

Sesi 2: Membuat Aplikasi Katalog Buku Mobile untuk Windows Mobile 5.0 dengan Visual Studio .NET 2005

Oleh: Narenda Wicaksono

Tujuan Umum:

Tujuan umum dari latihan ini adalah untuk mempelajari penggunaan Visual Studio .NET 2005 dan Windows Mobile 5.0 SDK dalam rangka mengembangkan aplikasi Windows Mobile 5.0 yang dapat memanfaatkan Web service, memanfaatkan Platform API piranti, serta menangani keadaan offline.

Teknologi yang Digunakan:

Microsoft Visual Studio .NET 2005
Microsoft .NET Framework 2.0
ASP.NET 2.0
Microsoft Visual C#
Microsoft .NET Compact Framework 2.0
Microsoft Windows Mobile 5.0 SDK

Prasyarat:

Akses ke web service "WSKatalog"
Koneksi Internet/layanan Internet Information System (IIS) dalam localhost

Penting:

Anda akan menggunakan layanan web service KatalogBukuWS dalam workshop ini. Jika Anda tidak mempunyai akses ke Internet atau tidak dapat mengakses web service tersebut, Anda dapat memasang web service WSKatalog dalam komputer pengembangan. Ikuti instruksi pada Lampiran D.

Catatan:

Agar latihan ini berjalan dengan baik, komputer tidak boleh di RAS ke Microsoft Network

Daftar Isi:

Pendahuluan
Bagian 1 : Membuat Form User Interface
Bagian 2 : Menggunakan Web Service
Bagian 3 : Menggunakan Platform API
Bagian 4 : Menangani Skenario Offline
Lampiran A : Instalasi Microsoft Loopback Adapter
Lampiran B : Instalasi Microsoft Virtual Machine Network Services Driver
Lampiran C : Instalasi Konektivitas Emulator
Lampiran D : Mengatur Web Service dalam komputer pengembangan
Lampiran E : Memeriksa Web Service

Pendahuluan

Target latihan ini adalah membiasakan diri dengan pengembangan aplikasi dalam mobile device, seperti Pocket PC, dengan menggunakan Microsoft Visual Studio .NET 2005.

Pertama-tama, Anda akan menggunakan Integrated Development Environment (IDE) Microsoft Visual Studio .NET untuk menciptakan form Microsoft Windows.

Selanjutnya, Anda akan meningkatkan kemampuan aplikasi dengan memanfaatkan Web Service untuk mengambil dataset yang mengandung katalog produk, mengakses platform API piranti, dan untuk melakukan skenario offline.

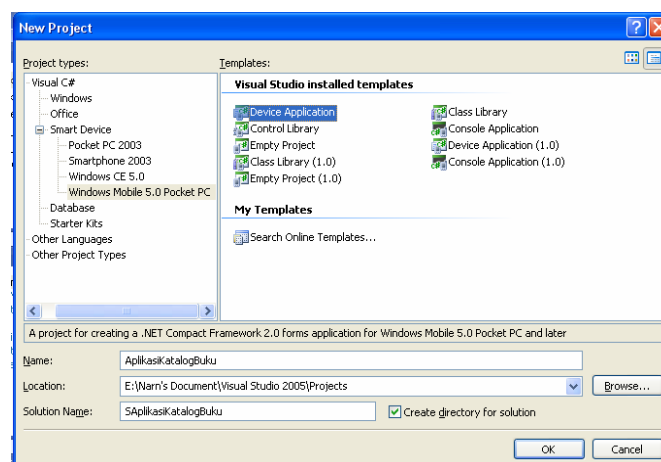
Bagian 1 : Membuat Form User Interface

Tinjauan Umum

Dalam bagian ini, Anda akan menciptakan form Windows menggunakan Microsoft Visual C#. Anda akan membuat project baru, menambahkan kontrol, mengatur properti image, kemudian menjalankan aplikasi dalam emulator Pocket PC.

Membuat Proyek

1. Mulai **Visual Studio 2005** dengan memilih **Start | All Programs | Microsoft Visual Studio 2005 | Microsoft Visual Studio 2005**.
2. Pilih perintah menu **File | New | Project...**
3. Dalam kotak dialog **New Project** buka entri **Visual C# | Smart Device | Windows Mobile 5.0 Pocket PC**.



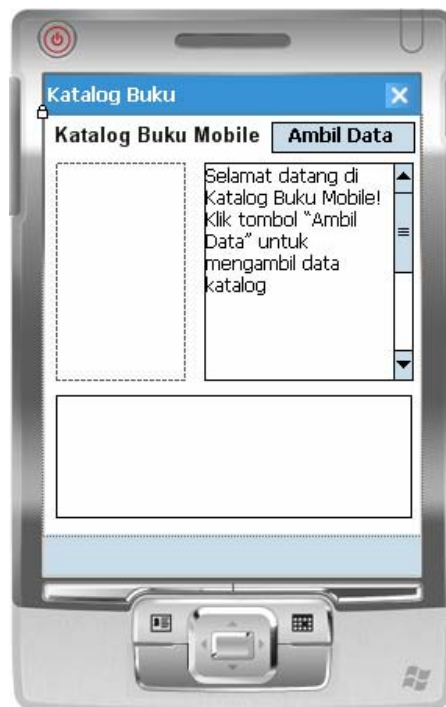
Gambar 1 Form Membuat Project Baru

4. Dalam **Templates**, pilih **Device Application**.
5. Ubah **Name** dari **DeviceApplication1** menjadi **AplikasiKatalogBuku**, lalu pilih lokasi untuk project Anda (bebas)
6. Ubah Solution Name menjadi **SaplikasiKatalogBuku**
7. Klik **OK**

Menambahkan Kontrol Pada Form

Pada bagian ini, Anda akan membuat form dengan susunan kontrol seperti yang dideskripsikan dalam Gambar 2

8. Pilih **Form1** pada **Designer Windows Forms**. Properti dari **Form1** akan ditampilkan pada jendela Properties
9. Pada bagian **Text** dalam jendela properties **Form1**, ketikkan
- 10.
11. Pada menu View, klik Toolbox
12. Drag sebuah kontrol Label dari tab **Common Device Control** ke bagian kiri atas form
13. Ubah nilai properti label tersebut sebagai berikut:
 - a. **Text** : Katalog Buku Mobile
 - b. **Font** : Arial, 10pt, style=bold
14. Ubah ukuran label sehingga seluruh isi text terlihat
15. Drag sebuah kontrol **Button** dari tab **Common Device Control** ke bagian kanan atas form
16. Ubah properti **Text** dari **Button** tersebut menjadi **Ambil Data**
17. Drag sebuah kontrol **PictureBox** dari tab **Common Device Control**
18. Ubah properti **Size** PictureBox menjadi 83, 140
19. Posisikan **PictureBox** di sisi kiri form
20. Drag sebuah kontrol **TextBox** dari tab **Common Device Controls** ke bagian kanan atas form.



Gambar 2 Susunan User Interface Aplikasi Katalog Buku Mobile

21. Berikan nilai properti **TextBox** seperti berikut:

- a. **ReadOnly** : True
- b. **Multiline** : True
- c. **ScrollBars** : Vertical
- d. **Text** : Selamat datang di Katalog Buku Mobile! Klik tombol "Ambil Data" untuk mengambil data katalog

22. Drag sebuah kontrol **ListView** dari tab **Common Device Control** ke bagian bawah form, sesuaikan ukurannya.

Mengatur properti image

23. Pada properti **pictureBox1**, ubah nilai **SizeMode** menjadi **StretchImage**

24. Pada menu View, pilih **Solution Explorer**.

25. Dalam Solution Explorer, klik kanan pada project

AplikasiKatalogBuku, klik **Add**, dan pilih **Add Existing Item**.

26. Pada bagian bawah dari dialog **Add Existing Item**, ubah nilai **Files of type** menjadi **Image Files**

27. Buka folder **Katalog Buku\images** dan pilih file **icon.png**
28. Pada jendela Properties, ubah properti **Build Action** ke nilai **Embedded Resource**. Aksi ini akan mengembed file ke output dari project utama
29. Pada Solution Explorer, klik kanan pada **Form1**, lalu pilih **View Code**
30. Tambahkan kode berikut pada references yang ada di bagian atas kode

```
using System.IO;
```

31. Pada properti **pictureBox1**, klik **Image**, lalu klik kotak kecil di kanan properti **Image**.



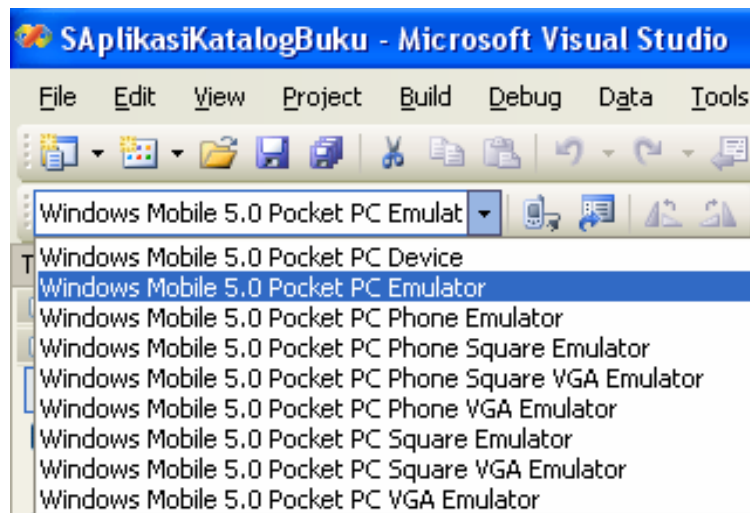
32. Kemudian akan muncul kotak **Open File Dialog**, pilih file **icon.png**. Aksi ini akan langsung secara otomatis memasukkan file **icon.png** pada **pictureBox1** (gambar dari file **icon.png** dapat langsung terlihat di form).



Gambar 3 Tampilan pictureBox1 setelah diberi image

Menjalankan aplikasi

33. Pada menu File, klik **Save All**
34. Pada toolbar **Device Extensions**, pilih **emulator Pocket PC** di dalam kotak **Deployment Device** seperti yang ditunjukkan pada Gambar 3



Gambar 4 Tampilan Pemilihan Emulator

35. Tekan tombol **CTRL + SHIFT + B** untuk mengkompilasi
36. Tekan tombol **CTRL + F5** untuk menjalankan aplikasi ini

Bagian 2: Menggunakan Web Service

Tinjauan Umum:

Pada latihan ini, Anda akan menambahkan suatu Web reference ke dalam project dan mengambil data dari Web Service. Ketika Anda menambahkan Web reference ke dalam suatu project, VS .NET 2005 akan membuat suatu kelas proxy yang mengkapsulasi logic yang digunakan untuk mengakses Web Service menggunakan protokol HTTP. Untuk memanfaatkan Web Service, aplikasi Anda cukup menggunakan method dari proxy class tersebut. Lampiran C akan memberikan gambaran mengenai cara kerja Web Services dari perspektif client.

Pada prosedur selanjutnya, Anda akan membangun logic yang mengakses Web service dan menangani response. Web Services akan mengembalikan DataSet ke aplikasi sebagai response XML dengan protokol HTTP. Kelas yang sebelumnya ditambahkan dalam aplikasi ketika Anda menambahkan Web Reference akan melakukan parsing terhadap response XML dan menciptakan copy dari dataset di dalam aplikasi.

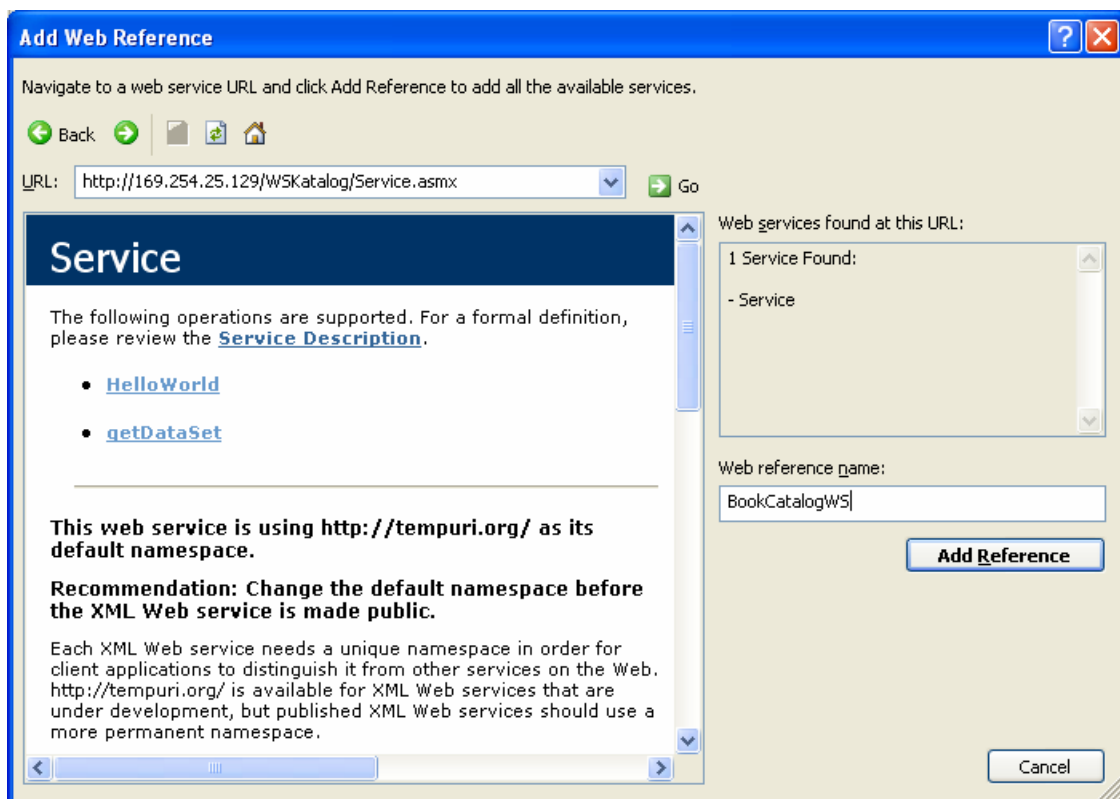
Pada bagian selanjutnya, Anda akan menggunakan DataSet yang dikembalikan Web Service untuk mempopulasikan kontrol ListBox dengan mengambil DataTable yang ada di dalamnya dan melakukan iterasi pada DataRow. Aplikasi lalu akan mengambil sebuah DataRow (ekivalen dengan

suatu kolom di dalam baris) dari DataRow tertentu untuk dimasukkan ke kontrol ListBox. Salah satu kolom yang dikembalikan Web Service bertipe array of byte yang merepresentasikan gambar yang berasosiasi dengan DataRow (gambar cover buku). Ketika user memilih DataRow tertentu dari kontrol ListBox, aplikasi akan mencari DataRow spesifik di dalam Dataset, mengambil array of byte untuk gambar, dan membuat Bitmap untuk ditampilkan dalam kontrol PictureBox. Pada akhir latihan ini, Anda akan menambahkan fungsi untuk menyimpan Dataset secara lokal di dalam piranti sehingga user dapat mengakses Dataset ketika tidak terhubung ke jaringan.

Menambahkan Web Reference ke dalam Project

37. Klik kanan **References** pada Solution Explorer
38. Klik **Add Web Reference**
39. Dalam jendela **Add Web Reference** pada kotak **Address**, ketikkan lokasi URL web service, misalnya:

<http://169.254.25.129/WSKatalog/Service.asmx>



Gambar 5 Menambahkan web reference

40. Ubah nama reference menjadi **BookCatalogWS**
41. Klik **Add Reference**

Mengakses Web Service

42. Pada Solution Explorer, klik kanan **Form1** lalu pilih **View Code**

43. Tambahkan kode berikut pada bagian atas file

```
using System.Net;
```

44. Tambahkan anggota berikut pada kelas **Form1**. Aksi ini akan menciptakan instance dari kelas proxy, DataSet, dan DataTable.

```
private DataSet BookCatalogDS;  
private DataTable BookCatalogTable;  
private BookCatalogWS.Service ws = new BookCatalogWS.Service();
```

45. Pada Solution Explorer, klik kanan pada **Form1**, lalu pilih **View Designer**. Doppel klik tombol **Ambil Data** untuk menciptakan event handler kosong untuk event **Click**.

Mengambil Data dari Web service

46. Panggil method **getDataSet()** dengan menambahkan kode berikut:

```
private void button1_Click(object sender, EventArgs e) {  
    try  
    {  
        BookCatalogDS = ws.getDataSet();  
        BookCatalogTable = BookCatalogDS.Tables["BUKU"];  
        AddDataToListView();  
    }  
    catch (WebException we)  
    {  
        MessageBox.Show("Gagal koneksi. Error: " + we.Message,  
            "Koneksi Gagal");  
    }  
}
```

Loading data dalam kontrol ListView

47. Tambahkan method yang diberi nama **AddDataToListView()** untuk meload data

```
private void AddDataToListView() {  
    ListViewItem item;  
    listView1.Clear();  
    listView1.Columns.Add("Judul", 234, HorizontalAlignment.Left);  
    listView1.View = View.Details;  
  
    foreach(DataRow row in BookCatalogTable.Rows) {  
        item = new ListViewItem(row["JUDUL"].ToString());  
        listView1.Items.Add(item);  
    }  
}
```

```

        textBox1.Text = "Klik pada judul buku untuk melihat keterangan
                        dan gambar";
    }

```

Event untuk merubah keterangan dan image dari buku yang dipilih

48. Dari Solution Explorer, klik kanan **Form1**, klik **View Designer**
49. Dobel klik **ListView1** untuk menciptakan event handler kosong untuk event **SelectedIndexChanged**.
50. Dalam event handler ini, pastikan kontrol PictureBox diupdate dengan gambar yang benar dari dataset dengan mengkonstruksi MemoryStream dari array byte yang disimpan dalam dataset untuk image tersebut. MemoryStream kemudian digunakan untuk mengkonstruksi objek yang dapat ditampilkan oleh PictureBox sebagai sebuah image
51. Tambahkan kode berikut ini dalam method **SelectedIndexChanged** event handler

```

private void listView1_SelectedIndexChanged(object sender, EventArgs e) {
    DataRow row;
    if (listView1.SelectedIndices.Count > 0)
    {
        row = BookCatalogTable.Rows[listView1.SelectedIndices[0]];
        textBox1.Text = "Deskripsi:\r\n" +
row["KETERANGAN"].ToString();
        try
        {
            pictureBox1.Image = new Bitmap(new
MemoryStream((byte[])row["GAMBAR"]));
        }
        catch (System.InvalidCastException ne)
        {
            MessageBox.Show("Tidak bisa meload gambar. Error: " +
ne.Message);
        }
    }
}

```

52. Tekan tombol **CTRL + SHIFT + B** untuk mengkompilasi
53. Jalankan aplikasi dengan **CTRL+F5**. Tekan button **Ambil Data** untuk menguji pemanggilan Web service oleh client dan amati hasilnya.

Bagian 3: Mengakses API Platform untuk Device

Tinjauan Umum:

Pada bagian ini, Anda akan menambahkan kursor tunggu untuk Feedback secara Visual. Dalam latihan ini, Anda akan menggunakan Platform Invoke Services untuk memanggil fungsi DLL unmanaged yang akan menambahkan feedback visual ke aplikasi dalam bentuk kursor tunggu. Proses ini dapat diringkas, yaitu: mengimport layanan interoperabilitas, mendeklarasikan fungsi yang mendefinisikan kursor tunggu, dan membungkus fungsionalitas agar memudahkan penggunaan kursor tunggu.

Menambahkan kursor tunggu ke dalam aplikasi

54. Dari Solution Explorer, klik kanan pada **Form1**, lalu klik **View Code**

55. Tambahkan kode ini pada bagian atas file

```
using System.Runtime.InteropServices;
```

56. Tambahkan deklarasi dibawah ini pada kelas **Form1** untuk mendefinisikan tipe kursor

```
private static int hourGlassCursorID = 32514;
```

57. Deklarasikan fungsi unmanaged DLL untuk memberitahukan compiler bahwa implementasi datang dari **coredll.dll**. Gunakan atribut **DllImport**

```
[DllImport("coredll.dll")]  
public static extern int LoadCursor (int zeroValue, int cursorID);  
[DllImport("coredll.dll")]  
public static extern int SetCursor(int cursorHandle);
```

58. Tambahkan method berikut pada kelas **Form1** untuk mengaktifkan tampilan kursor tunggu

```
private static void ShowWaitCursor (bool value) {  
    SetCursor (value ? LoadCursor(0, hourGlassCursorID) : 0);  
}
```

59. Tambahkan kode **ShowWaitCursor(true)** untuk mengaktifkan kursor tunggu pada waktu mengakses web service. Contohnya pada method **Button1_Click**

```
private void button1_Click(object sender, EventArgs e) {  
    ShowWaitCursor(true);  
    try {  
        BookCatalogDS = ws.GetItems();  
        BookCatalogTable = BookCatalogDS.Tables["BUKU"];  
    }
```

```
        AddDataToListView();
    }
    catch (WebException we)
    {
        MessageBox.Show("Gagal koneksi. Error: " + we.Message,
            "Koneksi Gagal");
    }
    ShowWaitCursor(false);
}
```

60. Tekan tombol **CTRL + SHIFT + B** untuk mengkompilasi

61. Tekan tombol **CTRL + F5** untuk menjalankan aplikasi ini

Bagian 4: Menangani Skenario Offline

Tinjauan Umum:

Pada latihan ini, Anda akan menyimpan dataset ke sebuah file dan mengambil dataset yang tersimpan dalam sebuah file

Dataset yang diambil dari Web Service akan disimpan, sehingga dapat dilihat secara offline (saat membuka aplikasi selanjutnya tidak perlu online lagi). Dataset akan disimpan dalam sebuah file XML yang disimpan di dalam local storage piranti ketika aplikasi ditutup. Ketika aplikasi dibuka kembali, dataset akan diload dari file XML yang telah tersimpan sebelumnya dalam local storage piranti.

Menyimpan Dataset ke File

62. Dari Solution Explorer, klik-kanan Form1, lalu pilih **View Code**

63. Tambahkan kode seperti berikut:

```
using System.Xml;
```

64. Tambahkan kode untuk referensi lokasi file XML

```
private static string DataDirectory =
@"\Program Files\AplikasiKatalogBuku\Data\";
private static string DataFile =
@"\Program Files\AplikasiKatalogBuku\Data\Katalog.xml";
```

65. Dari Solution Explorer, klik kanan **Form1**, klik **View Designer**

66. Klik menu paling bawah untuk menambahkan menu, ketik **Simpan Data**, kemudian **enter**



67. Double klik menu item **Simpan Data**, kemudian tambahkan kode berikut ini:

```
private void menuItem1_Click(object sender, EventArgs e) {
    ShowWaitCursor(true);
    DirectoryInfo dir;
    FileInfo CatalogFile;
    XmlWriter Writer;

    dir = new DirectoryInfo(DataDirectory);
    if (!dir.Exists) {
        dir.Create();
    }

    CatalogFile = new FileInfo(DataFile);
    if (CatalogFile.Exists) {
        CatalogFile.Delete();
    }

    if (BookCatalogDS.Tables.Count != 0) {
        Writer = new XmlTextWriter(DataFile,
            System.Text.Encoding.Unicode);
        BookCatalogDS.WriteXml(Writer, XmlWriteMode.WriteSchema);
        Writer.Close();
        MessageBox.Show("Menyimpan data sukses");
    }
    ShowWaitCursor(false);
}
```

Meload Dataset dari File

68. Tambahkan method **LoadCatalogFromFile** untuk meload dataset dari file

```
private void LoadCatalogFromFile() {
    FileInfo CatalogFile = new FileInfo(DataFile);
    if (CatalogFile.Exists) {
        try {
            BookCatalogDS = new DataSet();
            BookCatalogDS.ReadXml(DataFile);
            BookCatalogTable = BookCatalogDS.Tables["BUKU"];
        }
        catch (Exception ex) {
            MessageBox.Show(ex.Message);
        }
        AddDataToListView();
    }
}
```

69. Tambahkan pemanggilan method **LoadCatalogFromFile** ketika form diload

```
private void Form1_Load(object sender, EventArgs e){
    LoadCatalogFromFile();
}
```

70. Tekan tombol **CTRL + SHIFT + B** untuk mengkompilasi

71. Tekan tombol **CTRL + F5** untuk menjalankan aplikasi ini

Kesimpulan:

Selamat! Anda sudah menyelesaikan seluruh materi workshop. Pada materi sesi II ini, yang Anda pelajari adalah:

1. Membuat Forms User Interface
2. Menggunakan Web Service
3. Menggunakan Platform API
4. Menangani Skenario Offline

Lampiran A: Instalasi Microsoft Loopback Adapter

Jika komputer Anda tidak memiliki koneksi ke jaringan, Anda harus melakukan instalasi Microsoft Loopback Adapter untuk mensimulasikan koneksi yang aktif. Langkah-langkah berikut mengasumsikan komputer Anda berjalan dengan Windows XP Service Pack 2

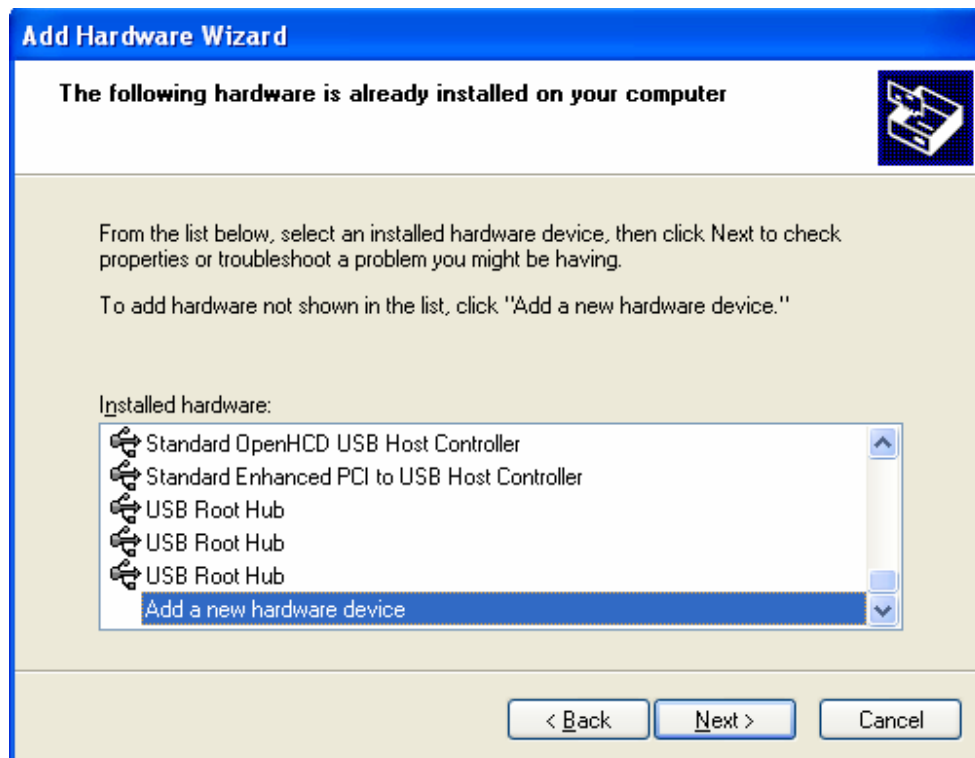
Menambahkan adapter

1. Buka **Control Panel**, klik **Add Hardware**

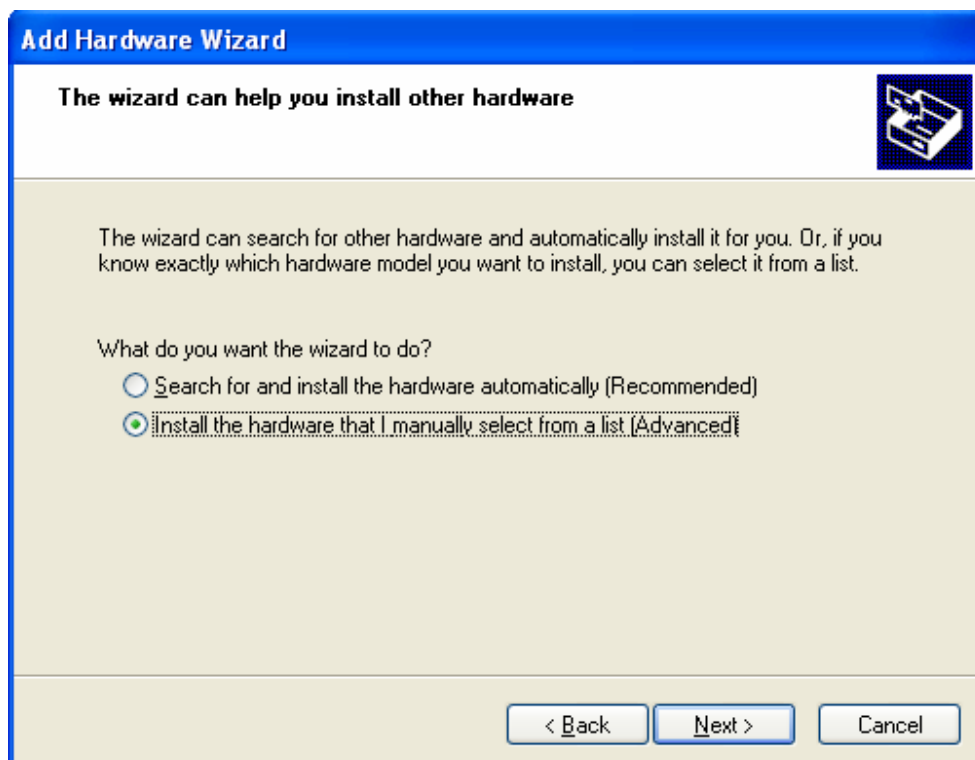
2. Pada **Add Hardware Wizard**, klik **Next** sampai Anda melihat **Is the hardware connected?** Pilih **Yes, I have already connected the hardware**, lalu klik **Next**.



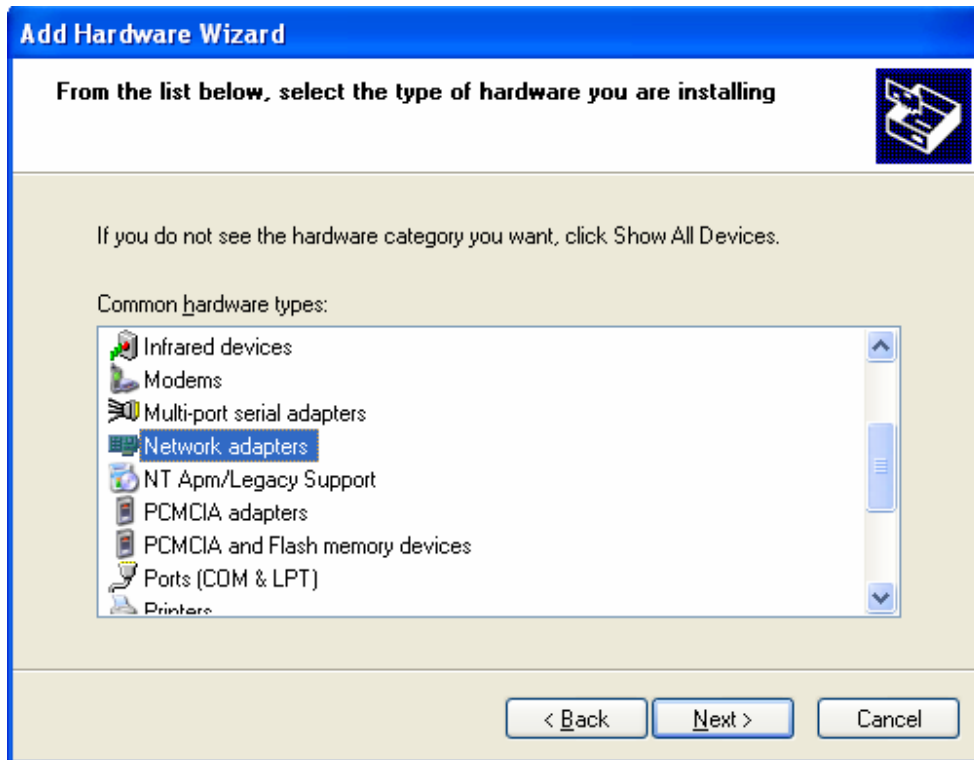
3. Pada **Installed Hardware list**, pilih **Add a new hardware device**, lalu klik **Next**.



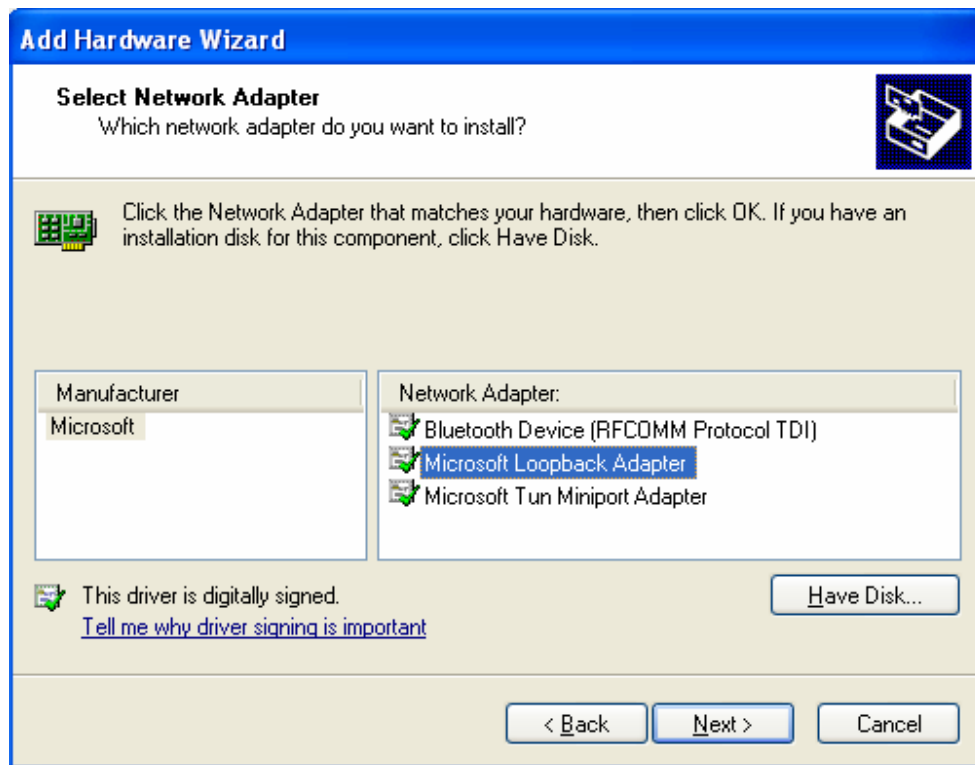
4. Pilih **Install the hardware that I manually select from a list (Advanced)**, lalu klik **Next**.



5. Pada list **Common hardware types**, pilih **Network Adapters**, lalu klik **Next**.



6. Pada list **Manufacturer**, pilih **Microsoft**. Pada list **Network Adapter**, pilih **Microsoft Loopback Adapter**, lalu klik **Next**.



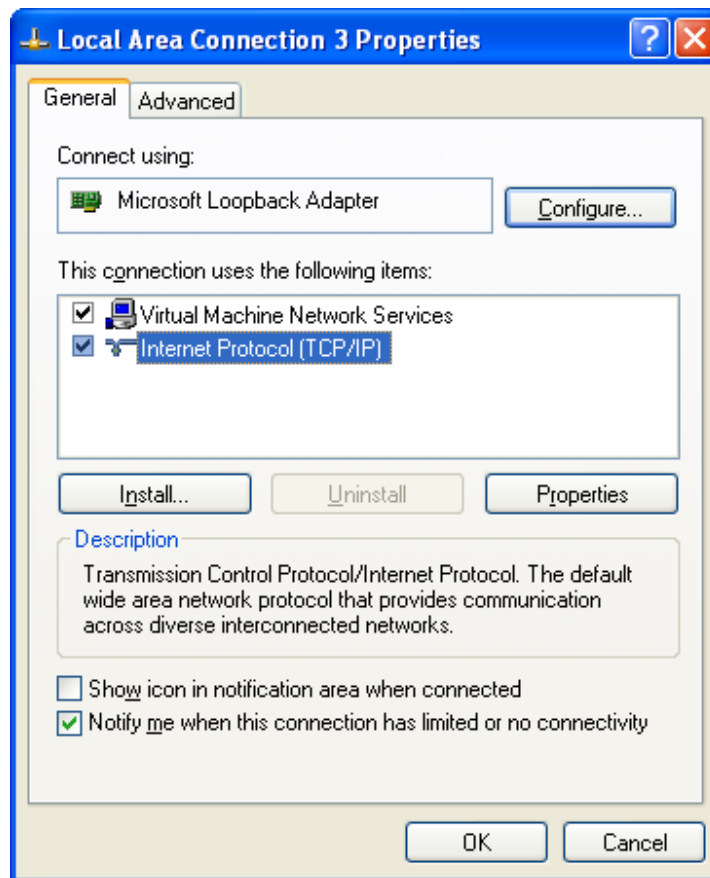
7. Terus klik **Next** pada **Add Hardware Wizard** selama instalasi Microsoft Loopback Adapter.
8. Restart komputer jika diperintahkan

Lampiran B : Instalasi Microsoft Virtual Machine Network Services Driver

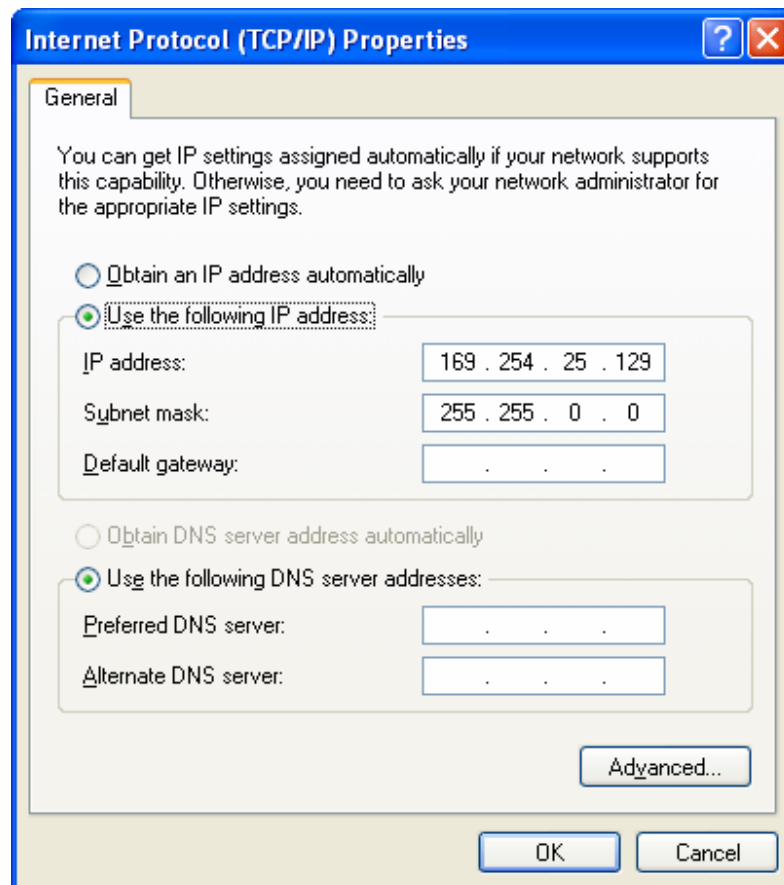
Setelah melakukan instalasi Microsoft Loopback Adapter, emulator membutuhkan virtual switch driver agar adapter tersebut dapat digunakan. Cara melakukan instalasi klik file **netsvwrap.msi** yang disertakan pada workshop ini dan ikuti petunjuk yang ada.

Konfigurasi pada Microsoft Loopback Adapter

1. Buka **Control Panel**, klik **Network Connections**
2. Klik kanan pada koneksi **Microsoft Loopback Adapter**, pilih **properties**
3. Klik pada **Internet Protocol (TCP/IP)**, kemudian klik **Properties**



4. Klik radio button **Use the following IP address**



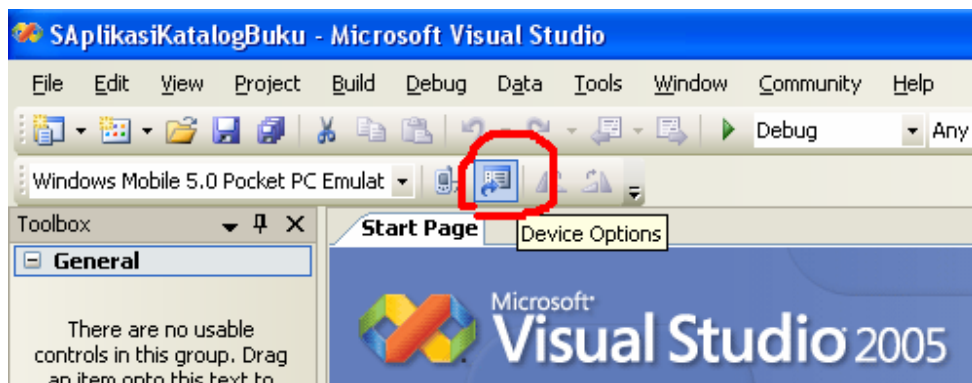
5. Isikan field **IP address** dengan alamat **169.254.25.129** dan field **Subnet mask** dengan alamat **255.255.0.0**
6. Klik **OK**, kemudian klik **OK** lagi
7. Klik kanan pada koneksi **Microsoft Loopback Adapter** di menu **Network Connection**, pilih **Bridge Connection**
8. Pastikan koneksi berada dalam kondisi **Connected**



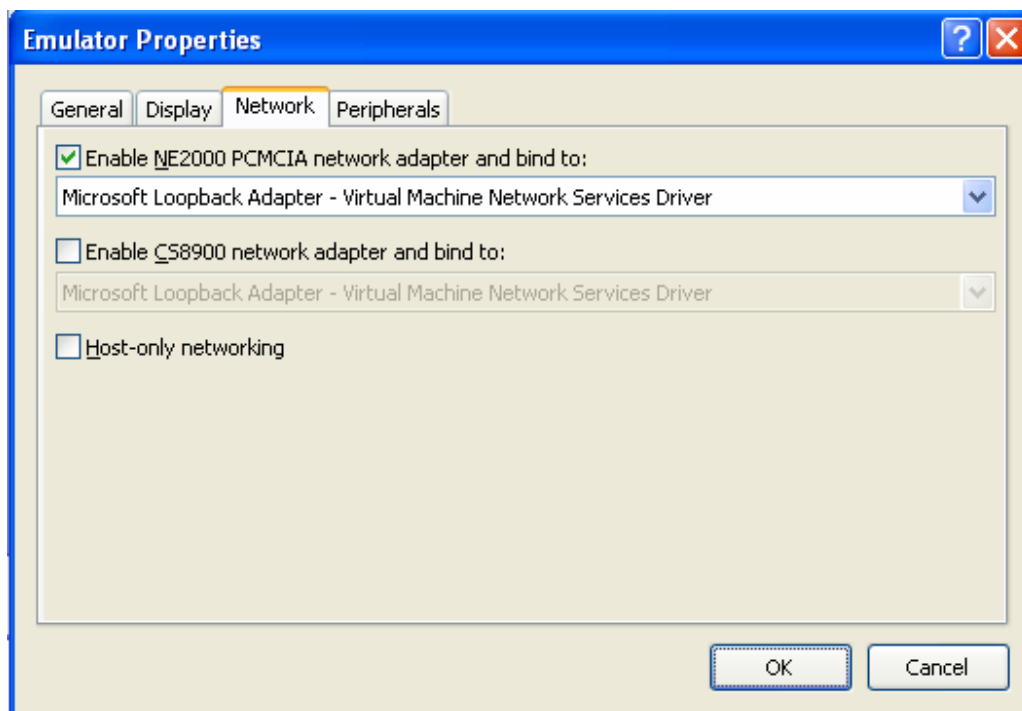
Lampiran C : Instalasi Konektivitas Emulator

Setelah melakukan instalasi **Microsoft Loopback Adapter** dan **Microsoft Virtual Switch Driver**, lakukan konfigurasi sebagai berikut agar emulator dapat mengakses **localhost** komputer development.

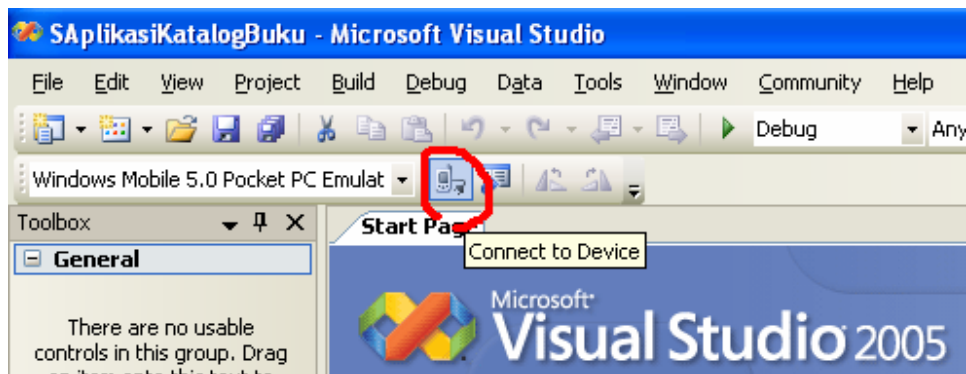
1. Klik **Device Option** pada toolbar **Visual Studio 2005**



2. Pilih **Windows Mobile 5.0 Pocket PC Emulator** lalu klik **Properties**
3. Pilih **Emulator Option**
4. Klik tab **Network**, kemudian checklist **Enable NE2000 PCMCIA network adapter and bind to:** pilih **Microsoft Loopback Adapter – Virtual Machine Network Services Driver.**



5. Klik **Connect to Device** pada toolbar **Visual Studio 2005**



Konfigurasi Network Card pada emulator

6. Setelah emulator tampil, klik **Start | Setting | Connections | Network Cards**
7. Pilih **NE2000 Compatible Ethernet Driver**
8. Pilih **Use server-assigned IP address**, lalu klik OK.
9. Reset emulator, pilih **File | Reset | Soft**
10. Setelah emulator ter-reset, klik **Start | Setting | Connections | Connections**
11. Klik tab **Advanced**, kemudian pilih button **Select Networks**
12. Ganti **combo box "Program that automatically connect to the Internet should connect using"** dengan **My Work Network**, klik **OK**

Menguji konektivitas emulator

1. Buka **Pocket Internet Explorer** pada emulator.
2. Jika address bar tidak terlihat, klik menu **View**, lalu klik **Address Bar**
3. Ketikkan alamat Web service yang terdapat pada localhost komputer pengembangan, misalnya
<http://169.254.25.129/WSKatalog/Service.asmx>
4. Jika halaman Web service tampil dengan benar, konektivitas jaringan pada emulator bekerja dengan baik, jika tidak maka akan ada pesan tampilan kesalahan.

Lampiran D: Mengatur Web Service WSKatalog pada Komputer Pengembangan

Jika Anda tidak mempunyai akses ke Internet atau tidak dapat mengakses Web service untuk alasan lainnya, Anda dapat memasang WSKatalog pada komputer pengembangan.

Untuk membuat Web service WSKatalog

1. Buka Microsoft Visual Studio .NET 2005
2. Dalam menu **File**, klik **New**, lalu klik **Web Site**
3. Pada panel tipe project, pilih **ASP.NET Web Service**
4. Pada ComboBox **Location**, pilih HTTP
5. Ubah nama location dengan lokasi yang dikehendaki. Misal :
<http://localhost/WSKatalog>
6. Pada ComboBox **Language**, pilih Visual C#
7. Klik **OK**. Akan terbentuk project Web service

Untuk menambahkan file kode

1. Pada Solution Explorer, klik-kanan project Web Service. Klik **Add**, dan klik **Add Existing Item**
2. Cari lokasi file Web service di **Katalog Buku\Service** pilih **Service1.asmx** dan **Service1.asmx.cs**. Jika salah satu file tidak ada, pilih **All Files**. Pemilihan All Files akan menampilkan segala jenis file di folder

Untuk menambahkan reference ke System.Drawing

1. Pada Solution Explorer, klik-kanan **Project**, Klik **Add Reference**
2. Klik tab .NET. Dalam list, cari item yang bernama **System.Drawing.dll**
3. Pilih **System.Drawing.dll** dan klik **OK**
4. Cari lokasi file Web service di \Service\, pilih **Service1.asmx** dan **Service1.asmx.cs**. Jika salah satu file tidak ada, pilih **All Files**

Untuk menguji Web Service BookCatalogWS

1. Tekan **CTRL + F5** untuk melakukan debug dan mendeploy web service. Browser web akan membuka halaman Service1.
2. Pada jendela browser, klik hyperlink **getDataSet**. Web browser akan membuka halaman web yang menunjukkan operasi yang ada pada Web Service dan pilihan untuk menguji Web Service
3. Klik **Invoke** untuk menguji Web service. Anda seharusnya bisa melihat jendela baru yang membuka invokasi Web service. Di dalam jendela baru tersebut ada XML yang merepresentasikan XML Dataset yang dikembalikan
4. Tutup semua jendela browser
5. Save seluruh perubahan dengan menu **File**, lalu **Save All**

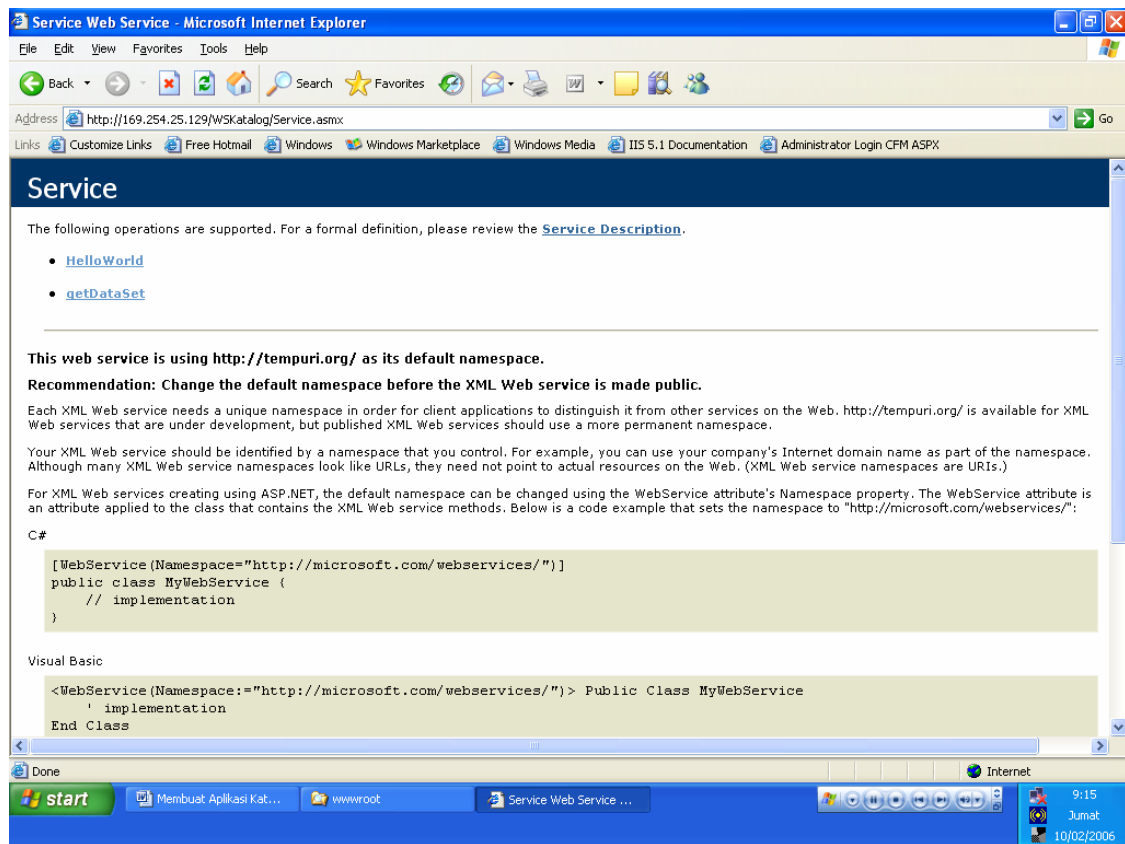
Lampiran E: Memeriksa Web Service

Pada lampiran ini, anda akan menginvoke Web service dan mnginvestigasi file Web Services Description Language (WSDL)

Menginvoke Web Service

Pada lampiran ini, Anda akan memeriksa Web service dan mengobservasi bagaimana objek yang dikembalikan dari Web service diserialisasi sebagai XML. Ikuti langkah-langkah sebagai berikut ini:

1. Buka jendela Web browser, lalu ketikkan URL Web service, misalnya <http://169.254.25.129/WSKatalog/Service.asmx>



2. Klik pada link **getDataSet**, lalu klik tombol **Invoke**
3. Pelajari bagaimana objek diserialisasi sebagai XML

```
<?xml version="1.0" encoding="utf-8" ?>
<DataSet xmlns="http://tempuri.org/">
  <xs:schema id="NewDataSet" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
  <xs:complexType>
  <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element name="BUKU">
  <xs:complexType>
  <xs:sequence>
    <xs:element name="ID" type="xs:int" minOccurs="0" />
    <xs:element name="JUDUL" type="xs:string" minOccurs="0" />
    <xs:element name="KETERANGAN" type="xs:string" minOccurs="0" />
    <xs:element name="GAMBAR" type="xs:base64Binary" minOccurs="0" />
  </xs:sequence>
  </xs:complexType>
  </xs:element>
  </xs:choice>
```

```
</xs:complexType>  
</xs:element>  
</xs:schema>
```

4. Kenali nama dan tipe elemen. Anda akan menggunakannya saat mempopulasi kontrol ListView

Memeriksa file WSDL

Dalam lampiran ini Anda akan memeriksa file WSDL yang disediakan oleh .NET untuk pengguna Web services.

1. Navigasi ke halaman WSDL Web services
2. Periksa file WSDL. Perhatikan ada beberapa definisi yang diulangi sampai tiga kali. Masing-masing definisi untuk SOAP, HTTP GET, dan HTTP POST. Definisi ini menggambarkan bagaimana Web service bisa diakses menggunakan protokol yang berbeda-beda. SOAP adalah protokol yang biasa digunakan oleh aplikasi klien untuk mengakses suatu Web service, sementara method HTTP GET dikembangkan oleh web browser saat memanggil web service