *built in* pertama yang dijadikan contoh yaitu fungsi untuk menampilkan kalimat *Hello World*. Yang perlu diperhatikan adalah awalan `<WebMethod(>` yang menandakan bahwa fungsi tersebut merupakan fungsi publik yang akan dapat dieksekusi di dalam lingkup XML Web Service. Jika nantinya terdapat fungsi lain tanpa prefix tersebut, maka fungsi itu nantinya akan menjadi fungsi “biasa” dalam class XML Web Service, tidak lagi dapat dieksekusi dalam lingkup XML Web Service.

4. Selanjutnya, fungsi contoh tadi dapat dihapus dan diganti dengan fungsi XML Web Service buatan kita sendiri. Berikut adalah fungsi XML Web Service pertama yang akan dibuat, yang memiliki fungsi untuk menjumlahkan dua buah bilangan integer :



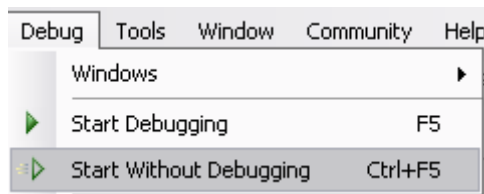
```
<WebMethod(> _
Public Function Penjumlahan _
    (ByVal a As Integer, _
```

```

ByVal b As Integer) As Integer
Return a + b
End Function

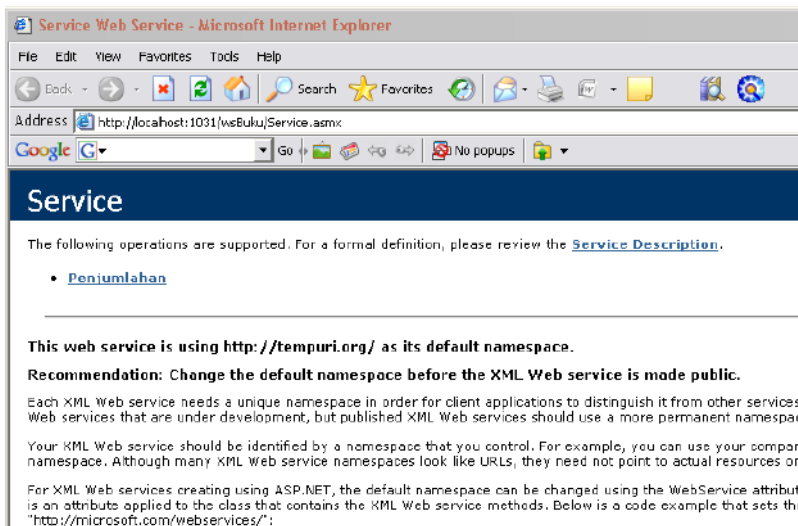
```

5. Kini, XML Web Service telah siap dieksekusi, pilih menu *Debug* → *Start Without Debugging* atau tekan kombinasi tombol *Ctrl+F5* untuk mengeksekusinya.



6. Jika pengetikan sudah benar, maka akan muncul browser default (tergantung dari browser default di komputer Anda masing-masing, tidak akan banyak berpengaruh, baik yang terinstall Internet Explorer ataupun Firefox), yang menampilkan daftar fungsi XML Web Service yang telah dibuat.

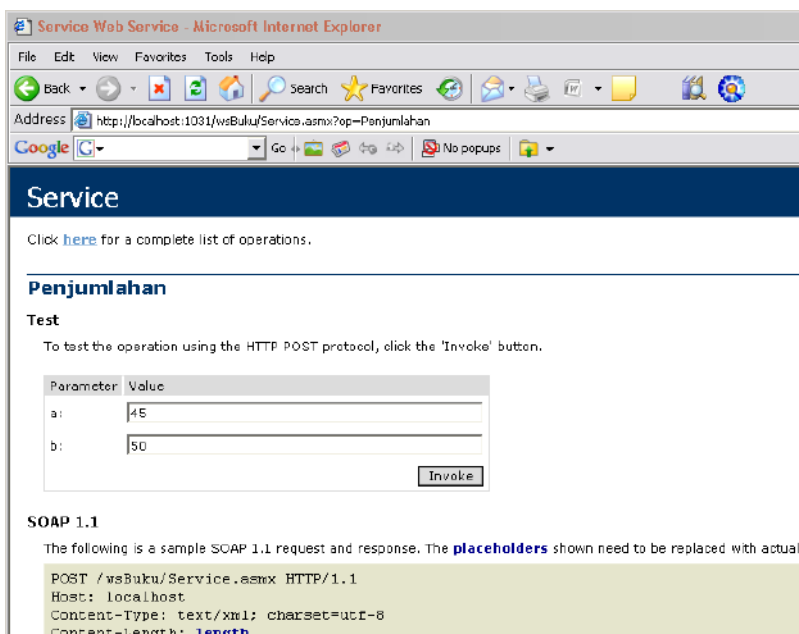
XML Web Service yang telah dibuat belum terkompilasi, sehingga meski sudah berhasil dieksekusi, tetapi XML Web Service tersebut belum siap untuk dipublish dan diupload ke sebuah web hosting. Agar bisa diupload, maka XML Web Service tersebut harus dipublish terlebih dulu.



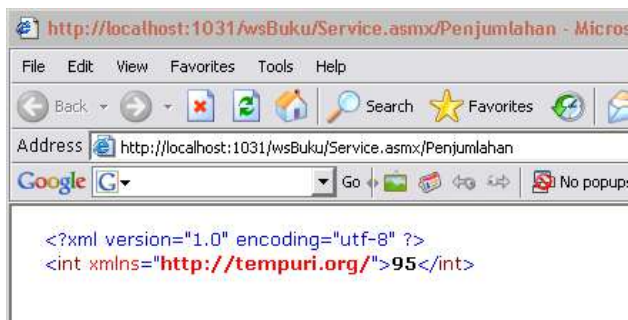
7. Jika diklik di link nama fungsi yaitu *Penjumlahan* maka parameter fungsi akan muncul untuk diisi.

Perlu diingat juga bahwa proses testing ini hanya berlaku saat XML Web Service berada di komputer lokal, jika XML Web Service telah diupload ke web hosting, maka proses invoke tidak berlaku lagi.

Hal ini disebabkan karena pada komputer lokal, kita bertindak sebagai administrator, sehingga proses tersebut dapat dijalankan, sedangkan pada web hosting kita hanya menjadi user yang diberi hak khusus untuk mengkonsumsi XML Web Service tersebut.




8. Hasil dari XML Web Service tersebut seperti pada gambar berikut :



String

Sekarang, XML Web Service yang telah berhasil dibuat kita lanjutkan dengan menambahkan satu fungsi baru untuk menghasilkan nilai balik dengan tipe string. Fungsi berikutnya yang akan dicoba adalah sebuah fungsi sederhana untuk menghasilkan sebuah rangkaian kata yang disusun dari huruf acak. Umumnya fungsi ini digunakan untuk melakukan *generate* password awal saat terjadi sebuah proses registrasi member dari sebuah sistem informasi. Berikut ini adalah langkah pembuatan fungsi tersebut (masih dalam lingkup XML Web Service yang sama dengan sub bab sebelumnya).



Sebuah XML Web Service dapat berisi lebih dari satu fungsi dengan tipe nilai balik yang berbeda.

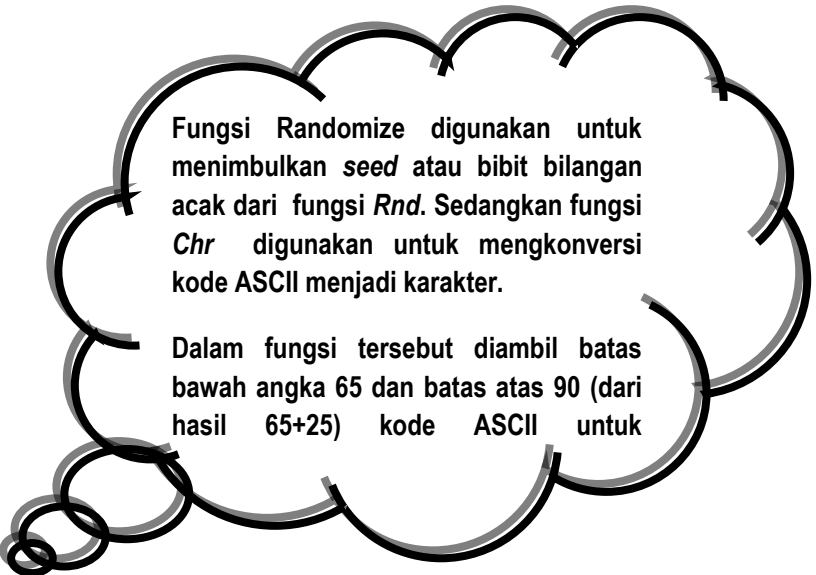
Terlebih jika semua fungsi tersebut memiliki relasi antara satu dengan yang lain, maka disarankan untuk

1. Setelah fungsi pertama dari XML Web Service yang mengembalikan nilai balik bertipe integer berhasil

dieksekusi, maka kini kita kembali ke window code editor.

2. Kini, di bawah fungsi *Penjumlahan*, buat fungsi baru dengan listing sebagai berikut :

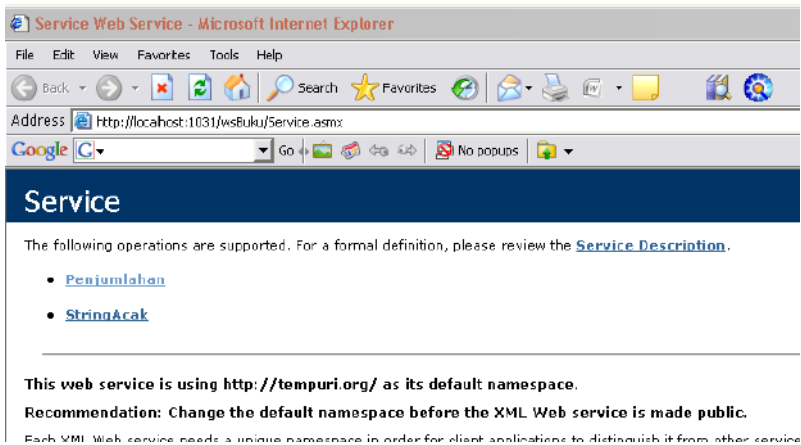
```
<WebMethod()>
Public Function StringAcak _
    (ByVal jumlah As Integer) _
    As String
    Dim xtemp As String
    For i As Integer = 1 To jumlah
        Randomize()
        xtemp &= Chr(CInt(25 * Rnd()) + 65)
    Next
    Return xtemp
End Function
```



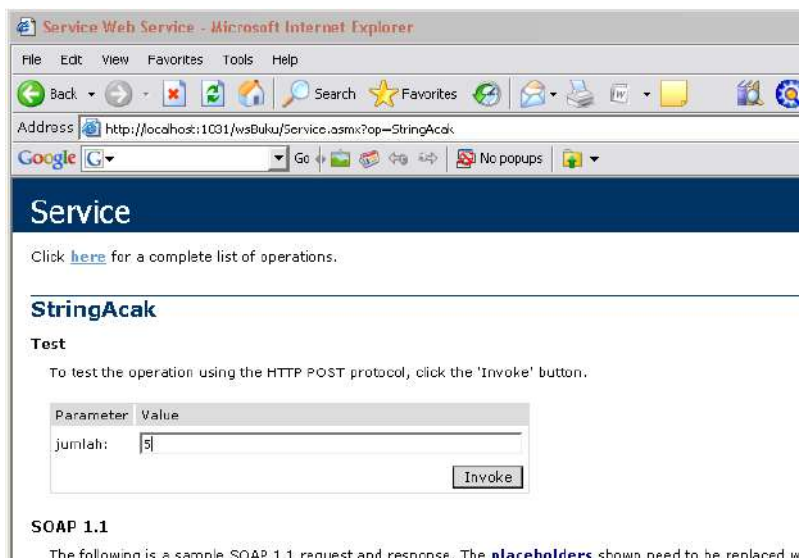
Fungsi *Randomize* digunakan untuk menimbulkan *seed* atau bibit bilangan acak dari fungsi *Rnd*. Sedangkan fungsi *Chr* digunakan untuk mengkonversi kode ASCII menjadi karakter.

Dalam fungsi tersebut diambil batas bawah angka 65 dan batas atas 90 (dari hasil 65+25) kode ASCII untuk

3. Eksekusi ulang XML Web Service tersebut dengan menekan kombinasi tombol *Ctrl+F5*



4. Pilih fungsi *StringAcak* dan berikan parameter dengan nilai 5



5. Hasil dari XML Web Service tersebut akan terlihat seperti pada gambar berikut (hasil dapat berubah pada

proses eksekusi Anda, karena proses yang terjadi akan mengikuti proses random atau acak).



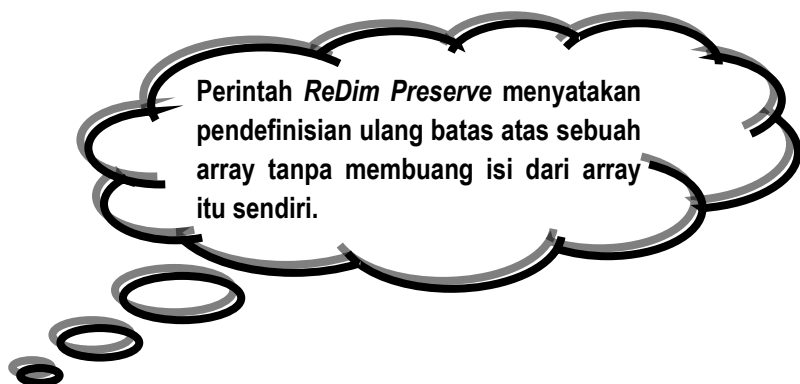
Dalam kedua fungsi di dalam XML Web Service tersebut tidak terdapat proses validasi, sehingga jika terjadi inputan parameter yang tidak sesuai, error yang terjadi belum terdefinisi.

Array

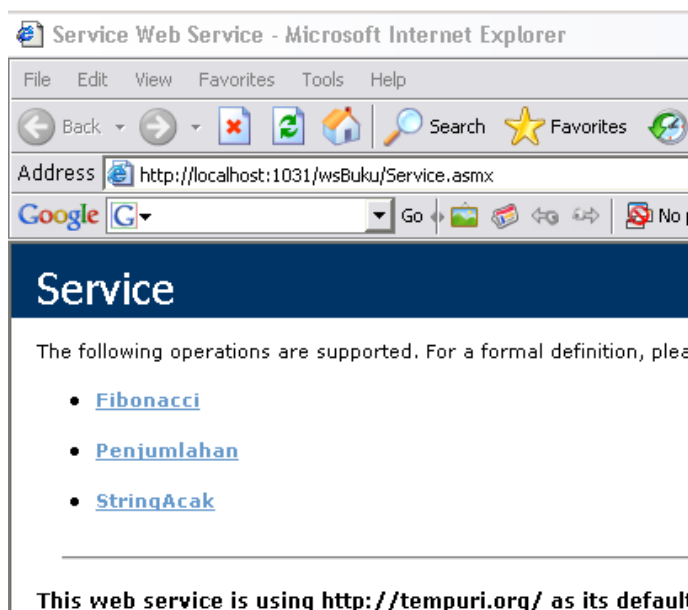
Berikutnya adalah fungsi ketiga dari XML Web Service yang akan menghasilkan nilai balik berupa array. Dalam contoh kali ini akan dibuat sebuah fungsi untuk menampilkan deret bilangan Fibonacci (deret bilangan yang mengasumsikan bahwa bilangan berikutnya adalah hasil penjumlahan dari dua bilangan sebelumnya) dengan batas maksimal sebagai parameter dari fungsi tersebut.

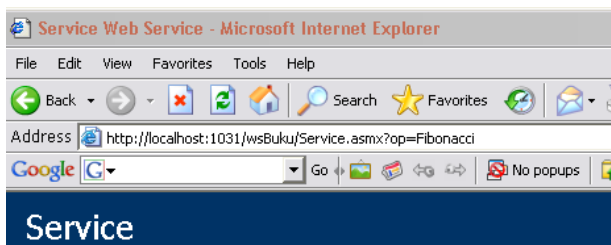
Masih dalam lingkup XML Web Service yang sama dengan dua sub bab sebelumnya, kini kita kembali lagi ke window *Code Editor* dan menuliskan fungsi ketiga sebagai berikut :

```
<WebMethod()> _
Public Function Fibonacci _
    (ByVal batas As Integer) _
    As Integer()
    Dim xtemp(1) As Integer
    Dim xawal As Integer = 2
    xtemp(0) = 0
    xtemp(1) = 1
    Do While xawal <= batas
        ReDim Preserve _
            xtemp(xawal)
        xtemp(xawal) = _
            xtemp(xawal - 1) + _
            xtemp(xawal - 2)
        xawal += 1
    Loop
    Return xtemp
End Function
```



Hasil yang terjadi terlihat pada rangkaian gambar berikut :





Click [here](#) for a complete list of operations.

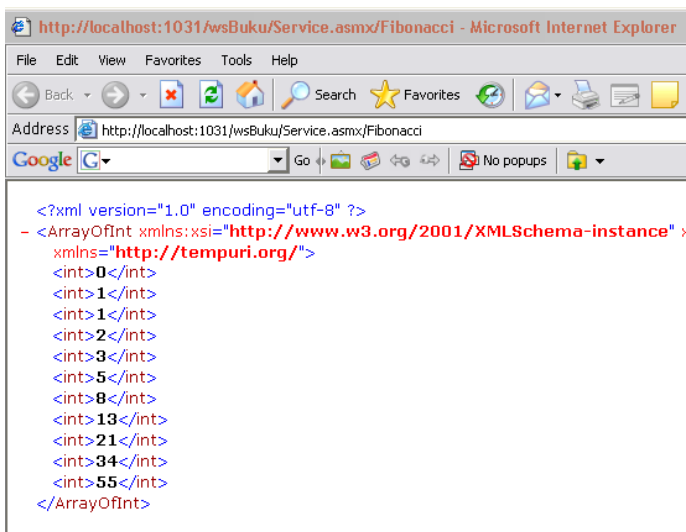
Fibonacci

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button

| Parameter | Value |
|-----------|---------------------------------|
| batas: | <input type="text" value="10"/> |

SOAP 1.1



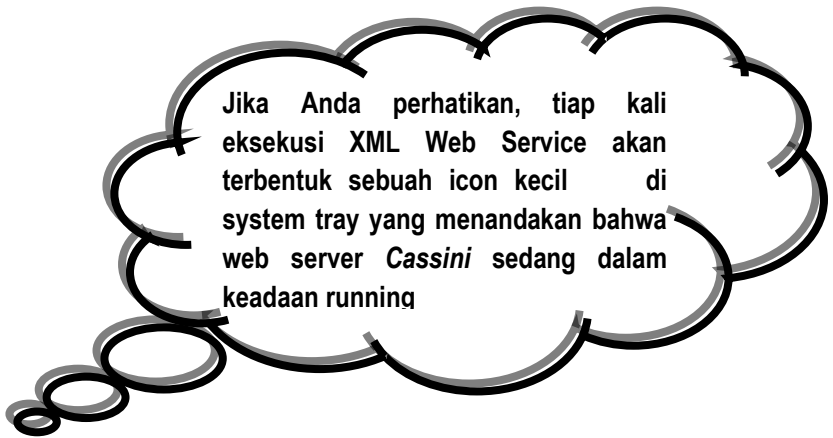
Publikasi dan Upload Web Server Lokal

Untuk mengimplementasikan lebih lanjut XML Web Service yang telah dibuat sebelumnya, maka langkah berikutnya adalah melakukan proses publikasi (publish) dan upload ke sebuah virtual directory dalam web server yang telah ditentukan. Sebagai percobaan awal, proses publikasi dan upload akan dilakukan secara lokal, atau di dalam komputer yang sama, sedangkan proses publikasi dan upload secara riil (online) akan dibahas di sub bab berikutnya.

Jika Anda bertanya, apakah fungsi dari XML Web Service yang sebelumnya telah dibuat tersebut belum bisa dikonsumsi oleh aplikasi lain ? Jawabannya bisa ya dan bisa juga tidak. Jawabannya “ya” jika kita melakukan konsumsi XML Web Service tersebut dalam sebuah project yang sama, tetapi jika kita ingin mengkonsumsi XML Web Service tersebut di aplikasi lain, misalkan di dalam aplikasi basis desktop, maka XML Web Service tersebut perlu dikompilasi dan dipublikasi terlebih dulu.

Jika Anda perhatikan ulang, maka tiap kali eksekusi XML Web Service, maka dalam browser akan terlihat sebuah alamat web server “gadungan” dengan menambahkan kode angka port secara acak (dalam tiap sesi eksekusi, kode angka port bisa berubah). Dan sesuai

dengan sifat ASP .NET 2.0 yang melakukan *compile on demand*, maka akan terbentuk sebuah virtual directory temporer hanya untuk sekedar mengeksekusi XML Web Service tersebut. Tetapi secara riil, belum terjadi sebuah virtual directory, dan juga XML Web Service yang ada masih dalam keadaan “mentah” alias belum terkompilasi dengan baik dan benar.



Karenanya, perlu dilakukan proses kompilasi dan publikasi terlebih dulu agar XML Web Service tersebut dapat dikonsumsi di aplikasi yang lain. Proses kompilasi tersebut memerlukan perintah yang harus dieksekusi melalui command prompt jika kita menggunakan Visual Web Developer Express. Tetapi jika menggunakan Visual Studio 2005, proses kompilasi (sekaligus publikasi) dapat dilakukan dengan memilih menu *Build → Publish Web Site*.

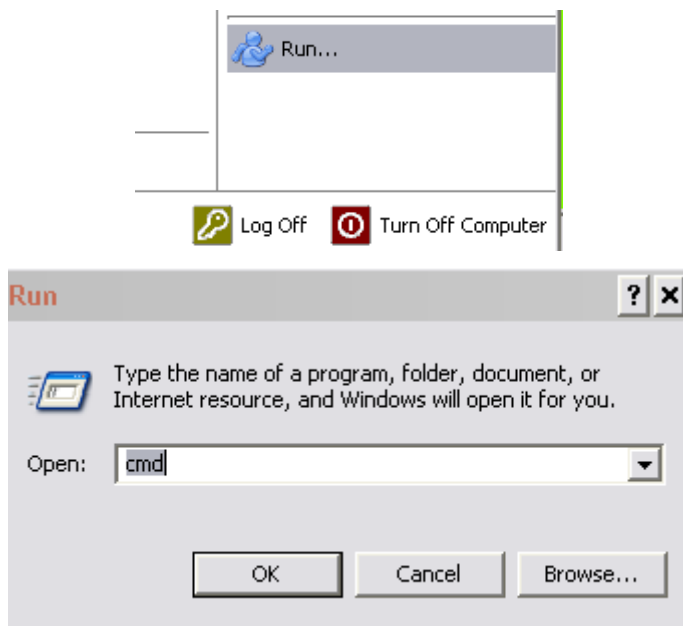
Perlu diperhatikan pula, jika Anda menggunakan .NET 1.x atau Visual Studio 2003, proses publikasi ini

tidak perlu dilakukan, karena secara default, saat Anda membuat XML Web Service dengan menggunakan Visual Studio 2003, maka virtual directory yang riil telah terbentuk secara otomatis dan saat XML Web Service dieksekusi juga secara otomatis akan melakukan proses kompilasi.



Berikut ini adalah langkah untuk melakukan proses kompilasi dan publikasi dengan menggunakan Visual Web Developer Express :

1. Masuklah ke dalam mode command prompt dengan memilih menu *Run* dari toolbar Start yang ada di Windows dan ketikkan *cmd* di dalam command.



2. Berikutnya ketikkan perintah berikut di dalam command prompt yang tersedia dan tekan tombol Enter untuk mengeksekusi (ingat, ketikkan perintah berikut hanya dalam satu baris eksekusi), dan jika proses eksekusi telah berhasil tutup command prompt tersebut dan lakukan langkah berikutnya :

```
"C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727
\aspnet_compiler.exe" -f -p
"C:\DataSoetam\TestLab\wsBuku" -v /
"C:\hasilWsBuku"
```

Sintaks dari perintah proses kompilasi adalah :

```
"C:\WINDOWS\Microsoft.NET\Framework
ork\v2.0.50727\aspnet_compiler.exe"
-f -p [nama folder asal XML Web
Service yang akan dikompilasi] -v /
[nama folder tujuan yang akan
dijadikan sebagai virtual directory]
```


3. Selanjutnya, cek di dalam Windows Explorer, apakah folder tujuan telah terbentuk dan cek isi yang ada didalamnya.

| Name | Size |
|-----------------------|------|
| bin | |
| PrecompiledApp.config | 1 KB |
| Service.asmx | 1 KB |
| web.config | 3 KB |

4. Jika sudah terbentuk, cek isi dalam folder *bin*, maka akan terlihat sebuah file dengan ekstensi *dll* yang merupakan hasil kompilasi dari source yang sebelumnya ada di dalam folder *App_Code*.

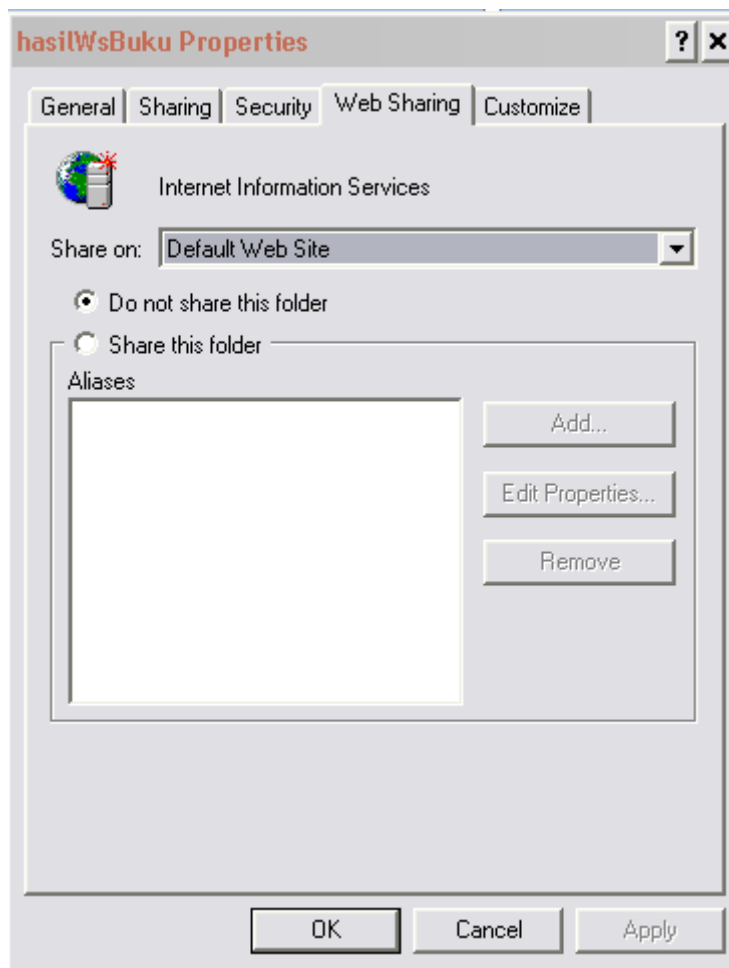
Sedangkan, jika Anda cek isi dari file *Service.asmx* maka hanya akan berisi satu baris *marker*.

5. Kini, buatlah sebuah virtual directory dari folder hasil tersebut dengan cara sebagai berikut (dengan asumsi bahwa IIS telah terinstalasi dengan baik dan dalam keadaan *running*) :



Secara default, jika Anda menggunakan Windows XP Pro atau Windows 2000, maka IIS akan terinstalasi dan dalam keadaan running. Anda bisa mengecek keadaan IIS dengan memilih *Control Panel* → *Administrative Tools* → *Internet Information Service* dan cek apakah *Default Web Site* telah running. Jika tidak, maka Anda perlu melakukan instalasi IIS melalui *Control Panel* → *Add or Remove Programs* → *Add/Remove Windows Components* → *IIS* atau dengan jalan menginstall Service Pack terbaru.

- a. Buka Windows Explorer, dan arahkan ke folder hasil kompilasi yang telah terbentuk.
- b. Klik kanan pada folder tersebut dan pilih *Properties* dan klik pada tab *Web Sharing*

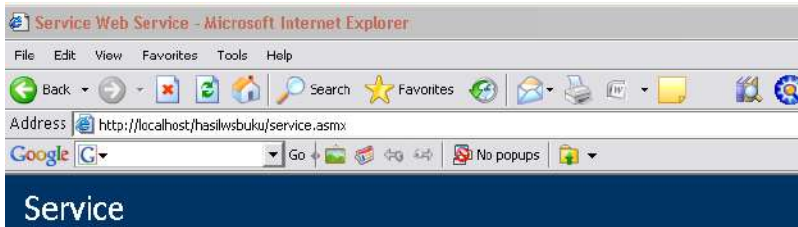


- c. Selanjutnya, klik opsi *Share this folder*, dan klik OK. Jika semuanya berjalan benar, maka virtual

directory baru dengan nama alias yang sama dengan folder tersebut telah terbentuk.

- d. Untuk melakukan cek virtual directory tersebut, bukalah browser dan ketikkan url berikut :

`http://localhost/[nama folder]/Service.asmx`



The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Fibonacci](#)
- [Penjumlahan](#)
- [StringAcak](#)

This web service is using `http://tempuri.org/` as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

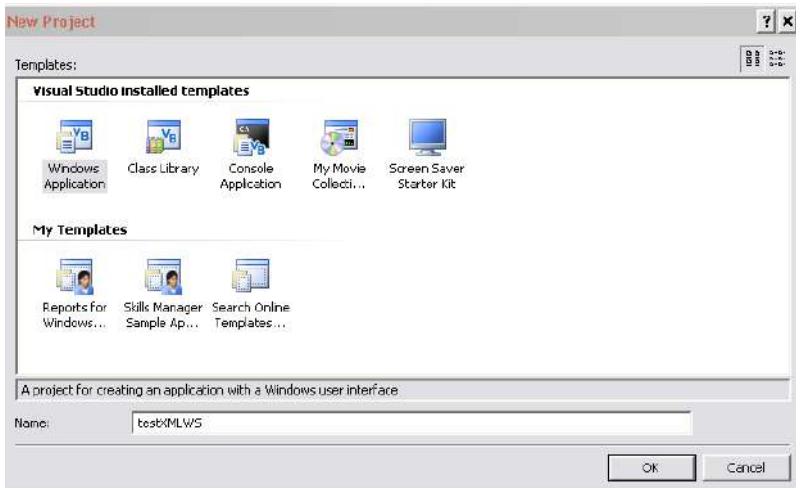


Cara lain untuk membuat virtual directory adalah dengan masuk ke dalam Internet Information Service dan melakukan klik kanan pada Default Web Site dan membuat virtual directory baru

Konsumsi XML Web Service

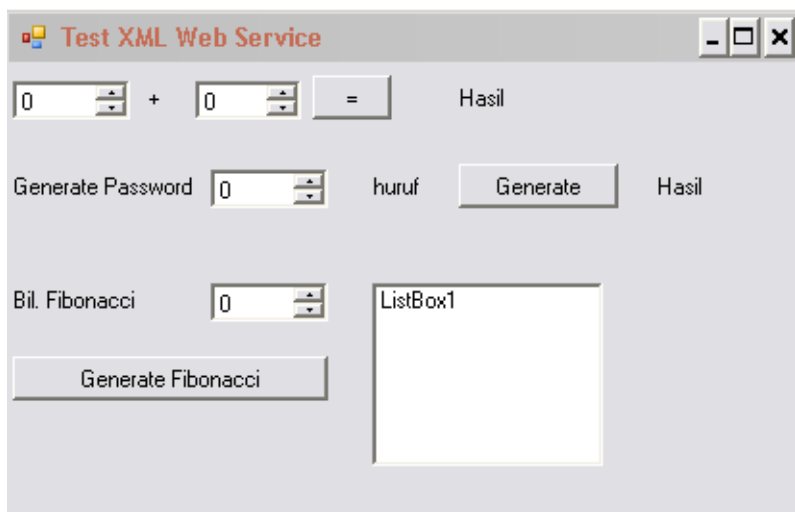
Sekarang, dengan adanya virtual directory yang telah terbentuk, maka kita siap untuk mengkonsumsi XML Web Service yang sebelumnya telah dibuat. Cara konsumsi dari XML Web Service yang akan dilakukan, kurang lebih sama dengan cara yang ada pada bab sebelumnya. Berikut adalah langkah yang harus dilakukan untuk mengkonsumsi XML Web Service dalam sebuah aplikasi desktop dengan menggunakan Visual Basic 2005 Express Edition :

1. Buka Visual Basic 2005 Express Edition, lalu buat sebuah project baru dengan tipe *Windows Application*.

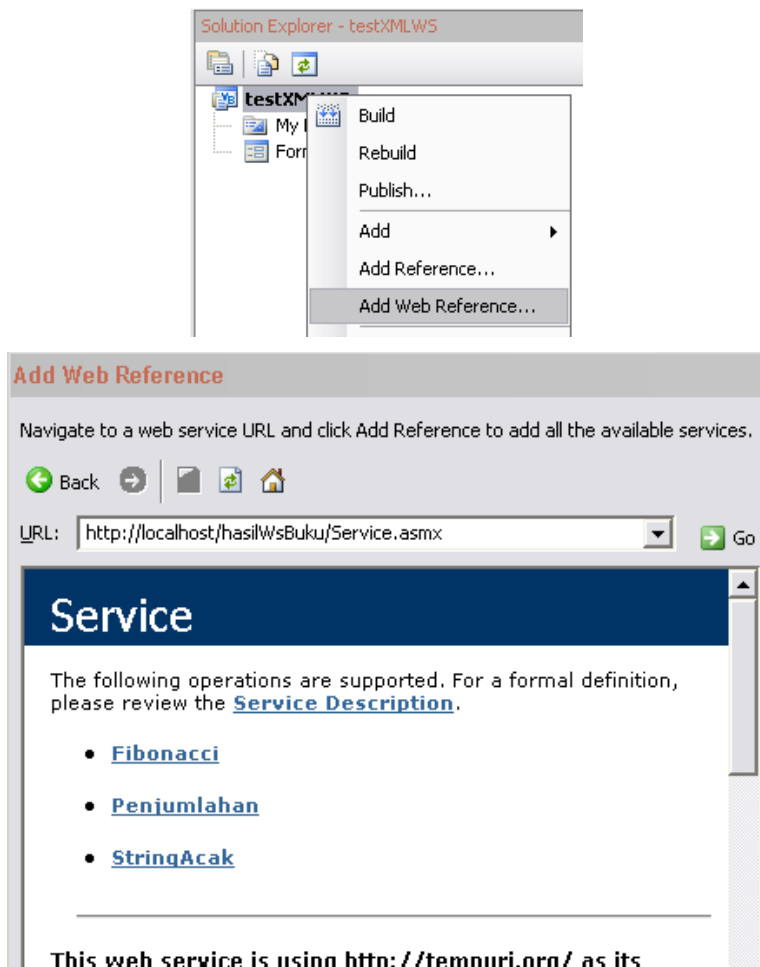


2. Kemudian dalam desain form yang tersedia, buatlah layout form untuk melakukan testing XML Web Service yang telah dibuat dengan menempatkan empat

komponen *NumericUpDown*, enam *Label*, tiga *Button* dan sebuah *ListBox*. Selanjutnya, susunlah komponen-komponen tadi seperti pada gambar berikut :

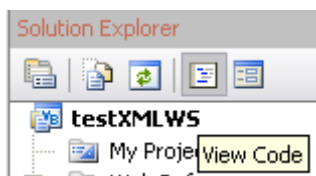


- Setelah desain form selesai dibuat, maka berpindahlah ke window *Solution Explorer*, dan klik kanan pada nama project yang tersedia, lalu pilih sub menu *Add Web Reference* hingga muncul sebuah kotak dialog untuk memilih URL XML Web Service yang tersedia. Lalu, ketikkan URL dari virtual directory yang telah dibuat sebelumnya pada kotak dialog dan tekan tombol Enter hingga isi dari XML Web Service muncul.



4. Pada kotak dialog yang tersedia, tekan tombol *Add Reference* untuk menambahkan XML Web Service sebagai proxy class dalam project yang tersedia.
5. Sekarang, berpindahlah ke window *Code Editor* dari form, dan tepat di bawah deklarasi class form (*Public*

Class Form1), ketikkan deklarasi obyek XML Web Service sebagai berikut :



Dim XMLWS As New localhost.Service

6. Kini, berpindahlah lagi ke mode desain form, dan double klik pada button pertama yaitu button untuk menampilkan hasil penjumlahan, dan ketikkan listing berikut untuk mengimplementasikan XML Web Service penjumlahan dua bilangan integer :

```
Label2.Text = _
    XMLWS.Penjumlahan(NumericUpDown1.Value, _
    NumericUpDown2.Value)
```

7. Dengan cara yang sama, ketikkan listing berikut untuk button *Generate Password* dan button *Generate Fibonacci* :

Button Generate Password

```
Label15.Text = _
    XMLWS.StringAcak(NumericUpDown3.Value)
```

Button Generate Password

```
ListBox1.DataSource = _
    XMLWS.Fibonacci(NumericUpDown4.Value)
```

8. Langkah terakhir adalah mencoba untuk mengeksekusi project untuk testing XML Web Service tersebut, tekan tombol F5 untuk mengeksekusi form tersebut, dan lihat hasil yang ada seperti pada gambar berikut :

Test XML Web Service

23 + 45 = 68

Generate Password 6 huruf XDTTNG

Bil. Fibonacci 6

0
1
1
2
3
5
8

Cobalah untuk melakukan testing XML Web Service tersebut dengan berbagai jenis nilai inputan.

Troubleshooting ? Directory Security ?

Tipe data kompleks

Implementasi XML Web Service dengan pengembalian nilai balik berupa tipe data kompleks, sesungguhnya sama halnya dengan tipe data primitif. Karena tipe data apapun yang dikembalikan oleh sebuah fungsi dalam XML Web Service tetap saja berupa file XML dan berupa string biasa. Perbedaan utamanya adalah saat

Upload dan Publikasi Online

XML Web Service dan Database

Review ADO .NET

Connected application

Datareader

Command

Disconnected application

Dataset

XML Web Service dan Microsoft Office

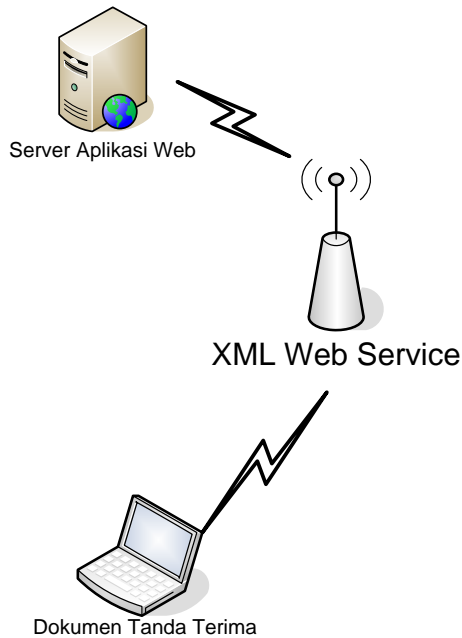
Apa yang dibutuhkan ?

Jika Anda bertanya, apakah Microsoft Office juga mampu mengkonsumsi XML Web Service ? Jawabannya adalah *ya* dengan menggunakan Visual Basic for Application atau lebih sering disebut sebagai *macro*. Tetapi perlu diingat bahwa Microsoft Office tidak mampu melakukan proses produksi XML Web Service karena keterbatasan yang dimiliki oleh Visual Basic for Application itu sendiri.

Versi Microsoft Office yang mampu melakukan konsumsi XML Web Service adalah Microsoft Office XP, 2003 dan 2007. Untuk Microsoft Office XP dan 2003, diperlukan utilitas *Office Web Service Toolkit* yang dapat didownload secara gratis dari situs www.microsoft.com ? Sedangkan untuk Microsoft Office 2007 telah tersedia fasilitas *built in* untuk melakukan konsumsi XML Web Service secara otomatis dalam Visual Basic for Application.

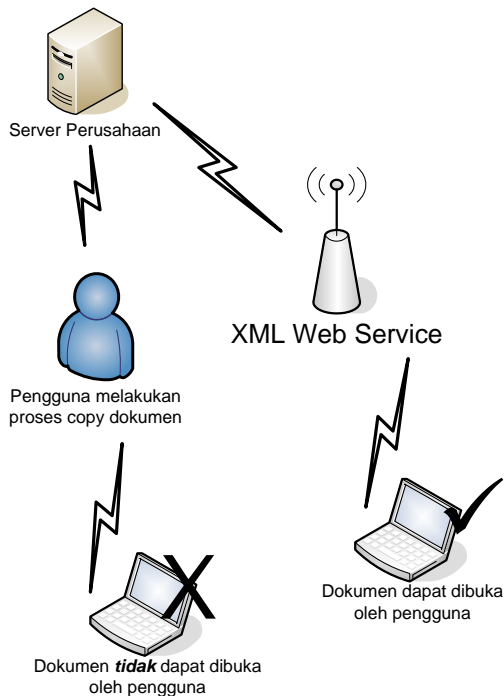
Tetapi, jika Anda bertanya lagi, mengapa harus mengkonsumsi XML Web Service dalam sebuah dokumen Microsoft Office ? Jawaban sederhananya adalah proses otomatisasi ! Proses otomatisasi yang seringkali dibutuhkan di sebuah lingkup perkantoran modern, misalkan saja : sebuah dokumen Microsoft Word berupa surat tanda terima dari sebuah transaksi online yang membutuhkan data transaksi dari situs perusahaan, maka

departemen marketing hanya perlu dibuatkan sebuah template surat yang didalamnya telah diinjeksikan XML Web Service yang terkoneksi dengan situs transaksi, sehingga data transaksi yang dibutuhkan dapat langsung diambil secara otomatis, dan yang terpenting, pengguna dokumen tersebut, tidak perlu tahu apa yang sebenarnya terjadi dalam proses pembuatan surat tadi.



Contoh lain adalah untuk kepentingan proteksi, bayangkan saja jika terdapat sebuah dokumen yang dianggap sangat rahasia bagi perusahaan, dan dokumen tersebut tidak boleh dibuka di luar lingkup perusahaan tersebut. Maka solusi yang bisa ditawarkan oleh XML Web Service adalah dengan melakukan injeksi terhadap

dokumen tersebut, sehingga dokumen tersebut hanya akan dapat dibuka jika dalam keadaan terkoneksi dengan server XML Web Service yang terdapat di dalam perusahaan, dengan kata lain, dokumen tersebut tidak akan dapat dibuka di luar perusahaan, meski dilakukan proses copy ke media lain.



Jadi, penggunaan XML Web Service sendiri memiliki jangkauan yang sangat luas, tidak hanya terbatas pada penggunaan fungsi secara global, tetapi juga merambah ke area lain seperti keamanan data, otomatisasi perkantoran (groupware) dan juga otomatisasi dokumen lintas platform yang menyebabkan proses aliran

data dapat berjalan lebih luas dan tidak lagi memperdulikan keseragaman bahasa pemrograman ataupun format data yang akan digunakan. Tentunya masih sangat banyak kasus yang dapat ditangani dengan solusi XML Web Service, bergantung kepada situasi dan kondisi yang dihadapi di suatu tempat tertentu.

Dan meski buku ini hanya menampilkan sebagian kecil dari contoh kasus sederhana, diharapkan tetap dapat memberikan inspirasi bagi solusi masalah yang akan dihadapi di dalam dunia nyata.

XML Web Service dalam Word

Solusi XML Web Service pertama yang akan dibuat adalah solusi XML Web Service dengan menggunakan Microsoft Word. Dalam kasus ini akan digunakan Microsoft Word 2007, meski solusi ini juga dapat dijalankan di dalam Microsoft Office 2003 bahkan juga versi XP.

XML Web Service dalam Excel

30. Your development team is creating a new Windows-based application for the company. The application consists of a user interface and several XML Web services. You develop all XML Web services and perform unit testing. Now you are ready to write the user interface code. Because some of your servers are being upgraded, the XML Web service that provides mortgage rates is currently offline. However, you have access to its description file. You must begin writing code against this XML Web service immediately. What should you do?

- A. Generate the proxy class for the XML Web service by using Disco.exe.
- B. Generate the proxy class for XML Web service by using WSDL.exe.
- C. Obtain a copy of the XML Web service assembly and register it on your local development computer.
- D. Add the description file for the XML Web service to your Visual Studio .NET project.

Answer: B

Explanation: Ordinarily to access an XML Web service from a client application, you first add a Web reference, which is a reference to an XML Web service. When you create a Web reference, Visual Studio creates an XML Web service proxy class automatically and adds it to your project. However, you can manually generate a proxy class using the XML Web services Description Language Tool, WSDL.exe, used by Visual Studio to create a proxy class when adding a Web reference. This is necessary when you are unable to access the XML Web service from the machine on which Visual Studio is installed, such as when the XML Web service is located on a network that will not be accessible to the client until run time. You then manually add the file that the tool generated to your application project. Reference: Visual Basic and Visual C# Concepts, Locating XML Web Services Visual Basic and Visual C# Concepts, Generating an XML Web

Service Proxy

44. You develop a Windows-based application named Retail3 by using Visual Studio .NET. Retail3 consumes an XML Web service named MortgageRate and exposes a method named GetCurrentRate. Retail3 uses GetCurrentRate to obtain the current mortgage interest rate. Six months after you deploy Retail3, users begin reporting errors. You discover that MortgageRate has been modified. GetCurrentRate now requires you to pass a postal code before returning the current mortgage interest rate. You must ensure that Retail3 consumes the most recent version of MortgageRate. You must achieve this goal in the most direct way possible. What should you do?

- A. Use Disco.exe to generate a new proxy class for MortgageRate.
- B. Modify the Retail3 code to pass the postal code to GetCurrentRate.
- C. Use the Update Web Reference menu item to update the reference to MortgageRate in Retail3.
- D. Use the Add Reference dialog box to recreate the reference to MortgageRate in Retail3.
- E. Remove the reference to MortgageRate in Retail3. Use the Add Web Reference dialog box to create the reference.

Answer: C

Explanation: If your application contains a Web reference to an XML Web service that has been recently modified on the server, you may need to update the reference in your project. To update a project Web reference 1. In Solution Explorer, access your project's Web References folder and select the node for the Web reference you want to update. 2. Right-click the reference and click Update Web Reference. Reference: Visual Basic and Visual C# Concepts, Managing Project Web References

45. Your development team is creating a Windows-based application for the company. The application asynchronously calls the `ProcessLoan` method of an XML Web service. The XML Web service will notify your code when it finished executing `ProcessLoan`.

You must ensure that your code can continue processing while waiting for a response from the XML Web service. Your code must establish when `ProcessLoan` finished executing.

What should your application do?

A. Use the `WaitHandle.WaitAny` method of the `IAAsyncResult.AsyncWaitHandle` object.

Examine the value of `IAAsyncResult.IsCompleted` to see if `ProcessLoan` is finished executing.

B. Use the `WaitHandle.WaitAll` method of the `IAAsyncResult.AsyncWaitHandle` object.

Examine the value of `IAAsyncResult.IsCompleted` to see if `ProcessLoan` is finished executing.

C. Supply a callback delegate to the `BeginProcessLoan` method of the XML Web service.

After the XML Web service returns its response, a thread will invoke the callback from the threadpool.

D. Supply a callback delegate to the `EndProcessLoan` method of the XML Web service.

After the XML Web service returns its response, a thread will invoke the callback from the threadpool.

Answer: C

Explanation: Calling an XML Web service asynchronously is a two-step operation. The first step, calling the `Begin` method, initiates the XML Web service call. The second step, calling the `End` method, completes the XML Web service call and returns the XML Web service response. There are different methods to determine when the asynchronous XML Web service call has completed. The preferred and most efficient method is to supply a callback delegate to the `Begin` method. Reference: Visual Basic and Visual C# Concepts, Accessing an XML Web Service Asynchronously in Managed Code Incorrect Answers A, C: It is possible to use `IAAsyncResult.AsyncWaitHandle` object to determine when the asynchronous XML Web service call has completed. This is less efficient

though. D: The callback delegate should be supplied to the Begin method, not the End method.

TestKing Inc. provides a credit card processing application for its customers. The current application supports only computers that run on a Microsoft Windows operating system.

You are asked to rewrite the current application as a .NET application. This .NET application does not need to be backward compatible with the current application.

You must ensure that this new application meets the following requirements:

- Must support asynchronous processing.
- Must be able to pass data through firewalls.
- Must pass only SOAP-Compliant formatted data validated by using an XSD schema.
- Must not be limited to client computers running on a Microsoft operating system.

You want to accomplish this task by using the minimum amount of development effort. Which type of .NET application should you use?

- A. Windows service
- B. XML Web service
- C. Serviced component
- D. .NET Remoting object

Answer: B

Explanation/Reference:

Explanation: An XML Web service would:

- * support asynchronous processing.
- * XML traffic would be allowed to passed through firewalls
- * can use SOAP-compliant formatted data validated by an XSD schema.
- * could be implemented on heterogeneous systems.

Reference: Designing Distributed Applications with Visual Studio .NET, Programming the Web with XML Web Services

Incorrect Answers

A: A Windows service can only run on a Windows computer.

C: Serviced components cannot be run on heterogenous systems.

Note: A serviced component is a class that is authored in a CLS-compliant language and that derives directly

or indirectly from the System.EnterpriseServices.ServicedComponent class. Classes configured in this way can

be hosted by a COM+ application and can use COM+ services.

D: .NET Remoting objects can be run on different operating systems. However, a XML Web service meets the

requirement in a better way.

You create an XML Web service that provides stock information to its customers. You successfully test the service. You are now ready to deploy the service to a new virtual directory on a production computer. You want to deploy the service without having to manually configure any settings on the production computer.

Which deployment mechanism should you use?

A. FTP

- B. Xcopy command
- C. Web setup project
- D. Copy Project command

Answer: C

Explanation: Using deployment to install files on a Web server provides an advantage over simply copying files, in that deployment handles any issues with registration and configuration automatically.

Reference: Visual Studio, Deployment of a Web Setup Project

Incorrect Answers

A: Download files through FTP would require manual registration and configuration.

B: Just copying the files would require manual registration and configuration.

D: Copying a project, rather than deploying it, is the simpler way to move your project's content to a target

Web server. However, copying does not automatically configure Internet Information Services (IIS) directory settings.

You create an XML Web service named myService. This service exposes a Web method named MyMethod. You need to register myService in UDDI. First, you add a new business name and a new tModel. You now need to list a valid access point to myService.

Which URL should you use?

- A. <http://TestKServer/AppPath/myService>
- B. <http://TestKServer/AppPath/myService=wsdl>
- C. <http://TestKServer/AppPath/myService.asmx>
- D. <http://TestKServer/AppPath/myService.asmx?MyMethod>

Answer: C

Explanation/Reference:

Explanation: XML Web services are access through Web browser using an URL. The URL to access an XML

Web service has the format:

<http://servername/apppath/webservicename.asmx>

The XML Web service's HTML description file is displayed. The XML Web service's HTML description page shows you all the XML Web service methods supported by a particular XML Web service, including a valid access to the method.

Note: The UDDI (Universal Description, Discovery and Integration) specifications define a standard way to publish and discover information about XML Web services. The XML schemas associated with UDDI define four types of information that would enable a developer to use a published XML Web service. These are: business information, service information, binding information, and information about specifications for services.

Incorrect Answers

A: The .asmx extension must be specified as well.

B: Use of Web Services Description Language (WSDL) is not required (and the syntax is incorrect as well). **D:** This is not the format used to access a method within a XML Web service. The format to directly access an method within a service is:

<http://servername/appath/webservicename.asmx/Methodname?parameter=value>

QUESTION 133

You create an XML Web service named WeatherService. This service contains a Web method named RetrieveWeather. RetrieveWeather takes as input a city named and returns the current weather conditions for that city.

You need to provide callers of this service with the URL they need to issue an HTTP-GET against WeatherService.

Which URL should you use?

A.

<http://TestKingSrv/AppPath/WeatherService.asmx/cityname=somecity>

B.

<http://TestKingSrv/AppPath/WeatherService.asmx/RetrieveWeather?cityname=somecity>

C.

<http://TestKingSrv/AppPath/WeatherService/RetrieveWeather.asmx?cityname=somecity>

D.

<http://TestKingSrv/AppPath/WeatherService/RetrieveWeather?cityname=somecity>

Answer: B