

# Membuat Windows Service Dengan .Net Core

## ***Lisensi Dokumen:***

*Copyright © 2003 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.*

**Junindar, ST, MCPD, MOS, MCT, MVP**

*junindar@gmail.com*

<http://junindar.blogspot.com>

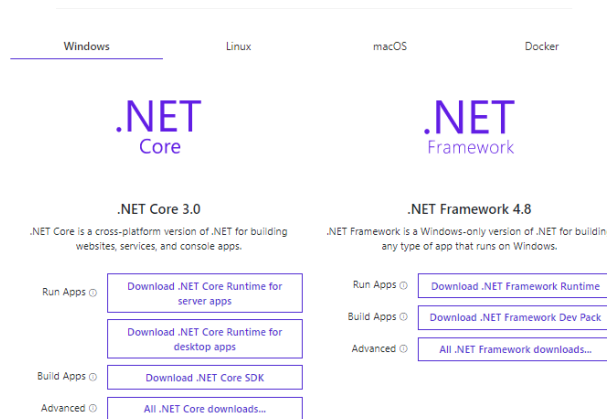
## Abstrak

Windows Service adalah sebuah aplikasi yang tidak memiliki user interface dan berjalan di belakang (*background*). Biasanya digunakan untuk membantu atau melayani dari sistem utama, seperti *Operating System*. Tidak seperti aplikasi pada umumnya, dimana aplikasi berjalan karena ada tindakan dari pengguna sebelumnya. Windows Service dapat berjalan secara otomatis tanpa ada campur tangan dari pengguna.

## Pendahuluan

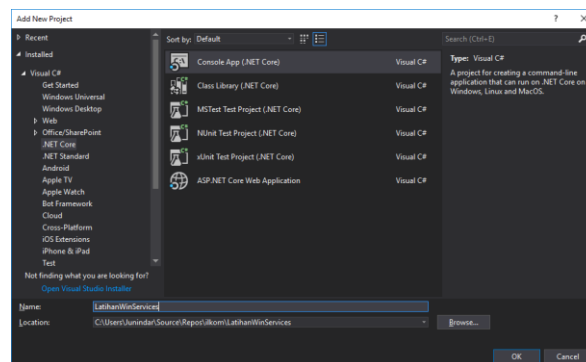
.Net Core adalah sebuah framework yang bersifat open-source dan multiplatform. Yang artinya dapat berjalan pada Linux, MacOS maupun Windows. Tidak seperti .Net Framework yang hanya dapat berjalan pada system operasi Windows saja.

Saat ini .Net Core hanya mendukung dua Bahasa pemrograman yaitu C# dan F#. Untuk mendapatkan .Net Core dapat diunduh pada link berikut :

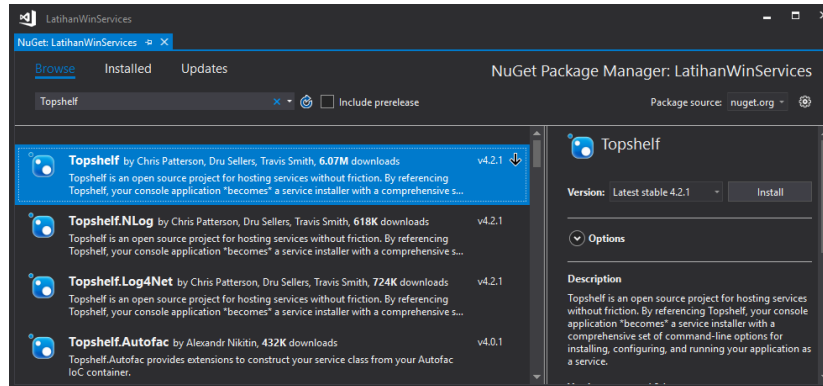


Untuk membuat Windows service dengan menggunakan .Net Core berbeda dengan menggunakan .Net Framework biasa. Pada Visual Studio untuk .Net Core tidak tersedia template untuk membuat Windows Service, seperti pada .Net Framework.

Untuk .Net Core kita akan menggunakan Console Project, seperti pada gambar dibawah ini.



Setelah selesai membuat project console seperti diatas, lamglah selanjutnya dengan menambahkan atau menginstall nuget package Topshelf. dengan cara klik kanan pada project lalu pilih “Manage Nuget Packages”, pada tab Browse ketikkan “Topshelf”, setelah keluar, pilih dan klik Install. Pada Packages ini akan disediakan sebuah ServiceControl yang akan digunakan untuk membuat Windows Service.



Pada ServiceControl terdapat method yang digunakan pada Windows Service, seperti Start dan Stop. Selanjutnya kita akan membuat class yang menggunakan ServiceControl, seperti dibawah ini.

```
class LatihanService : ServiceControl
{
    private const string LogLocation = @"C:\temp\mylog.txt";

    private void WriteLog(string message)
    {
        Directory.CreateDirectory(Path.GetDirectoryName(LogLocation));
        File.AppendAllText(LogLocation,
            $"{DateTime.Now.ToString(CultureInfo.CurrentCulture)} : {message}"
            {Environment.NewLine});
    }

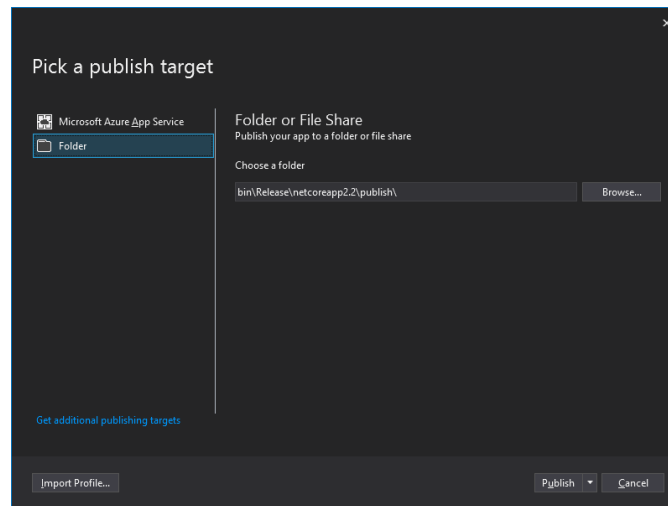
    public bool Start(HostControl hostControl)
    {
        WriteLog("Service Mulai");
        return true;
    }

    public bool Stop(HostControl hostControl)
    {
        WriteLog("Service Berhenti");
        return true;
    }
}
```

Pada class diatas dapat dilihat, kita melakukan implementasi pada method Start dan Stop. Dan terdapat method “WriteLog” untuk membuat file log, sehingga jika kita menjalankan atau memberhentikan service maka akan dicatat didalam file log. Langkah selanjutnya pada method “Main”, ketikkan sintaks dibawah.

```
HostFactory.Run(s => s.Service<LatihanService>());
```

Setelah selesai dengan langkah-langkah diatas, maka selanjutnya kita akan publish project ini terlebih dahulu. Klik kanan pada project dan klik Publish, sehingga keluar dialog seperti pada gambar dibawah. Pada “Pick a publish target” pilih “Folder”. Jika ingin mengganti lokasi hasil publish dapat melakukannya dengan klik button “Browse”. Dan yang terakhir klik button “Publish”.



Untuk menginstall service, kita akan melakukannya dengan menggunakan “command prompt”. Buka command prompt (**as an administrator**) dan ketikkan command seperti dibawah.

**“[Path]\[NamaProject].exe” Install**

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>"D:\WinServicePublish\LatihanWinServices.exe" install
Configuration Result:
[Success] Name LatihanWinServices
[Success] ServiceName LatihanWinServices
Topshef v4.2.1.215, .NET Framework v4.0.30319.42000

Running a transacted installation.

Beginning the Install phase of the installation.
Installing LatihanWinServices service
Installing service LatihanWinServices...
Service LatihanWinServices has been successfully installed.

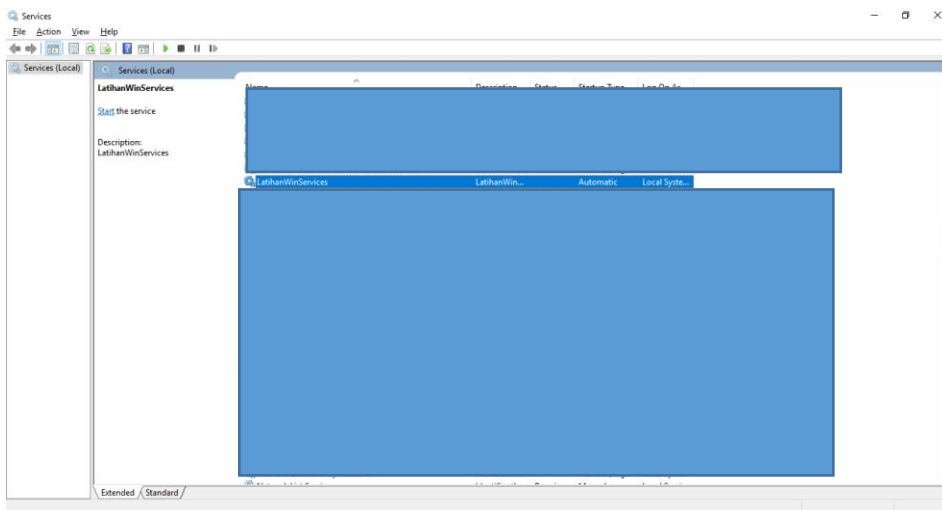
The Install phase completed successfully, and the Commit phase is beginning.

The Commit phase completed successfully.

The transacted install has completed.

C:\WINDOWS\system32>_
```

Coba buka service, pastikan “LatihanService” service terdapat disana, seperti pada gambar dibawah.



Untuk menjalankan, memberhentikan dan menghapus service, bisa menggunakan command seperti berikut.

SC START [Nama Service]	Start Service
SC STOP [Nama Service]	Stop Service
SC DELETE [Nama Service]	Delete Service

Untuk memastikan apakah sintaks kita berjalan dengan baik, buka file log dan pastikan mendapatkan hasil seperti pada gambar dibawah.

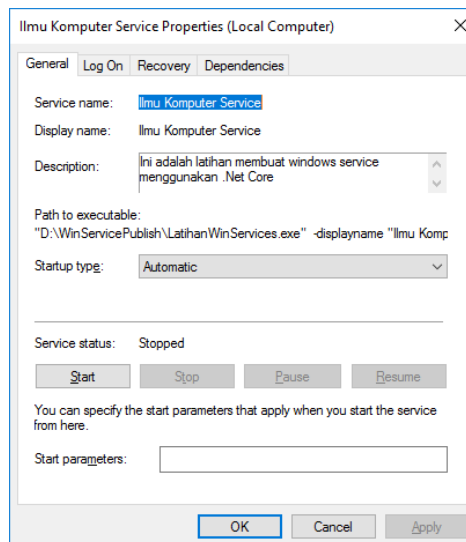
```
mylog.txt - Notepad
File Edit Format View Help
11/12/2019 6:08:27 PM : Service Mulai
11/12/2019 6:08:47 PM : Service Berhenti
```

Selanjutnya bagaimana jika kita ingin membuat nama service sendiri beserta deskripsi dari service kita. Dengan menggunakan Topshelf kita dapat membuat itu semua dengan mudah.

Ganti sintaks pada Main method, menjadi seperti dibawah ini.

```
HostFactory.Run(configurator =>
{
    configurator.SetServiceName("Ilmu Komputer Service");
    configurator.SetDescription("Ini adalah latihan membuat windows
service menggunakan .Net Core");
    configurator.Service<LatihanService>();
});
```

Seperti pada sintaks diatas, terdapat fungsi untuk membuat service name dan description pada service kita, sehingga mendapatkan hasil seperti dibawah.

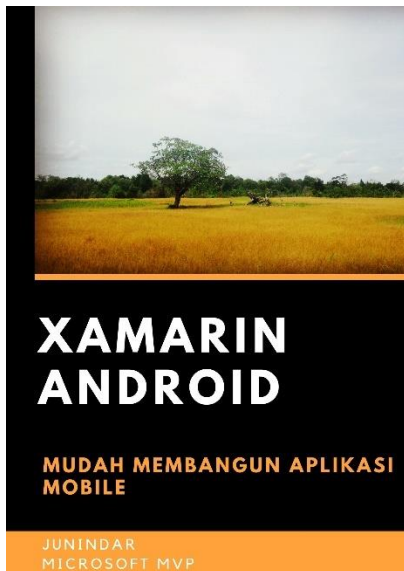


## **Penutup**

Sedangkan untuk memudahkan dalam memahami isi artikel, maka penulis juga menyertakan dengan full source code project latihan ini, dan dapat di download disini

<http://junindar.blogspot.com/2019/11/membuat-windows-service-dengan-net-core.html>

## Referensi



<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>

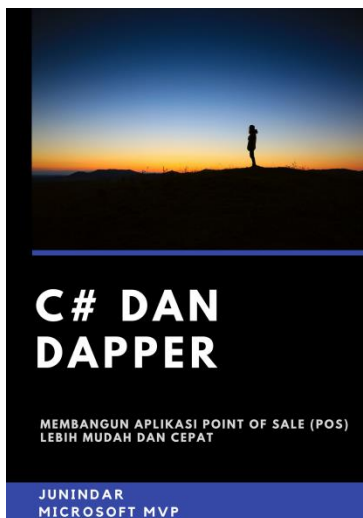


<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>





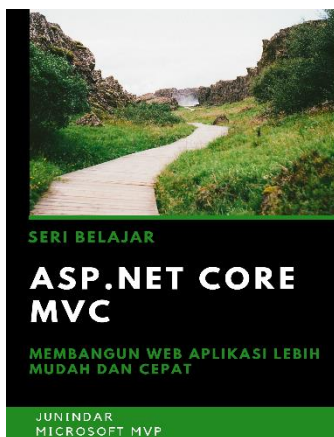
[https://play.google.com/store/books/details/Junindar\\_Xamarin\\_Forms?id=6Wg-DwAAQBAJ](https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ)



[https://play.google.com/store/books/details/Junindar\\_C\\_dan\\_Dapper\\_Membangun\\_Aplikasi\\_POS\\_Point?id=6TErDwAAQBAJ](https://play.google.com/store/books/details/Junindar_C_dan_Dapper_Membangun_Aplikasi_POS_Point?id=6TErDwAAQBAJ)



[https://play.google.com/store/books/details/Junindar ASP NET MVC Membangun Aplikasi Web Lebih?id=XLlyDwAAQBAJ](https://play.google.com/store/books/details/Junindar_ASP_NET_MVC_Membangun_Aplikasi_Web_Lebih?id=XLlyDwAAQBAJ)



[https://play.google.com/store/books/details/Junindar ASP NET CORE MVC?id=xEe5DwAAQBAJ](https://play.google.com/store/books/details/Junindar_ASP_NET_CORE_MVC?id=xEe5DwAAQBAJ)

## Biografi Penulis.



Junindar Lahir di Tanjung Pinang, 21 Juni 1982. Menyelesaikan Program S1 pada jurusan Teknik Inscreenatika di Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN-Bandung). Junindar mendapatkan Award Microsoft MVP VB pertanggal 1 oktober 2009 hingga saat ini. Senang mengutak-atik computer yang berkaitan dengan bahasa pemrograman. Keahlian, sedikit mengerti beberapa bahasa pemrograman seperti : VB.Net, C#, SharePoint, ASP.NET, VBA. Reporting: Crystal Report dan Report Builder. Database: MS Access, MY SQL dan SQL Server. Simulation / Modeling Packages: Visio Enterprise, Rational Rose dan Power Designer. Dan senang bermain gitar, karena untuk bisa menjadi pemain gitar dan seorang programmer sama-sama membutuhkan seni. Pada saat ini bekerja di salah satu Perusahaan Consulting dan Project Management di Malaysia sebagai Senior Consultant. Memiliki beberapa sertifikasi dari Microsoft yaitu Microsoft Certified Professional Developer (MCPD – SharePoint 2010), MOS (Microsoft Office Specialist) dan MCT (Microsoft Certified Trainer) Mempunyai moto hidup: **“Jauh lebih baik menjadi Orang Bodoh yang giat belajar, dari pada orang Pintar yang tidak pernah mengimplementasikan ilmunya”**.