IlmuKomputer.Com
Ikatlah ilmu dengan menuliskannya

Membuat Windows Service

Dengan .Net Core dan Quartz.NET

Junindar, ST, MCPD, MOS, MCT, MVP

junindar@gmail.com

http://junindar.blogspot.com

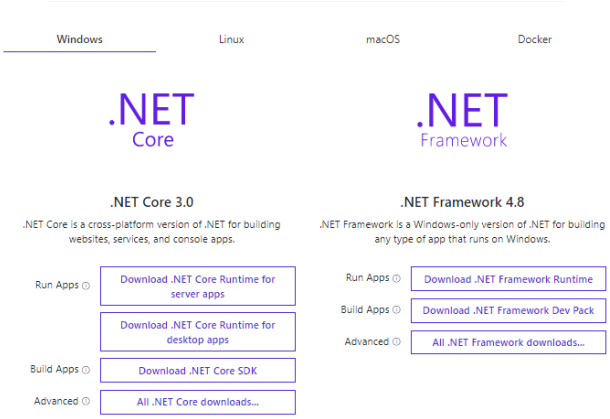
Abstrak

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 umum nya, 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 pemograman yaitu C# dan F#. Untuk mendapatkan .Net Core dapat diunduh pada link berikut : https://dotnet.microsoft.com/download



Untuk memulai artikel ini, disarankan untuk terlebih dahulu membaca dan membuat latihan pada artikel sebelumnya yang bisa didapatkan disini : http://junindar.blogspot.com/2019/11/membuat-windows-service-dengan-net-core.html

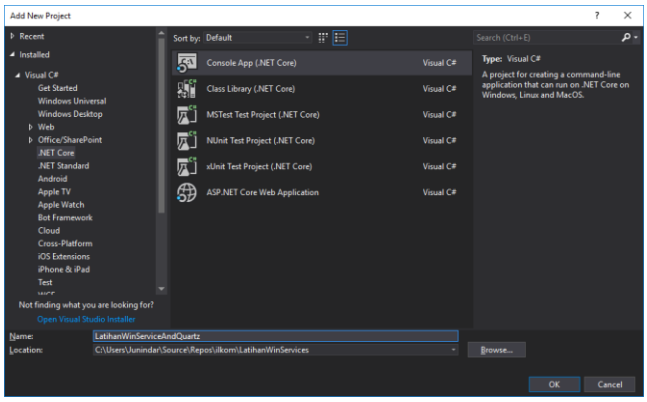
Pada latihan sebelumnya telah dijelaskan bagaimana membuat Windows Service dengan menggunakan .Net Core dan Topshelf. Tetapi pada latihan sebelumnya service yang kita buat belum menggunakan schedule atau “timer” kapan proses dijalankan.

Sebagai contoh kita ingin membuat service, dimana pada setiap menit service yang kita buat akan menjalankan sebuah proses tertentu. Sebenarnya pada Topshelf menyediakan fungsi untuk melakukan restart pada service dan kita bisa tentukan kapan akan dilakukan restart service tersebut. Tetapi menurut penulis menggunakan itu kurang ideal.

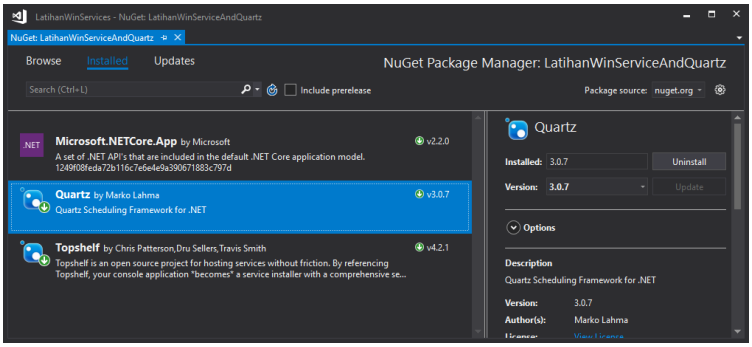
Oleh karena itu kita perlu menggunakan library selain Topshelf yaitu “Quartz”. Quartz adalah library yang bersifat open source yang digunakan untuk system job scheduling yang dapat digunakan baik dalam aplikasi kecil sampai dengan system yang besar. Library Quartz.Net dibangun dengan menggunakan Bahasa Pemograman C#. Quartz.Net

merupakan hasil *porting* dari Quartz yang merupakan Java job scheduling yang bersifat open source juga.

Untuk langkah awal, buat terlebih dahulu project console dengan menggunakan .Net Core.



Tambahkan Nuget Package Topshelf dan Quartz pada project.



Tambahkan sebuah class dengan nama “Log“ dan ketikkan sintaks seperti dibawah ini.

Class ini digunakan untuk membuat file log, yang akan kita gunakan nantinya. Dimana nama file akan menggunakan tanggal. Jika pada tanggal dieksekusinya program tidak terdapat file log, maka secara otomatis akan dibuat, dan jika file sudah ada maka hanya akan menambahkan text kedalam file log tersebut.

```
public class Log
{

        private static ReaderWriterLockSlim _readWriteLock = new ReaderWriterLockSlim();

        public static void WriteLog( string Message)
        {

            string LogFolder = Path.GetDirectoryName(Assembly.GetEntryAssembly().Location) + "\\Logs"; ;
            if (!Directory.Exists(LogFolder)) Directory.CreateDirectory(LogFolder);

            String LogFileName = DateTime.Now.ToString("yyyy.MM.dd") + ".txt";
            LogFileName = LogFolder + @"\" + LogFileName;
            string msg = DateTime.Now + ": " + Message + Environment.NewLine;

            _readWriteLock.EnterWriteLock();
            try
            {
                using (StreamWriter sw = File.AppendText(LogFileName))
                {
                    sw.WriteLine(msg);
                    sw.Close();
                }
            }
            finally
            {
                _readWriteLock.ExitWriteLock();
            }

        }


}
```

Selanjutnya tambahkan sebuah class lagi dengan nama “JobClass“. Class ini mengimplementasikan method pada Interface“IJob“ yaitu method “Execute“. Method ini digunakan untuk mengeksekusi proses yang diinginkan. Pada sintaks dibawah dapat dilihat, terdapat method WriteLog(“Proses Start“), dimana sebelum menjalankan proses kita catat dulu kedalam file log. Lalu kita buat Delay selama 5 detik, sintaks ini diasumsikan sebagai proses yang akan kita buat atau bisa dikatakan proses ini akan memakan waktu selama 5 detik. Setelah selesai, selanjutnya kita catat kembali “Proses Finish“ kedalam file log.

```
public  class JobClass : IJob
{
        public async Task Execute(IJobExecutionContext context)
        {
            Log.WriteLog("Proses Start");
            await Task.Delay(5000);
            Log.WriteLog("Proses Finish");
        }
}
```

Setelah selessai dengan langkah diatas, kita lanjutkan dengan membuat sebuah class kembali dengan nama “JobService“, seperti pada sintaks dibawah.

```
public class JobService
{
        public void OnStart()
        {
            Log.WriteLog("Service Start");
            Task<IScheduler> scheduler = StdSchedulerFactory.GetDefaultScheduler();
            scheduler.Result.Start();

            IJobDetail job = JobBuilder.Create<JobClass>().Build();

            ITrigger trigger = TriggerBuilder.Create()
                .WithIdentity("LatihanQuartz", "LatihanQuartz")
                .StartNow()
                .WithSimpleSchedule(x => x
                    .WithIntervalInMinutes(1)
                    .RepeatForever())
                .Build();

            scheduler.Result.ScheduleJob(job, trigger);


        }

        public void OnStop()
        {
            Log.WriteLog("Service Stop");
        }

}
```

Terdapat dua buah method yaitu OnStart dan OnStop. Pada method OnStart, disinilah kita akan mengatur scheduller dengan menggunakan Quartz. Sebelumnya kita panggil method WriteLog untuk mencatat kapan Service berjalan. Pada saat service dijalankan, maka pada saat yang sama scheduller juga akan dijalankan (scheduler.Result.Start();). Selanjutnya kita dapat mengatur interval, kapan secheduller akan dijalankan (WithIntervalInMinutes). Banyak terdapat pilihan pada pengaturan interval, seperti IntervalInHours, Seconds dan lain-lain.

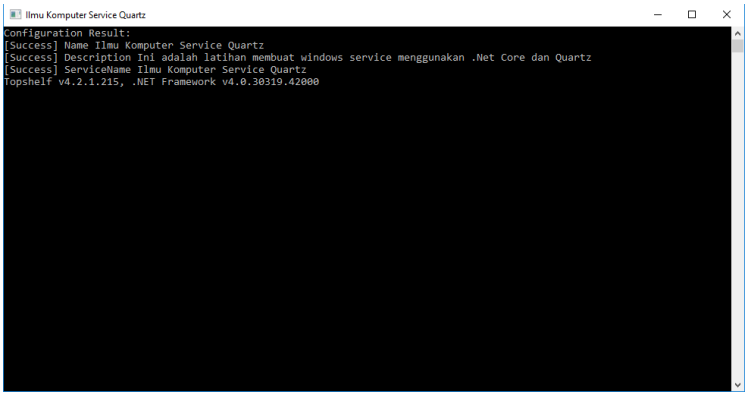
Untuk method OnStop, kita panggil method WriteLog untuk mencatat jika service dihentikan.

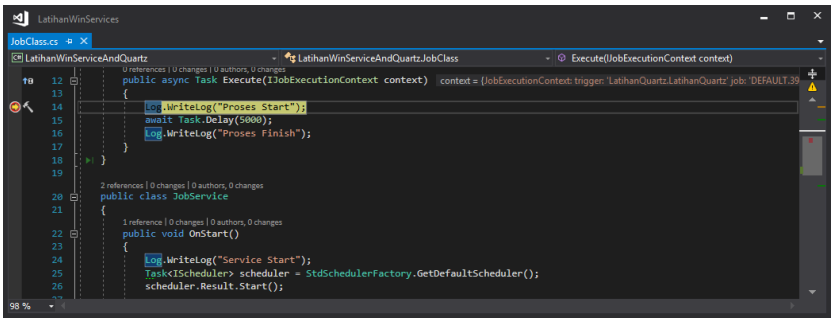
Langkah terakhir buka file Program.cs dan pada method Main ketikkan sintaks dibawah. Disini kita akan menggunakan Topshelf, untuk membuat windows service. Dengan menggunakan Topshelf kita dapat mengatur nama dan deskripsi dari service tersebut.

Sedangkan untuk mengakses method OnStart dan OnStop pada saat service dijalankan atau dihentikan kita gunakan method WhenStarted dan WhenStopped.

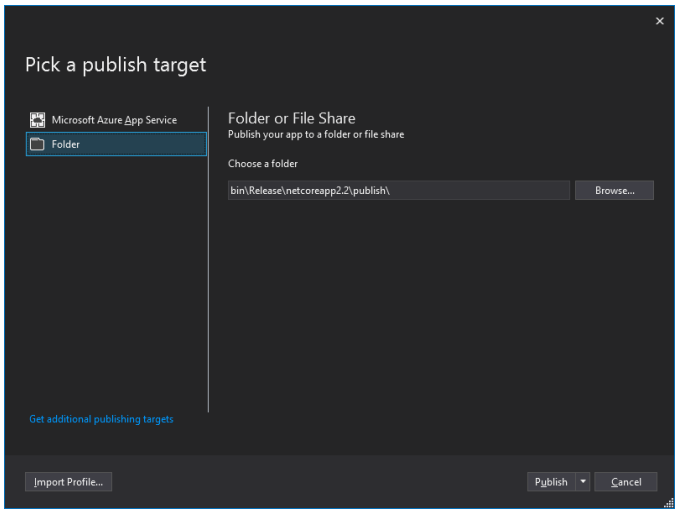
```
HostFactory.Run(configurator =>
{
    configurator.SetServiceName("Ilmu Komputer Service");

    configurator.SetDescription("Ini adalah latihan membuat windows service
    menggunakan .Net Core dan Quartz");



    configurator.Service<JobService>(serviceConfigurator =>
    {
         serviceConfigurator.ConstructUsing(() => new JobService());

         erviceConfigurator.WhenStarted((service, hostControl) =>
         {
               service.OnStart();
               return true;
          });
          serviceConfigurator.WhenStopped((service, hostControl) =>
          {
                 service.OnStop();
                 return true;
           });
      });
});
```

Dengan menggunakan Topshelf kita dapat melakukan proses debug dengan mudah, jalankan aplikasi dan kita dengan mudah mengecek sintaks yang telah dibuat.



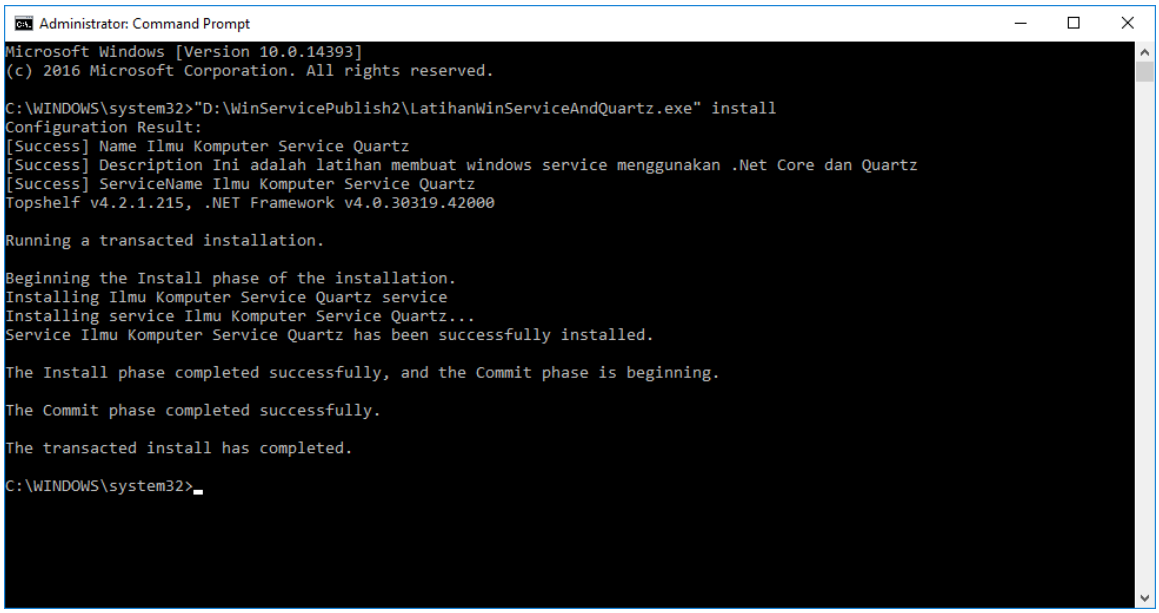


Setelah selesai dengan langkah-langkah diatas, maka selanjutnya kita akan publish project ini terlebih dahulu. Klik kanan pada project dan klik Publish, sehingga keluar dialog seperti pada gambar dibawah. Pada “Pick a publish target” pilih “Folder”. Jika ingin mengganti lokasi hasil publish dapat melakukannya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

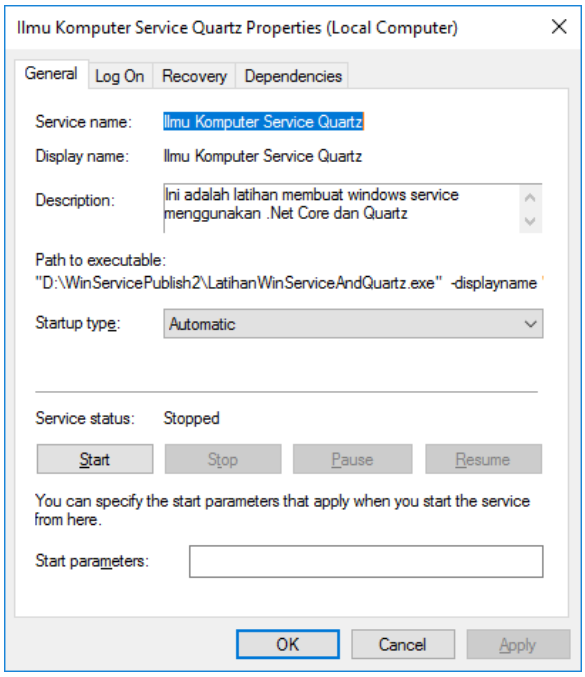


Untuk menginstall service, kita akan melakukannya dengan menggunakan “command prompt”. Buka command prompt (**as an administrator**) dan ketikkan command seperti dibawah.

“[Path]\[NamaProject].exe” Install



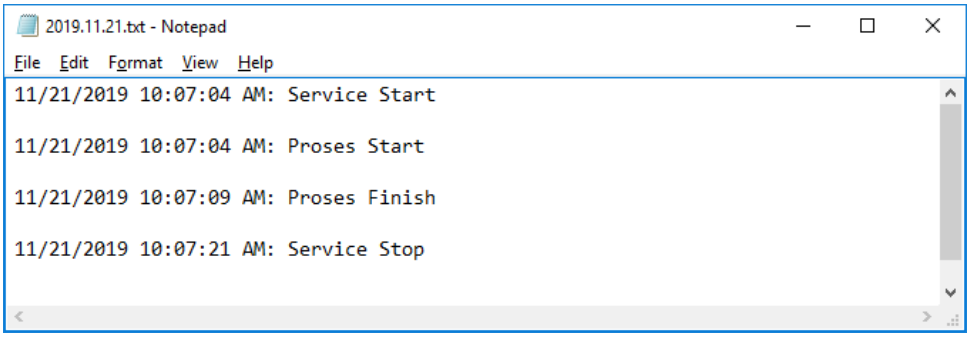
Coba buka service, pastikan “Ilmu Komputer Service Quartz” service terdapat disana, seperti pada gambar dibawah.



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.

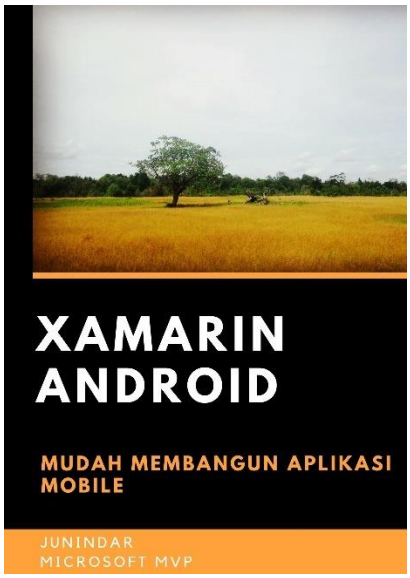


Dari file log diatas, dapat kita lihat antara proses Start dan finish, memiliki jeda selama 5 detik, sesuai dengan yang telah kita buat pada method Execute.

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_21.html

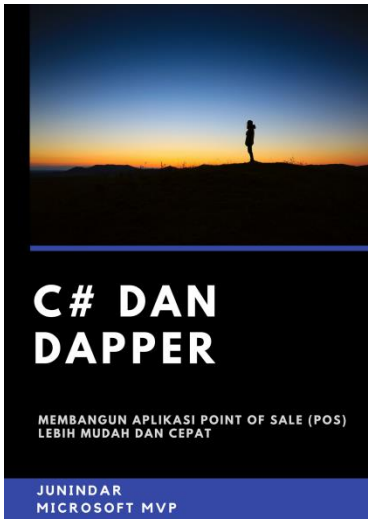
Referensi



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

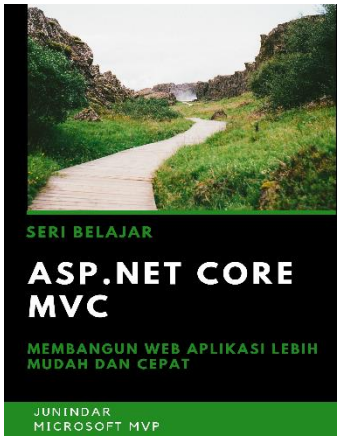
https://play.google.com/store/books/details?id=VSLiDQAAQBAJ

https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ



https://play.google.com/store/books/details/Junindar_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_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".**