

Berkenalan dengan Windows Communication Foundation (WCF)

Isman Subarkah

ismansubarkah91@gmail.com

http:// isman_subark12a.student.ipb.ac.id

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Peningkatan jumlah penjualan *smartphone* akhir-akhir ini meningkat tajam. Hal ini yang mendorong agar *developer/programmer* mau mempelajari ataupun beralih dari *developer* aplikasi *desktop* atau *web* ke aplikasi *mobile*. Tantangan para *developer* dalam mengembangkan aplikasi *mobile* ialah bagaimana membuat aplikasi yang berkualitas namun memperhatikan *resource* dari *smartphone/device* tersebut, mengingat spesifikasi *hardware* pada *smartphone* masih terbatas dan tidak sama seperti komputer. Sehingga perlu kreativitas para *developer* dalam menentukan proses apa saja ada pada aplikasi *mobile*. Oleh karena itu, pemisahan mengenai proses/aktivitas apa yang sebaiknya dan tidak sebaiknya dilakukan oleh aplikasi *mobile* perlu dipertimbangkan oleh *developer*.

Gagasan yang telah dijelaskan pada paragraf sebelumnya mengingatkan penulis ketika penulis sedang mengembangkan sebuah aplikasi *mobile*. Pada saat itu, penulis sedang berdiskusi dengan rekan-rekan tentang proses bisnis apa yang harus dilakukan aplikasi sehingga aplikasi tidak memberatkan *resource* perangkat. Hasil diskusi tersebut ialah bahwa proses bisnis *create, read, update, dan delete* sebaiknya tidak dilakukan di aplikasi *mobile* namun di sisi *server*. Sedangkan, aplikasi pada *mobile* hanya melakukan logika bagaimana ia berinteraksi dengan pemakai. Kesimpulan diskusi ini berdasar pada konsep *Service Oriented Architecture (SOA)*. Penulis berpendapat bahwa kesimpulan tersebut bukan solusi yang tepat. Namun, ketika *developer* mengembangkan sebuah sistem yang saling terintegrasi dan sistem tersebut akan diakses oleh platform yang berbeda, konsep *SOA* merupakan solusi yang sangat tepat. Salah satu teknologi yang mampu mengimplementasikan konsep *SOA* yaitu *Windows Communication Foundation (WCF)*.

WCF merupakan *Software Development Kit (SDK)* yang dirancang oleh *Microsoft* untuk membangun *SOA*. *WCF* dapat membantu *developer* dalam membuat *service* yang *secure*. *WCF* juga dapat membuat *service* yang dapat digunakan secara bersama meskipun berbeda *platform*.

Latar Belakang dan Pendahuluan

Latar belakang dalam penulisan ini adalah bahwa sangat sedikitnya tutorial yang menjelaskan bagaimana menerapkan atau membuat sebuah *networks-distributed service* dengan WCF secara kontinu. Meskipun tidak menutup kemungkinan bahwa tutorial tersebut ada, namun tutorial tersebut biasanya dalam bahasa Inggris. Hal ini yang menjadikan pembaca cukup sulit untuk memahami konsep-konsep yang dijelaskan. Di sisi lain, jika ada pembaca yang masih bingung dengan penjelasan pada tutorial ini, pembaca dapat menanyakan kepada penulis. Sehingga diharapkan pembaca dapat memahami isi tutorial ini dan tutorial ini bermanfaat bagi pembaca.

Untuk mempelajari materi/tutorial ini, pembaca minimal pernah melakukan pemrograman dengan C# dan sedikit pemahaman Pemrograman Berorientasi Objek. Namun pembaca tidak perlu khawatir karena nanti akan coba penulis jelaskan sintak-sintak yang ada ☺.

- **Development tools**

Development tools yang diperlukan pada tutorial ini yaitu:

1. Microsoft Visual Studio 2010 atau Microsoft Visual Studio 2012 (di sini penulis menggunakan Microsoft Visual Studio 2012).
2. Windows 7 atau Windows 8 (di sini penulis menggunakan Windows 2012).
3. Windows Communication Foundation 4.
4. Internet Information Service (IIS) 7 atau 7.5 atau 8 (di sini penulis menggunakan IIS 8).

- **Estimasi Waktu**

Waktu yang dibutuhkan untuk mempelajari tutorial ini kurang lebih 30 menit.

- **Deskripsi Aplikasi**

Aplikasi WCF akan memproses permintaan dari aplikasi *client*. Di sini, aplikasi *client* berupa aplikasi console. Proses yang dilakukan aplikasi WCF adalah aplikasi WCF mengirimkan data berupa string yang isinya “Hello World, ini adalah program WCF pertamaku” kepada aplikasi *client*/ aplikasi yang melakukan *request* ke *server*.

Isi

Sebelum mempelajari WCF, ada baiknya kita mengetahui dahulu konsep dasar WCF.

Definisi WCF

WCF merupakan *Software Development Kit* (SDK) atau *framework* yang dirancang oleh Microsoft untuk membangun SOA. WCF dapat membantu *developer* dalam membuat *service* yang *secure* dan *reliable*. WCF juga dapat membuat *service* yang dapat digunakan secara bersama meskipun berbeda *platform*.

WCF merupakan *unified framework* yang menyediakan fasilitas:

1. NET Remoting.
2. Distributed Transactions.
3. Message Queues.
4. Web Services.

Semua fasilitas tersebut terdapat pada model tunggal pemrograman berorientasi *service* untuk *distributed computing*.

Fitur dari WCF

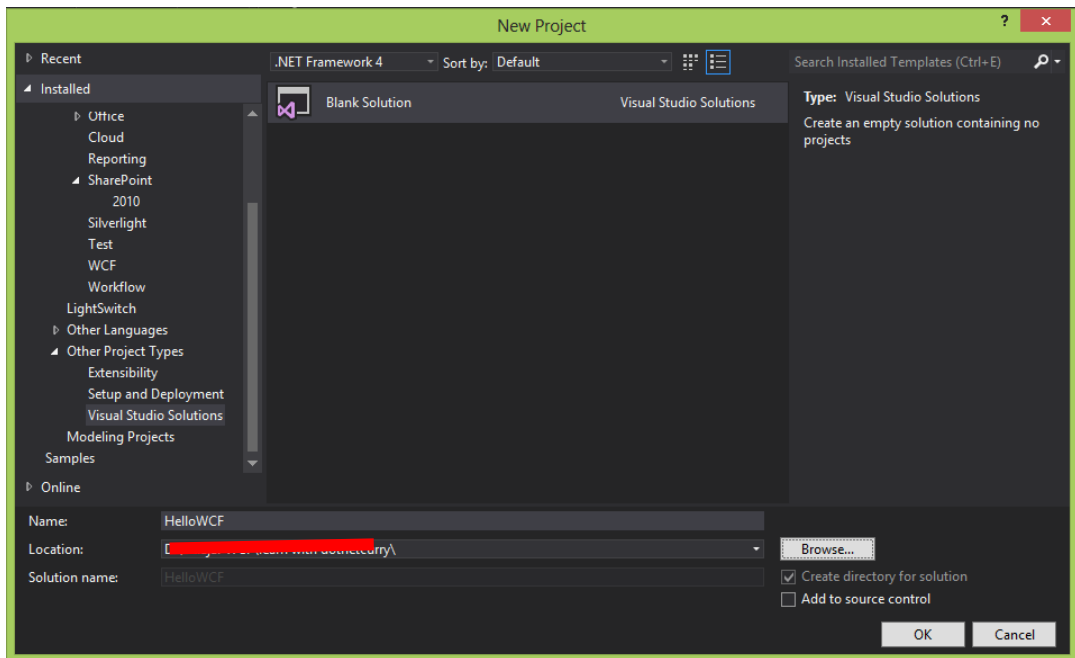
1. Service Orientation.

2. Interoperability.
3. Multiple Message Patterns.
4. Service Metadata.
5. Data Contracts
6. Security.
7. Multiple Transport and Encodings.
8. Reliable and Queued Messages.
9. Durable Messages.
10. Transactions.
11. AJAX and REST Support.
12. Extensibility.

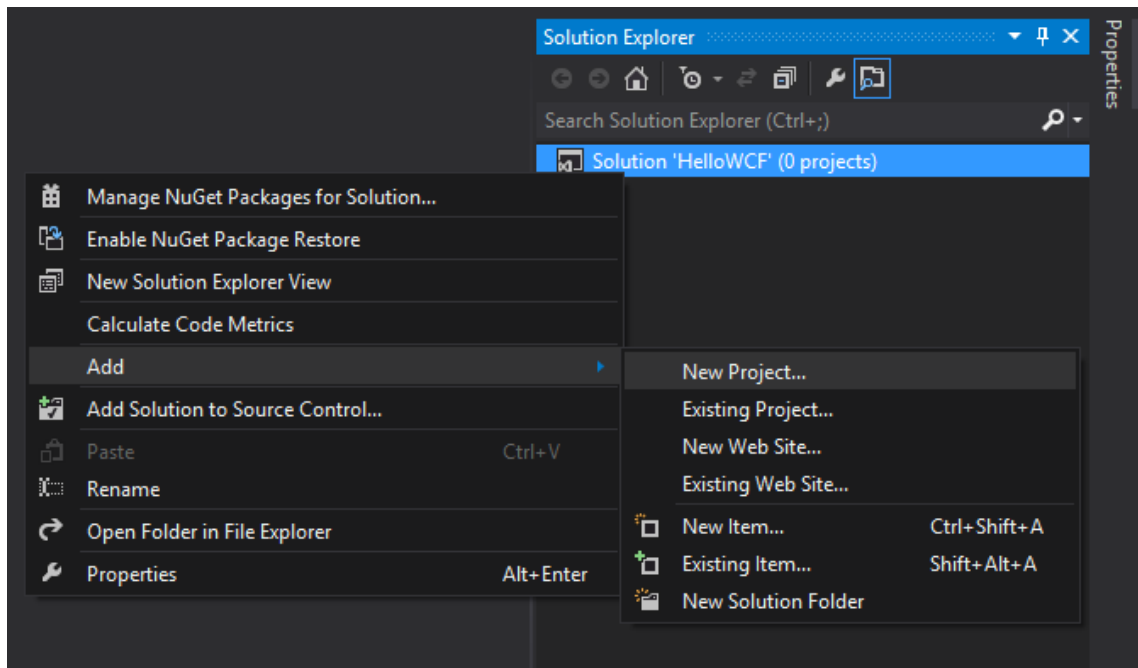
Explore The Code (Sesi 1)

Pada bagian ini kita langsung ke bagian praktik pemrograman untuk WCF Service.

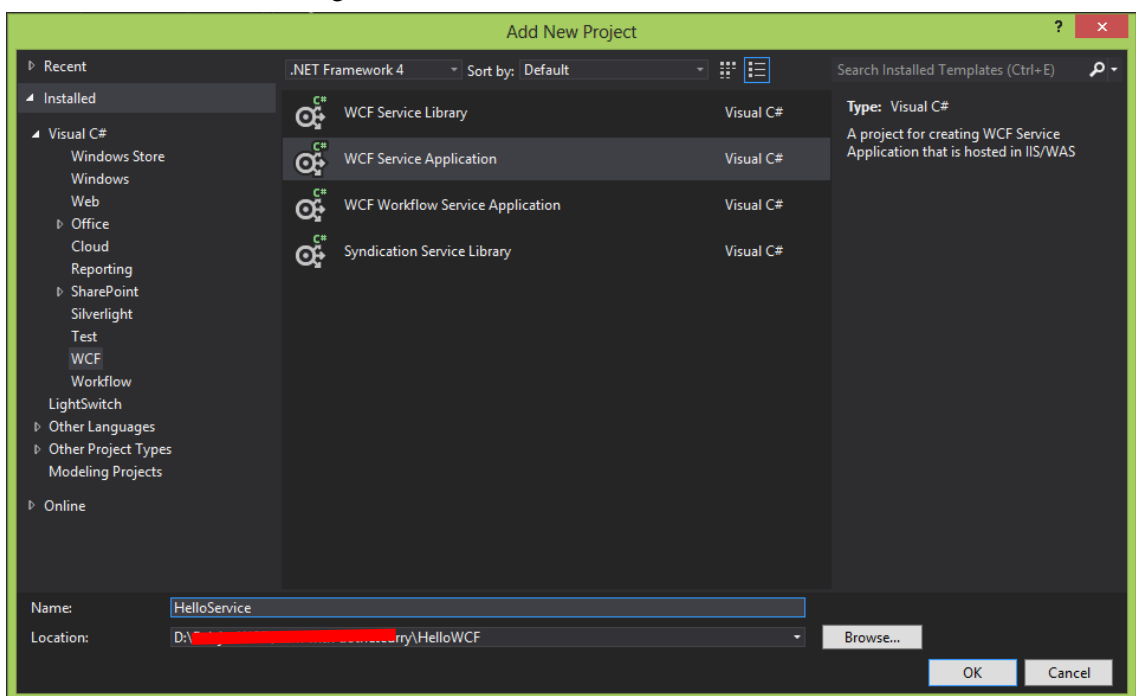
1. Buka Visual Studio 2010 atau 2012 and klik File -> New -> Project
2. Pilih Visual Studio Solution pada panel bagian kiri, kemudian isikan Name dengan “HelloWCF”.



3. Kemudian klik OK.
4. Setelah proses tersebut dilakukan maka Visual Studio akan menampilkan Solution Explorer dan terdapat sebuah solution dengan nama ‘HelloWCF’.
5. Pada Solution tersebut klik kanan, pilih menu Add -> New Project.



6. Pilih WCF pada *project template* di bagian kiri, lalu pilih WCF Service Application masukan textbox Name dengan HelloService. Sesudah itu klik OK.



7. Visual Studio akan menampilkan proyek HelloService pada tab Solution Explorer, kemudian buka file IService1.cs dan Service1.svc.
8. Hapus semua kode pada IService1.cs dan semua kode pada kode Service1.svc, sehingga Isi kedua *file* seperti di bawah ini.

```
namespace HelloService
{
    // NOTE: You can use the "Rename" command
    [ServiceContract]
    public interface IService1
    {
    }
}
```

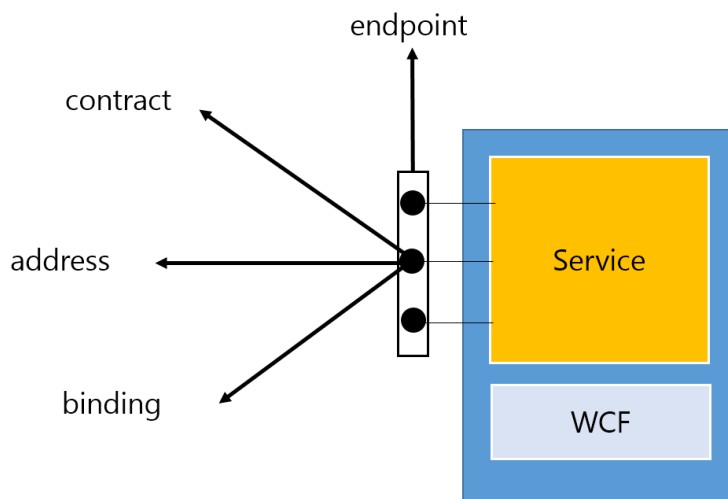
```
namespace HelloService
{
    // NOTE: You can use the "Rename" command
    // NOTE: In order to launch WCF Test Cli
    public class Service1 : IService1
    {
    }
}
```

[ServiceContract] merupakan merupakan sebuah implementasi kode dari konsep Service Contract. Service Contract merupakan bagian dari Contract pada WCF. Sebuah WCF Service merupakan layanan yang dipandang oleh pihak luar sebagai kumpulan endpoint. Lantas apa itu endpoint ?.

Endpoint adalah sebuah tempat atau lokasi dimana pesan akan disampaikan atau diterima atau keduanya. Endpoint terdiri atas ABC. ABC adalah singkatan dari

1. A = Address. Address ialah sebuah lokasi yang menentukan ke mana pesan akan dikirim atau address mendeskripsikan dimana endpoint berada. Address direpresentasikan sebagai URL (*Unified Resource Locator*).
2. B = Binding. Binding ialah spesifikasi dari mekanisme komunikasi yang digunakan oleh endpoint. Binding menjelaskan bagaimana operasi endpoint dapat dipanggil.
3. C = Contract. Contract ialah aturan tentang pesan apa saja yang dapat dikirim atau diterima (atau keduanya) dengan endpoint tersebut. Contract menjelaskan operasi yang dapat dipanggil menggunakan endpoint tersebut. Pada tutorial ini, Contract kita adalah IService1.

Sedangkan *service* pada sudut pandang WCF Service merupakan sebuah proses yang dapat menyimpan beberapa endpoint. Gambar di bawah ini bertujuan untuk memperjelas penjelasan di atas.



Service Contract merupakan komponen contract yang mengikat operation contract yang saling berhubungan satu sama lain ke dalam sebuah unit fungsional tunggal.

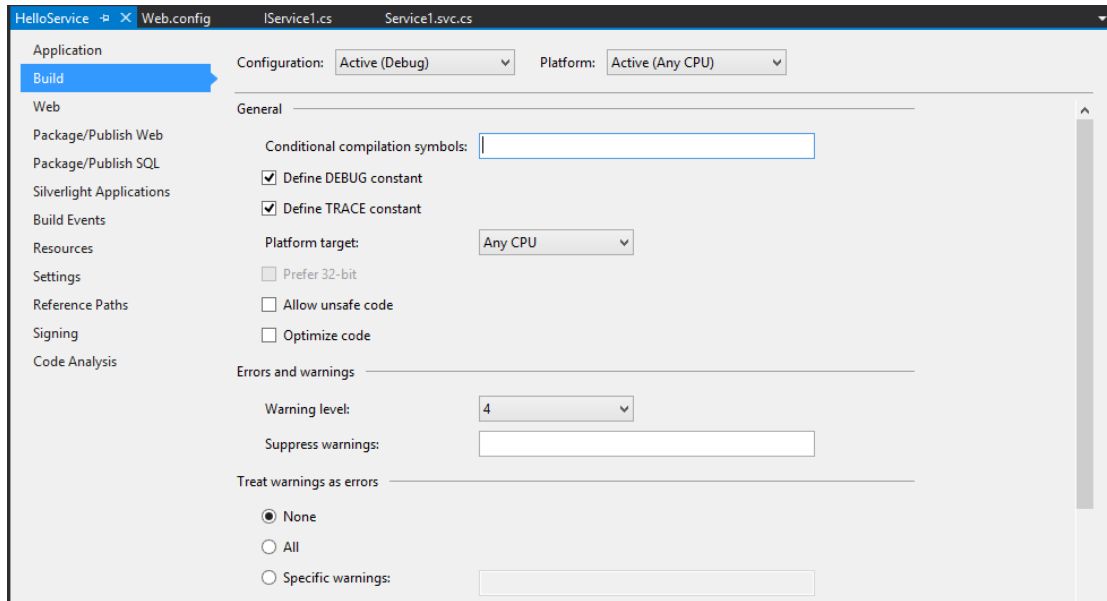
9. Setelah menghapus kode pada kedua *file* tersebut, klik F6 atau klik menu build -> Build Solution. Pastikan pada status bar bagian kiri bawah, Visual Studio menuliskan Build Succesed.
10. Ketikkan kode program di bawah ini pada interface IService1.

```
[OperationContract]
public interface IService1
{
    [OperationContract]
    string GetMessage();
}
```

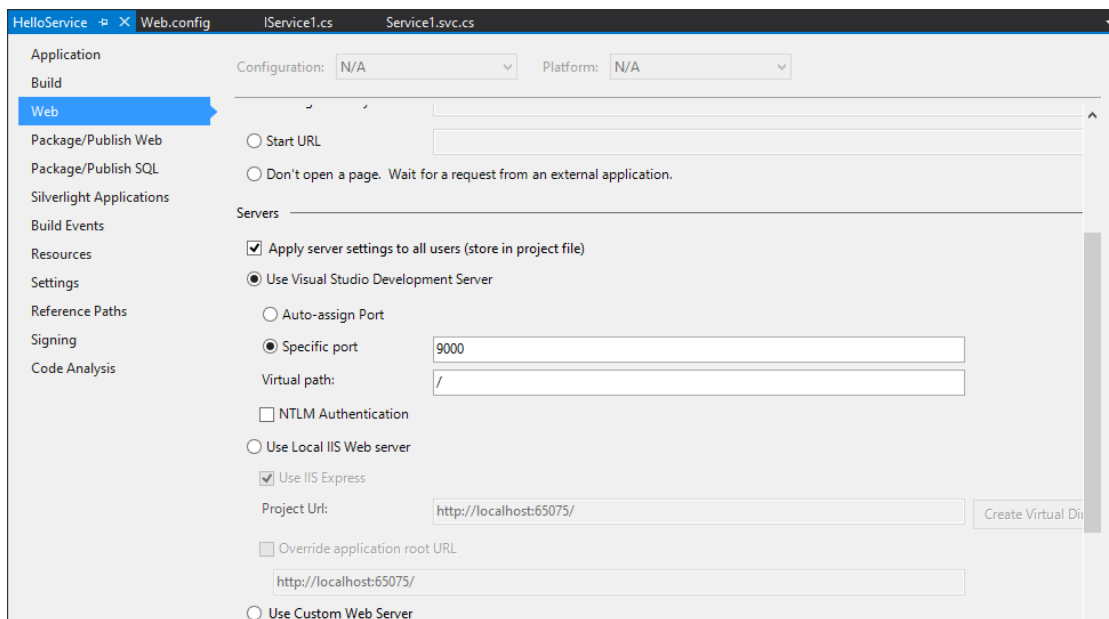
11. Ketikkan kode program di bawah ini pada class Service1.

```
public class Service1 : IService1
{
    public string GetMessage()
    {
        return "Hello World, ini adalah program WCF pertamaku";
    }
}
```

12. Klik F6 untuk memastikan tidak ada kesalahan pada kode program.
13. Kemudian klik kanan proyek HelloService pada Solution Explorer, pilih menu Properties.
 - . Maka Visual Studio akan menampilkan tab properti dari HelloService.



14. Pilih Web pada panel kiri, kemudian klik radiobutton Use Visual Studio Development Server, klik juga radiobutton Specific port dengan mengisi textbox di sampingnya dengan nilai 9000.



Konfigurasi ini dilakukan agar address dari endpoint service aplikasi kita terletak pada <http://localhost:9090/IService.svc>. Setelah konfigurasi dilakukan, Tekan ctrl+s agar konfigurasi benar-benar tersimpan. Lalu tekan kembali F6 untuk mem-build aplikasi.

15. Buka *file* Web.Config pada Solution Explorer, kemudian beri/tekan enter setelah kode `<system.serviceModel>`. Pemberian enter dimaksudkan untuk penambahan kode konfigurasi untuk endpoint *service* kita.

```
<services>
  <service name="HelloWCF.Service1">
    <endpoint binding="basicHttpBinding" //
      contract="HelloWCF.IService1" >

      <identity>
        <dns value="localhost"/>
      </identity>
    </endpoint>
  </service>
</services>
```

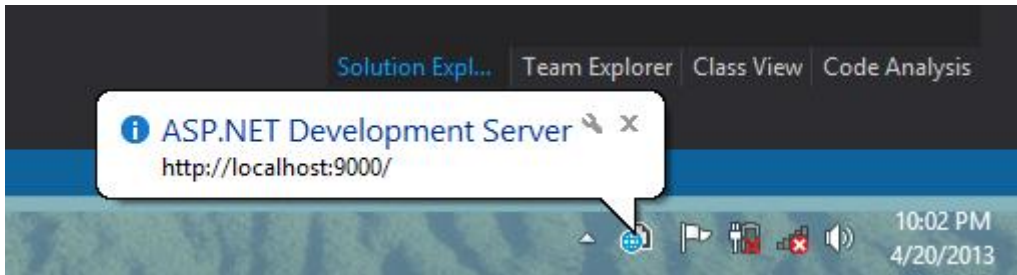
Web.Config merupakan *file* yang digunakan untuk memberikan konfigurasi pada proyek kita, khususnya aplikasi berbasis web. Pada *file* Web.Config, kita dapat menyimpan konfigurasi koneksi *database*, konfigurasi WCF Service, dan konfigurasi yang berhubungan dengan deployment aplikasi pada IIS web server.

Pada kode di atas, kita mendefinisikan sebuah *service* dengan nama HelloWCF.Service1. Pemberian nama ini sifatnya opsional, artinya Anda dapat sesuka hati memberikan namanya sesuai keinginan Anda. Namun lebih baik kita menamakan *service* tersebut sesuai dengan nama class yang berisikan implementasi dari contract kita (nama class pada Service1.svc). Di dalam tag *service*, kita mendefinisikan sebuah endpoint dengan cara menambahkan tag endpoint. Endpoint tersebut kita beri property binding dengan nilai basicHttpBinding, yang berarti bahwa *service* kita menggunakan komunikasi *web service* yang sederhana, tanpa adanya sekuriti pada *web service* tersebut dan basicHttpBinding menandakan bahwa *service* kita menggunakan protokol komunikasi SOAP (*Simple Object Access Protocol*). SOAP merupakan protokol komunikasi client server yang mengirim dan menerima pesan menggunakan protokol HTTP. Data yang dikirim dan diterima berupa XML.

Endpoint kita mendefinisikan property contract dengan nilai HelloWCF.IService1. Nilai ini tidak boleh sembarangan diisi, karena nilai dari property contract harus sama dengan nama interface yang kita deklarasikan dengan kode [ServiceContract]. Tag identity dan dns menyatakan bahwa *service* masih dalam kondisi pengembangan. Jika *service* kita ingin di-*hosting* pada sebuah *hosting provider* atau *service* sudah berada di tahap *deployment*, maka tag tersebut dapat dihapus.

16. Setelah kita memberikan konfigurasi, maka kita simpan konfigurasi tersebut dengan menekan tombol ctrl+s. Jika sudah, maka tekan F5 untuk mengetes *service* kita.

17. Pastikan terdapat notifikasi dari ASP.NET Development Server.



Notifikasi ini menunjukkan bahwa konfigurasi yang kita lakukan pada langkah 13 & 14 berhasil.

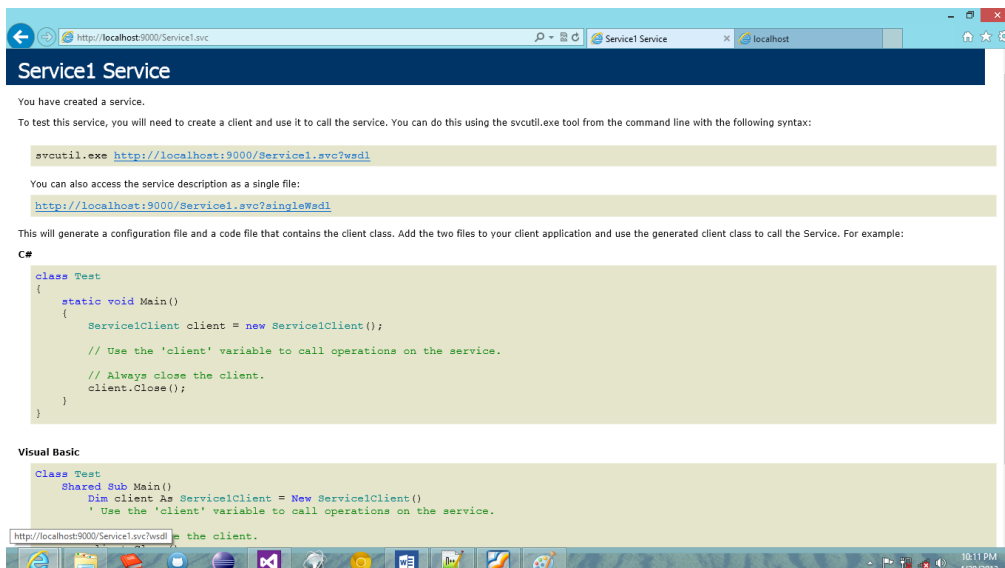
18. Maka secara otomatis *default web browser* Anda akan menampilkan informasi *file* seperti gambar di bawah ini.

Directory Listing -- /

Tuesday, April 16, 2013 11:40 AM	<dir> App_Data
Tuesday, April 16, 2013 01:35 PM	<dir> bin
Tuesday, April 16, 2013 02:03 PM	5,199 HelloService.csproj
Tuesday, April 16, 2013 02:03 PM	1,086 HelloService.csproj.user
Tuesday, April 16, 2013 01:39 PM	504 IService1.cs
Tuesday, April 16, 2013 11:40 AM	<dir> obj
Tuesday, April 16, 2013 11:40 AM	<dir> Properties
Tuesday, April 16, 2013 11:40 AM	109 Service1.svc
Tuesday, April 16, 2013 01:47 PM	714 Service1.svc.cs
Saturday, April 20, 2013 09:13 PM	1,426 Web.config
Tuesday, April 16, 2013 11:40 AM	1,299 Web.Debug.config
Tuesday, April 16, 2013 11:41 AM	1,360 Web.Release.config

Version Information: ASP.NET Development Server 11.0.0.0

19. Pilih Service1.svc pada *directory listing* tersebut, maka akan muncul tampilan di *web browser* seperti gambar di bawah ini.

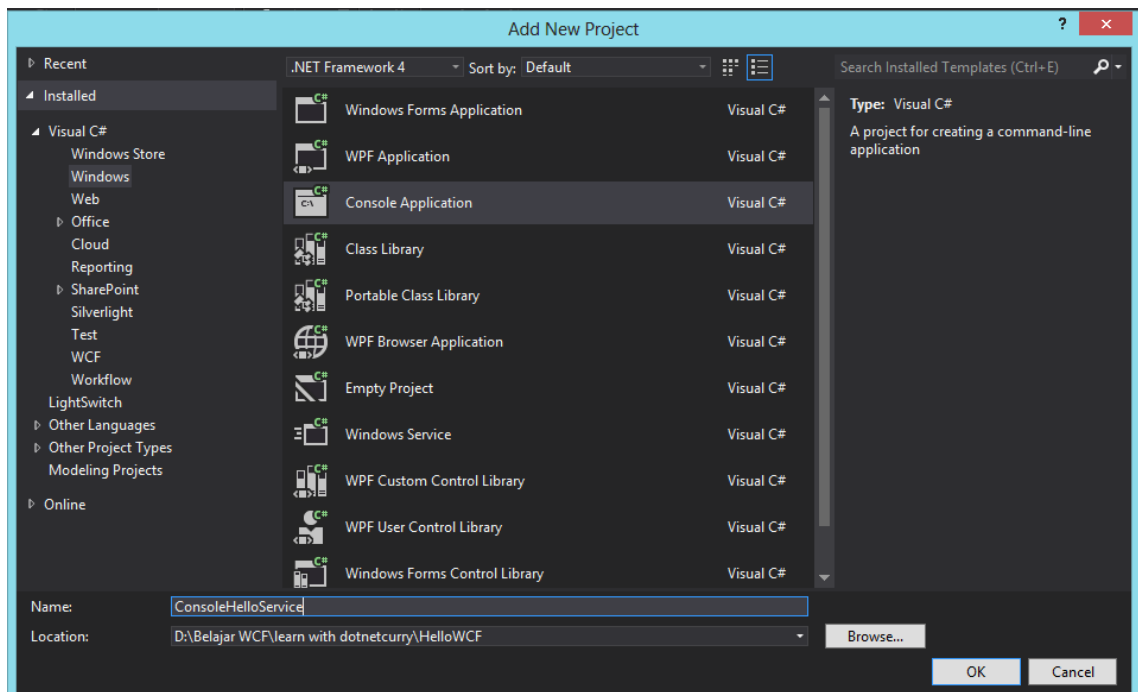


20. klik link ini <http://localhost:9000/Service1.svc?singleWsdL>, untuk menampilkan WSDL (Web Service Definition Language). WSDL merupakan jembatan antara *web service* dengan *developer* yang notabenehnya ingin menggunakan *web service* tersebut. Disarankan agar *web browser* tidak ditutup, hal ini untuk melakukan referensi *service* pada sesi 2.

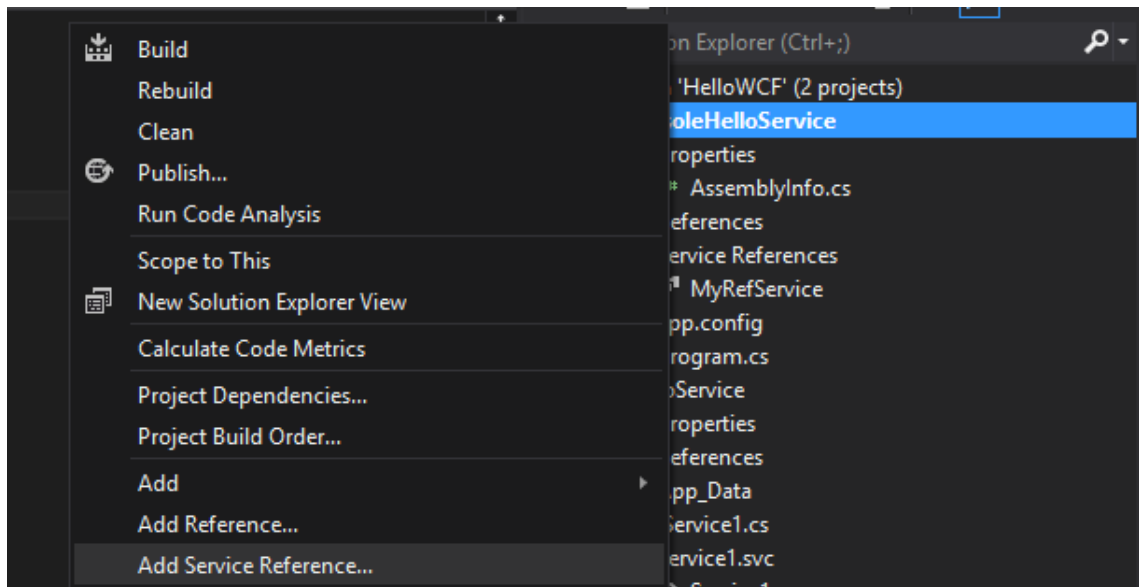
Explore The Code (Sesi 2)

Pada bagian ini, kita akan membuat aplikasi *client* dalam bentuk aplikasi console. Aplikasi ini melakukan aktivitas *consume service* pada WCF Service yang telah buat kita pada sesi 1.

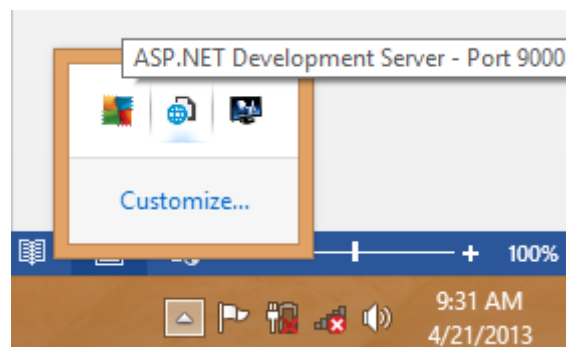
1. Klik kanan pada Solution 'HelloService' lalu pilih menu Add -> New Project. (Langkah ini sama seperti langkah ke-5 pada sesi 1).
2. Pada Window Add New Project, pilih project template tipe Windows pada panel kiri window, kemudian pilih Console Application dan beri nama aplikasi console tersebut dengan nama ConsoleHelloService.



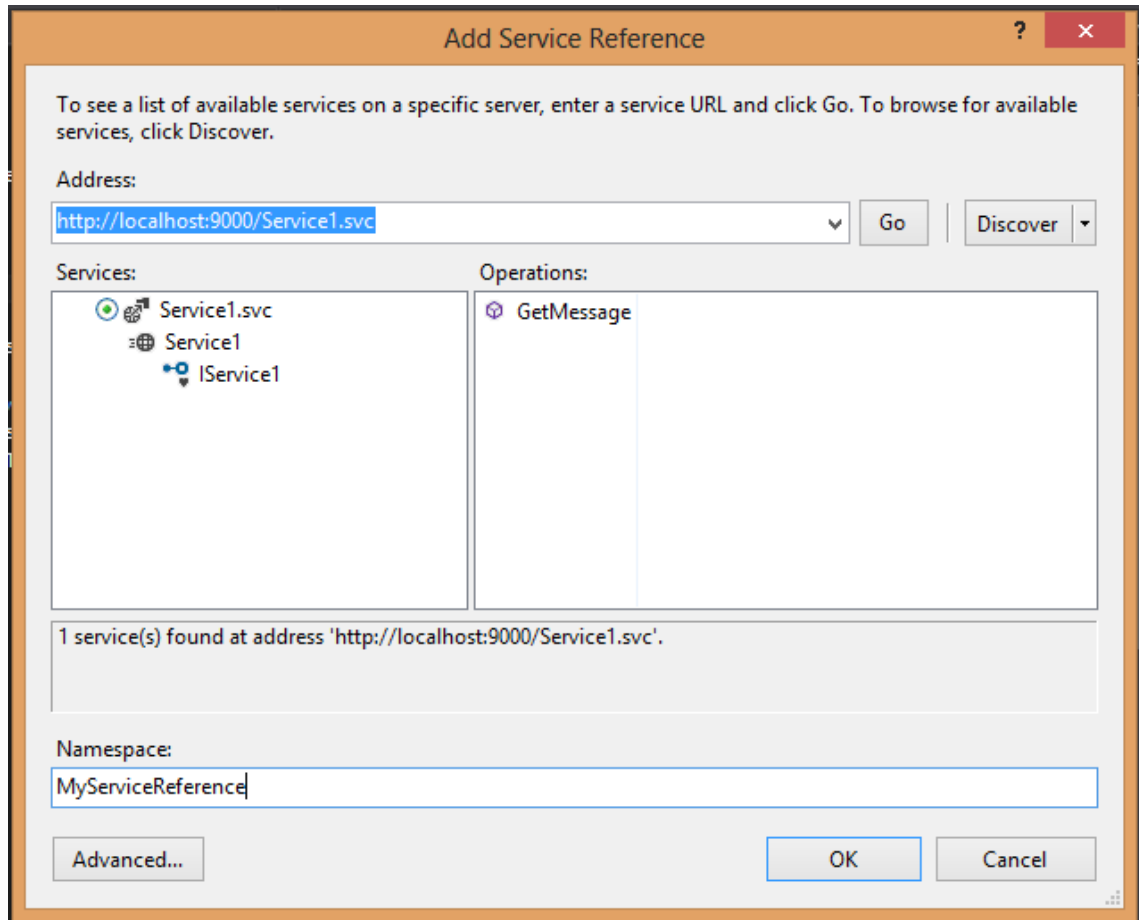
3. Setelah aplikasi terbentuk. Klik kanan proyek ConsoleHelloService pilih menu Add Service Reference untuk menambahkan service reference pada aplikasi kita. Setelah memilih menu tersebut maka Visual Studio menampilkan sebuah window Add Service Reference,



4. Pastikan WCF Service HelloService masih dalam keadaan run pada ASP.NET Development Server. Untuk memastikan apakah service masih berjalan, lihat pada Notification Area Icon dan perhatikan apakah ikon dari ASP.NET Development Server ada. Jika ikon ada seperti gambar di bawah maka HelloService masih berjalan.



5. Pada window Add Service Reference, masukan <http://localhost:9000/Service1.svc> pada textbox Address. Hal ini memberitakuan kepada Visual Studio mengenai Endpoint Address. Kemudian klik tombol Discover untuk men-*discover* HelloService. Sesudah itu maka Window akan menampilkan rincian HelloService.
6. Klik Service1.svc pada kolom Services. Maka akan ditampilkan Service1 beserta IService1 dimana keduanya menjelaskan implementasi *service* dan nama service contract kita.
7. Klik IService1 pada kolom Services, pastikan bahwa pada kolom Operations terdapat GetMessage.

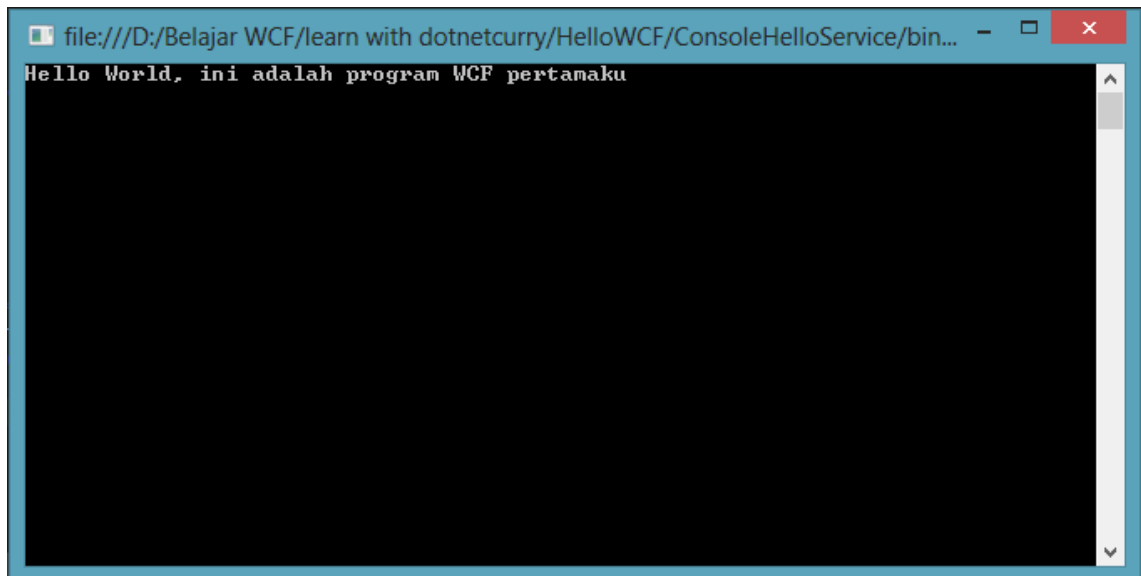


8. Ketik MyServiceReference pada textbox Namespace, lalu klik tombol OK. Visual Studio akan men-*generate* class untuk mengakses ke HelloService. Hal ini dilakukan agar para *developer* tidak perlu memfokuskan untuk bagaimana mengakses *service* tersebut secara rinci, namun fokus terhadap permasalahan bisnis aplikasi.
9. Buka file Main.cs pada ConsoleHelloService, lalu ketikkan kode di bawah ini.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using ConsoleHelloService.MyServiceReference;
6
7 namespace ConsoleHelloService
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Service1Client proxy = new Service1Client();
14            var result = proxy.GetMessage();
15            Console.WriteLine(result.ToString());
16            Console.ReadKey(true);
17        }
18    }
19 }
```

Pada baris ke-5, kita menambahkan namespace `ConsoleHelloService.MyServiceReference` agar kita dapat menggunakan class yang mengakses `HelloService`. Baris 13 memberitahukan kepada kita bahwa aplikasi kita akan mengakses `Service1` dengan cara melakukan instansiasi class `Service1Client`. Baris ke-14 kita memanggil operation contract `GetMessage` pada service contract `IService1` dan menyimpannya pada variabel `result` dengan tipe data varian. Pada baris 15, kita ingin menampilkan pesan yang berasal pada `HelloService` dan baris 16 untuk digunakan untuk menunggu inputan user sekaligus digunakan untuk menahan layar console.

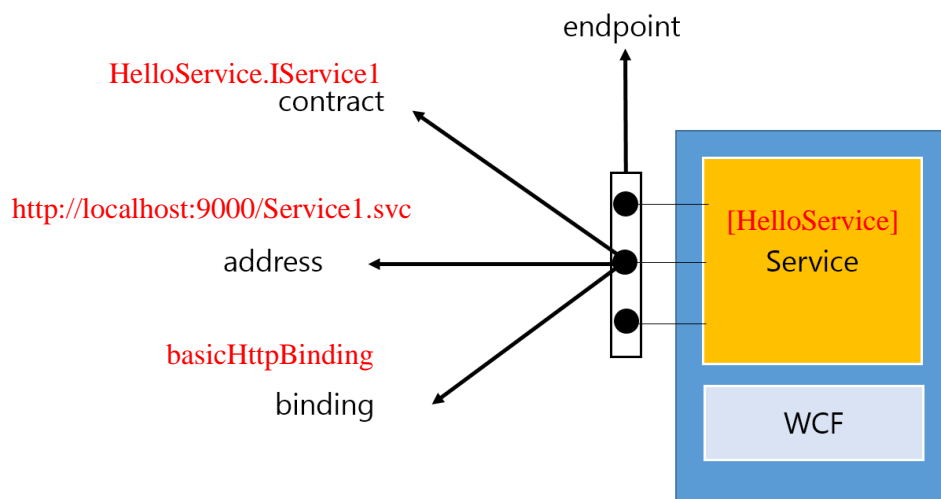
10. Klik kanan `ConsoleHelloService` pada `Solution Explorer` lalu pilih menu `Set as StartUp Project`. Hal ini dilakukan agar `ConsoleHelloService` agar dapat dijalankan karena sebelumnya default startup project adalah `HelloService`.
11. Jalankan aplikasi dengan tekan tombol `F5`, maka aplikasi console akan muncul dan menampilkan output "Hello World, ini adalah program WCF pertamaku".



Aplikasi console kita berhasil meng-*consume* HelloService. Selamat Anda telah berhasil membuatnya ☺.

Penutup

Pembuatan WCF Service beserta aplikasi client berhasil dibuat, sekarang saatnya kita membuat kesimpulan dari tutorial ini. Kesimpulan dari *service* kita yang lebih jelas digambarkan pada gambar di bawah ini. Pada gambar di bawah ini, Endpoint yang berperan sebagai tempat di mana *service* dapat diakses terdiri atas 3 komponen utama yaitu address, binding, dan contract (ABC).



Pada tutorial ini, penulis hanya memaparkan konsep dasar saja karena konsep dasar merupakan hal penting ketika ingin mengembangkan sesuatu yang lebih kompleks. Tutorial ini akan menjadi dasar pengembangan untuk tutorial-tutorial selanjutnya. Semoga tutorial ini bermanfaat bagi pembaca dan semoga tulisan ini dapat memajukan Teknologi Informasi Indonesia. Selamat berkarya.

Referensi

MSDN – The Microsoft Developer Network (www.msdn.microsoft.com).

Klien, Scott. *Professional WCF Programming: .NET Development with the Windows Communication Foundation..* United State of America: Worx publishing.

Biografi Penulis



Isman Subarkah. Lahir di Bogor, 22 Oktober 1991. Penulis telah menyelesaikan pendidikan vokasi di D3 Institut Pertanian Bogor (IPB) Angkatan 46 dengan program keahlian Manajemen Informatika. Sekarang Penulis melanjutkan studinya kembali di IPB pada Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam. Penulis memiliki *passion* dengan Teknologi Microsoft baik perangkat lunak maupun *programming*, namun di sisi lain penulis juga sangat senang dalam mengutak-atik teknologi GNU/Linux untuk administrasi jaringan & sistem dan Teknologi Java untuk bidang *programming*-nya. Pada tahun 2012,

Penulis pernah menjadi *Software Developer* selama 6 bulan di sebuah *software house* di Bogor (Internship). Teknologi yang pernah menjadi proyek penulis saat bekerja adalah Windows Presentation Foundation (WPF), Windows Phone 7, WinForm, ASP.NET & AJAX, Windows Communication Foundation (WCF). WCF Data Service (ODATA), ASP.NET MVC, ASP.NET, ASP.NET Web Service, PHP dan Android. Untuk saran ataupun kritik dapat pembaca layangkan melalui email penulis, yaitu ismansubarkah91@gmail.com.