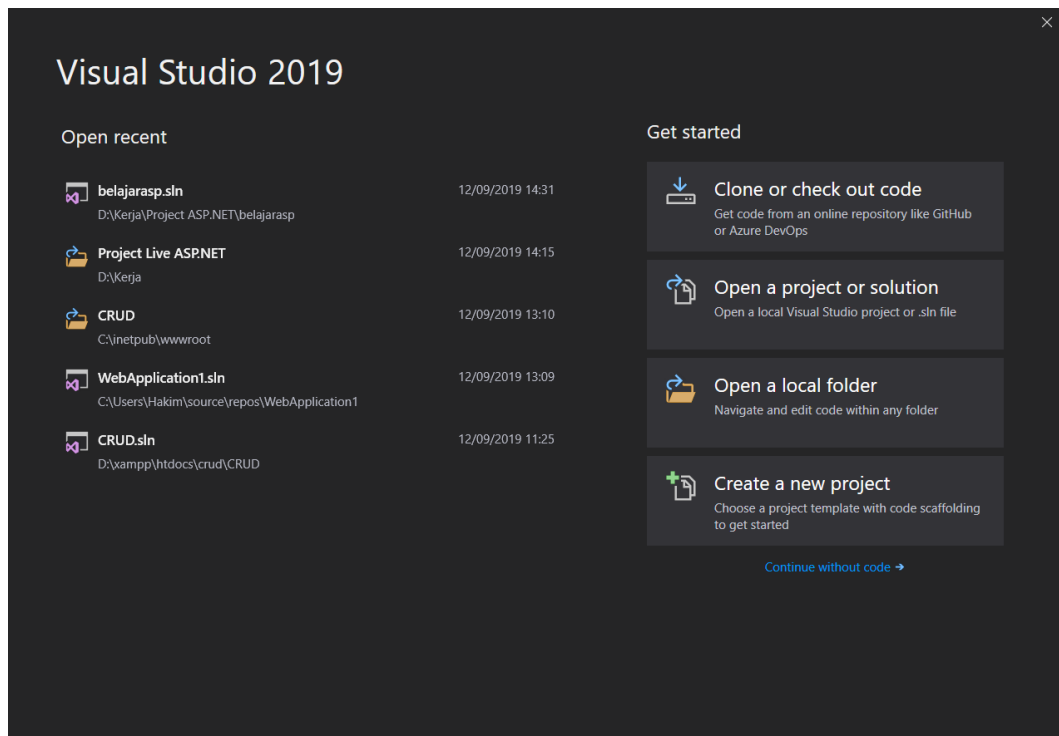
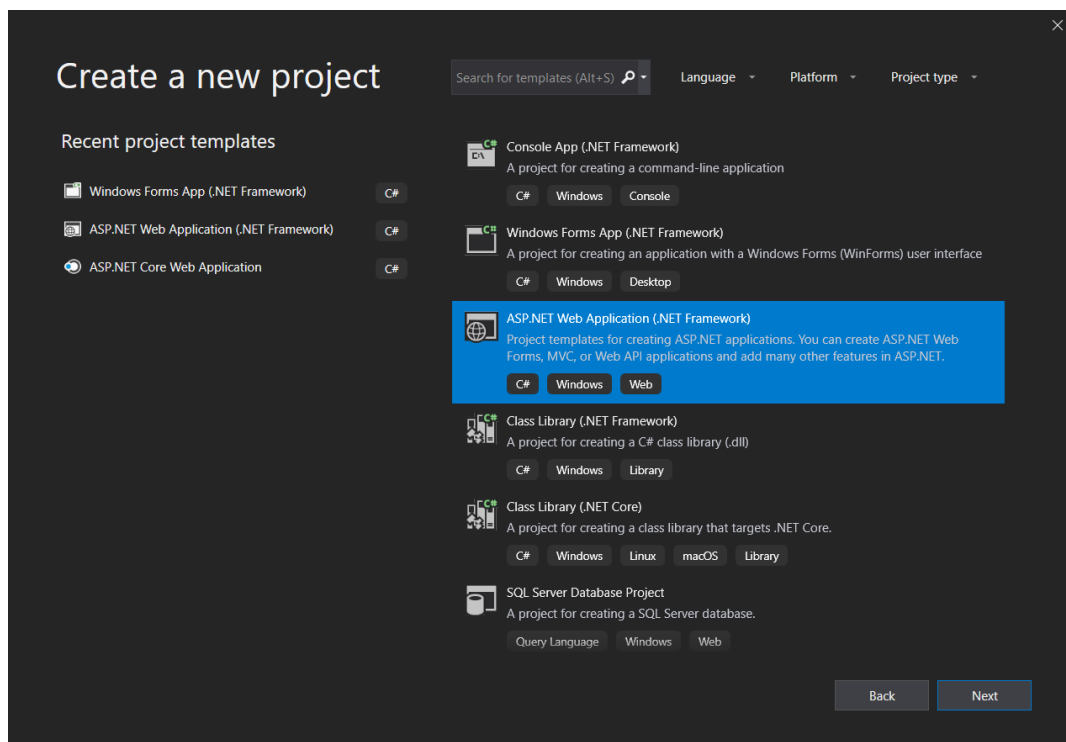


Dokumentasi install crud di asp.net

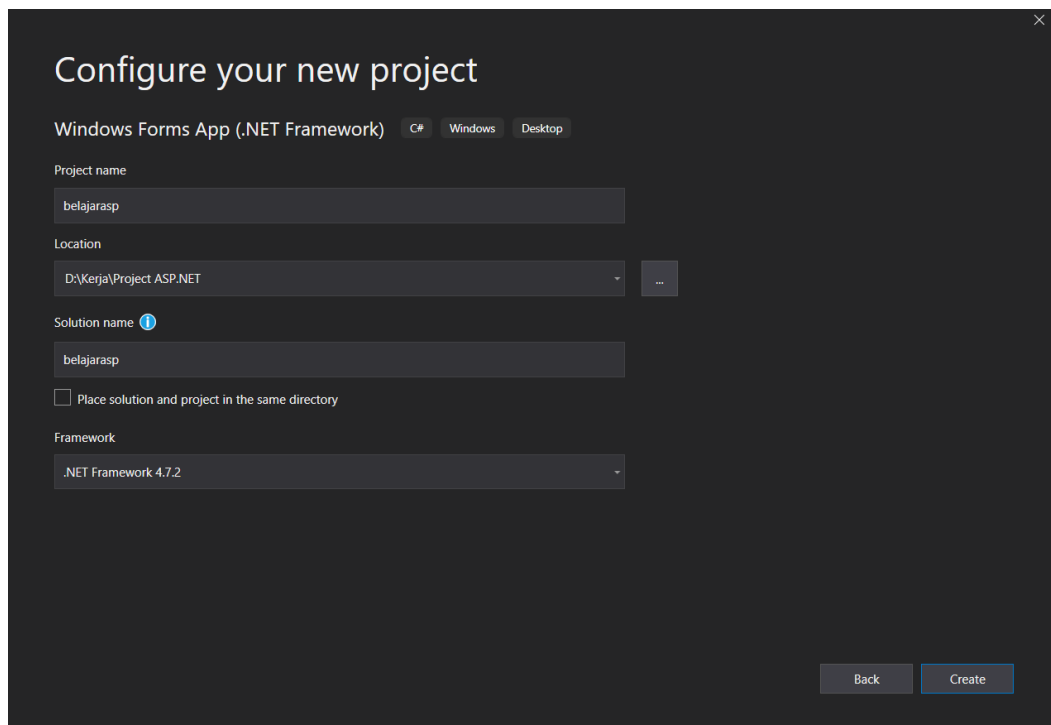
1. Buka aplikasi visual studio 2019 dengan “administrator” dan akan muncul tampilan seperti ini.



2. Kemudian “create new project” dan pilih seperti gambar berikut :



3. Klik next dan akan muncul nama project serta lokasi penyimpanan project. Jika sudah, maka akan seperti gambar berikut :



Configure your new project

Windows Forms App (.NET Framework) C# Windows Desktop

Project name
belajarasp

Location
D:\Kerja\Project ASP.NET

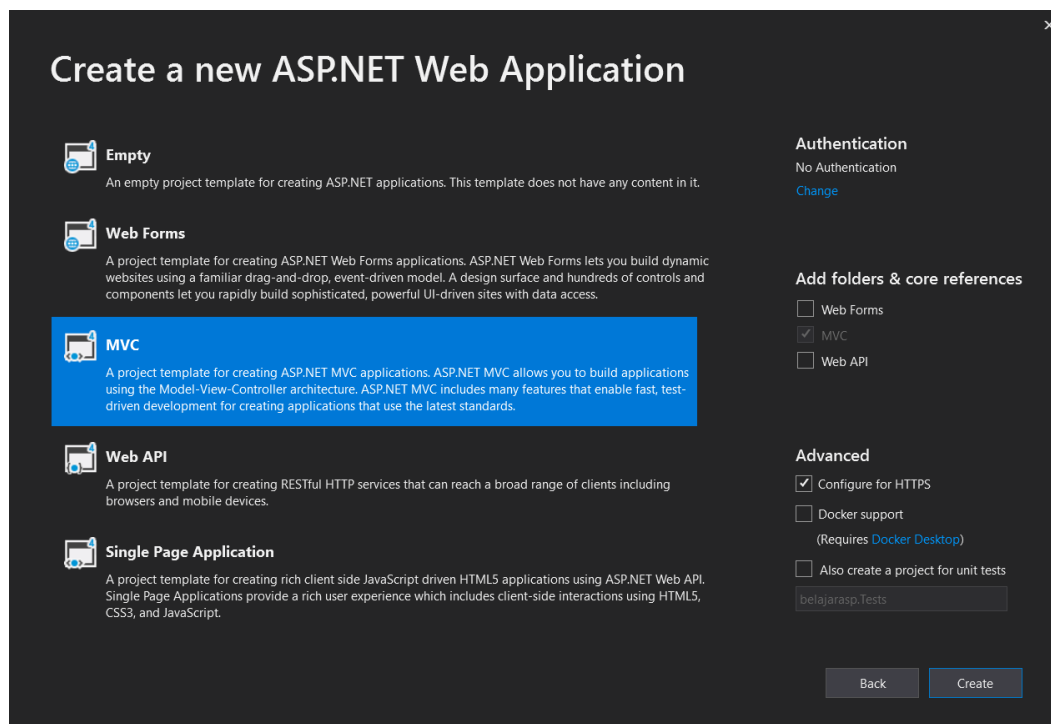
Solution name ⓘ
belajarasp

☐ Place solution and project in the same directory

Framework
.NET Framework 4.7.2

Back Create

4. Jika sudah yakin selanjutnya klik create, pilih menu mvc dan klik create seperti gambar berikut :



Create a new ASP.NET Web Application

Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
[Change](#)

Add folders & core references

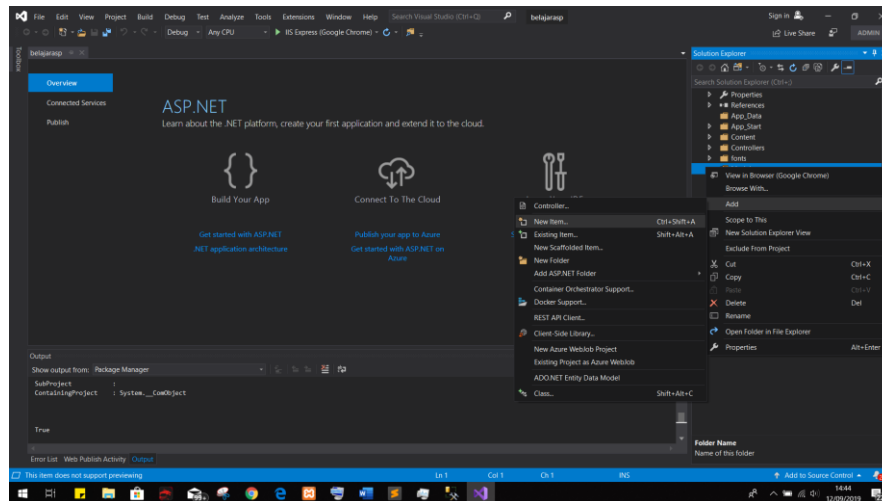
☐ Web Forms
☒ MVC
☐ Web API

Advanced

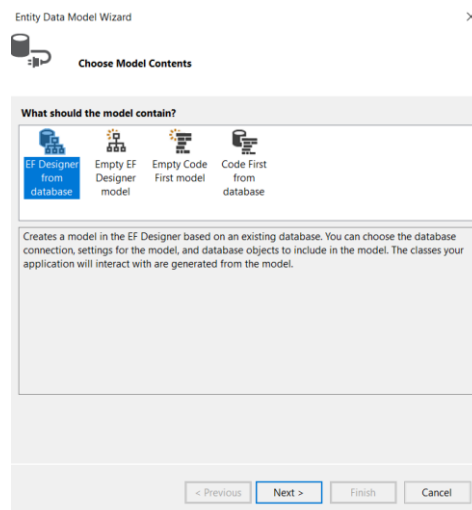
☒ Configure for HTTPS
☐ Docker support
(Requires [Docker Desktop](#))
☐ Also create a project for unit tests
belajarasp.Tests

Back Create

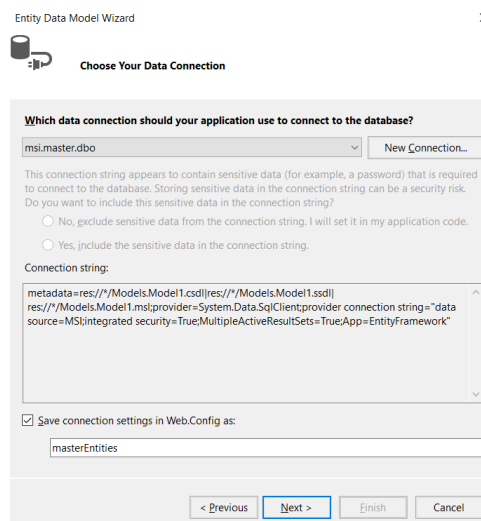
5. Maka selanjutnya klik kanan pada folder Models, kemudian pilih add->new item seperti berikut :



6. Pilih menu data kemudian ado.net entity data model, klik add. Maka akan muncul gambar seperti berikut :



7. Klik next, dan akan muncul gambar seperti berikut :



8. Pilih new connection dan untuk server name samakan seperti nama yang di sqlserver. Pilih database yang diinginkan misalnya disini “belajar”. Maka akan muncul seperti berikut :

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient) Change...

Server name: MSI Refresh

Log on to the server

Authentication: Windows Authentication

User name: Password: Save my password

Connect to a database

Select or enter a database name: belajar

Attach a database file: Browse...

Logical name:

Advanced...

Test Connection OK Cancel

9. Jika sudah selesai, maka muncul seperti berikut :

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

msi.belajar.dbo New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

No, exclude sensitive data from the connection string. I will set it in my application code.

Yes, include the sensitive data in the connection string.

Connection string:

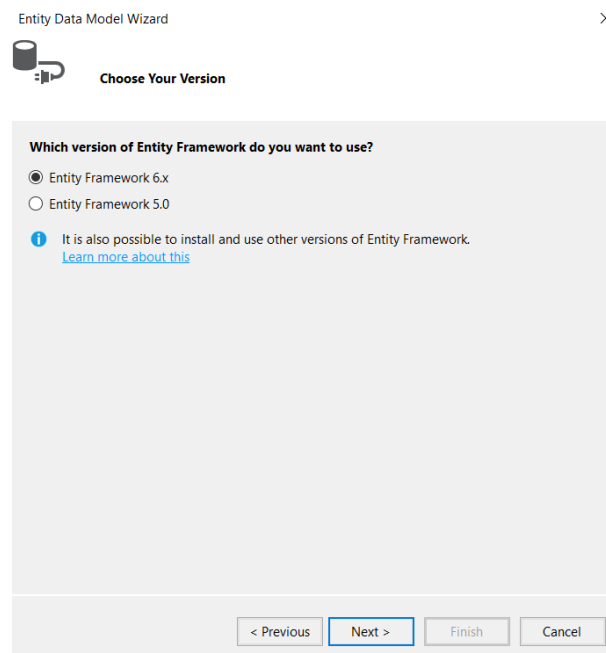
metadata=res://*/Models.Model1.csdl|res://*/Models.Model1.ssdl|res://*/Models.Model1.msl;provider=System.Data.SqlClient;provider connection string="data source=MSI;initial catalog=belajar;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

Save connection settings in Web.Config as:

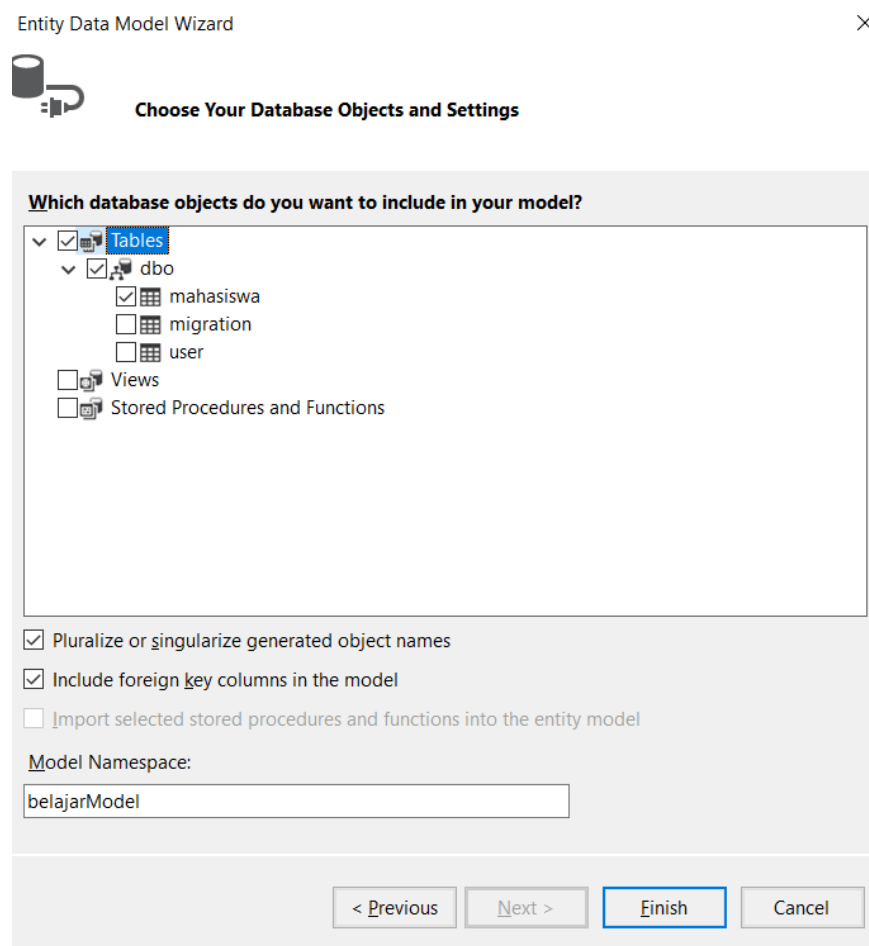
belajarEntities

< Previous Next > Finish Cancel

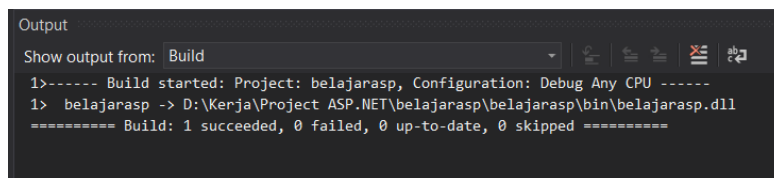
10. Kemudian pilih entity framework 6.x seperti gambar berikut :



11. Setelah itu pilih tabel yang ingin kita kerjakan, misalnya disini menggunakan tabel mahasiswa, maka akan muncul gambar seperti berikut :

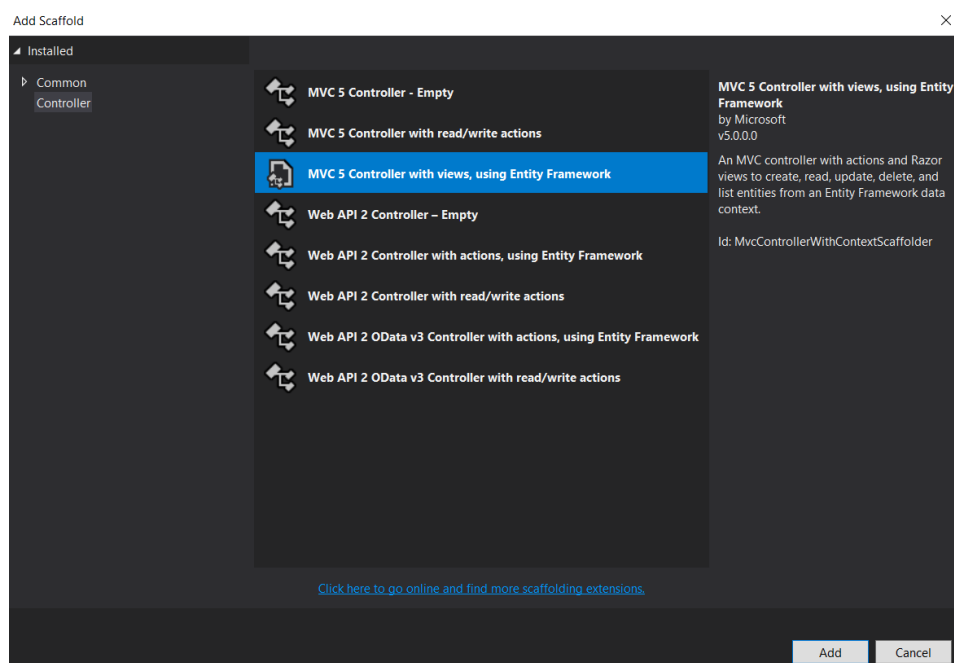


12. Setelah sudah klik finish, pilih menu build->build solution, maka akan muncul output seperti berikut :

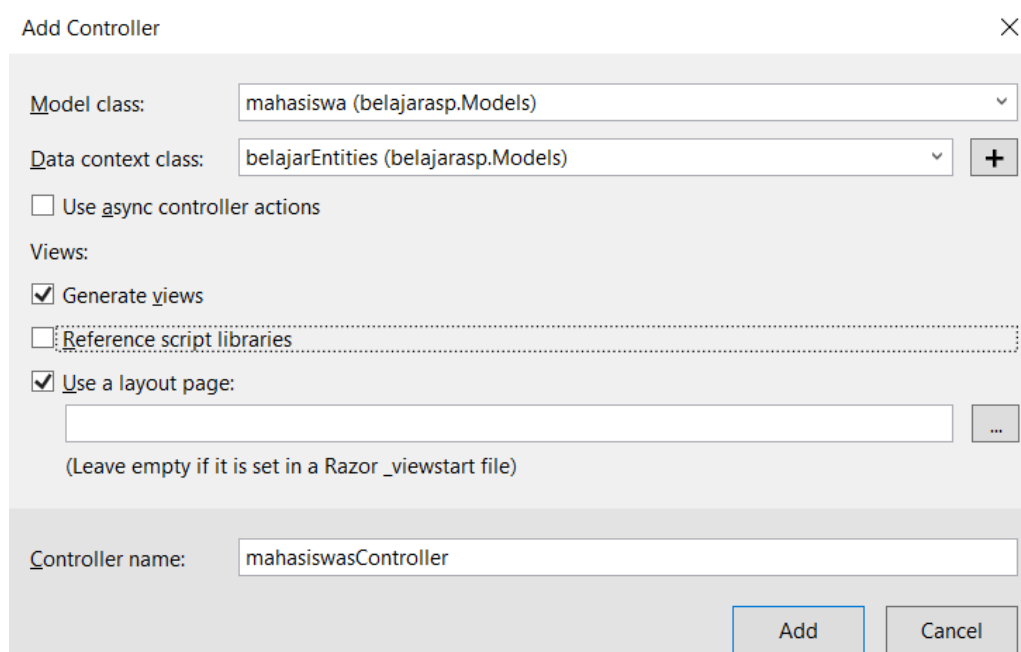


```
Output
Show output from: Build
1>----- Build started: Project: belajaras, Configuration: Debug Any CPU -----
1> belajaras -> D:\Kerja\Project ASP.NET\belajaras\belajaras\bin\belajaras.dll
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

13. Selanjutnya buat controller, klik kanan pada folder Controllers->add->Controller, pilih menu nomor 3 seperti gambar berikut :



14. Kemudian klik add, pilih model class dan data context class yang dibuat tadi. Maka akan muncul seperti gambar berikut :



Add Controller

Model class: mahasiswa (belajaras.Models)

Data context class: belajarEntities (belajaras.Models)

☐ Use async controller actions

Views:

☒ Generate views

☐ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor `_viewstart` file)

Controller name: mahasiswasController

Add Cancel

15. Otomatis controller crud akan ke generate otomatis seperti berikut :

```

10
11 namespace belajarasp.Controllers
12 {
13     [Route("")]
14     public class mahasiswaController : Controller
15     {
16         private belajarEntities db = new belajarEntities();
17
18         // GET: mahasiswa
19         public ActionResult Index()
20         {
21             return View(db.mahasiswa.ToList());
22         }
23
24         // GET: mahasiswa/Details/5
25         public ActionResult Details(int? id)
26         {
27             if (id == null)
28             {
29                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
30             }
31             mahasiswa mahasiswa = db.mahasiswa.Find(id);
32             if (mahasiswa == null)
33             {
34                 return HttpNotFound();
35             }
36             return View(mahasiswa);
37         }
38
39         // POST: mahasiswa/Create
40         // To protect from overposting attacks, please enable the following binding property.
41         // For more details on binding, see http://go.microsoft.com/fwlink/?LinkId=317594
42         public ActionResult Create([Bind(Exclude = "Id")] mahasiswa mahasiswa)
43         {
44             if (ModelState.IsValid)
45             {
46                 db.mahasiswa.Add(mahasiswa);
47                 db.SaveChanges();
48                 return RedirectToAction("Index");
49             }
50             return View(mahasiswa);
51         }
52
53         // PUT: mahasiswa/Update/5
54         // To protect from overposting attacks, please enable the following binding property.
55         // For more details on binding, see http://go.microsoft.com/fwlink/?LinkId=317594
56         public ActionResult Update(int id, mahasiswa mahasiswa)
57         {
58             if (id != mahasiswa.Id)
59             {
60                 return HttpNotFound();
61             }
62             if (ModelState.IsValid)
63             {
64                 db.Entry(mahasiswa).State = EntityState.Modified;
65                 db.SaveChanges();
66                 return RedirectToAction("Index");
67             }
68             return View(mahasiswa);
69         }
70
71         // DELETE: mahasiswa/Delete/5
72         public ActionResult Delete(int id)
73         {
74             if (id == null)
75             {
76                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
77             }
78             mahasiswa mahasiswa = db.mahasiswa.Find(id);
79             if (mahasiswa == null)
80             {
81                 return HttpNotFound();
82             }
83             db.mahasiswa.Remove(mahasiswa);
84             db.SaveChanges();
85             return RedirectToAction("Index");
86         }
87     }
88 }

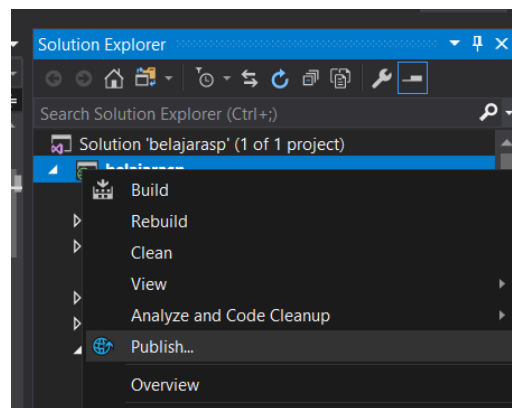
```

16. Selanjutnya konfigurasi IIS (Internet Information Services). Pada folder sites klik kanan dan pilih add website, maka akan muncul seperti gambar berikut :

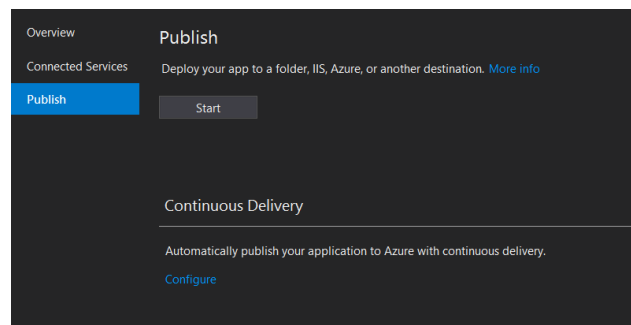
17. Jadi yang harus diisi ialah site name, physical path (folder kosong) dan mengganti port. Jika sudah klik ok, maka akan muncul gambar seperti berikut :

Name	ID	Status	Binding	Path
belajarasp	2	Started (ht...)	*:2409 (http)	D:\Kerja\Project Publish ASP.NET\belajarasp
Default Web Site	1	Started (ht...)	*:80 (http)	%SystemDrive%\inetpub\wwwroot

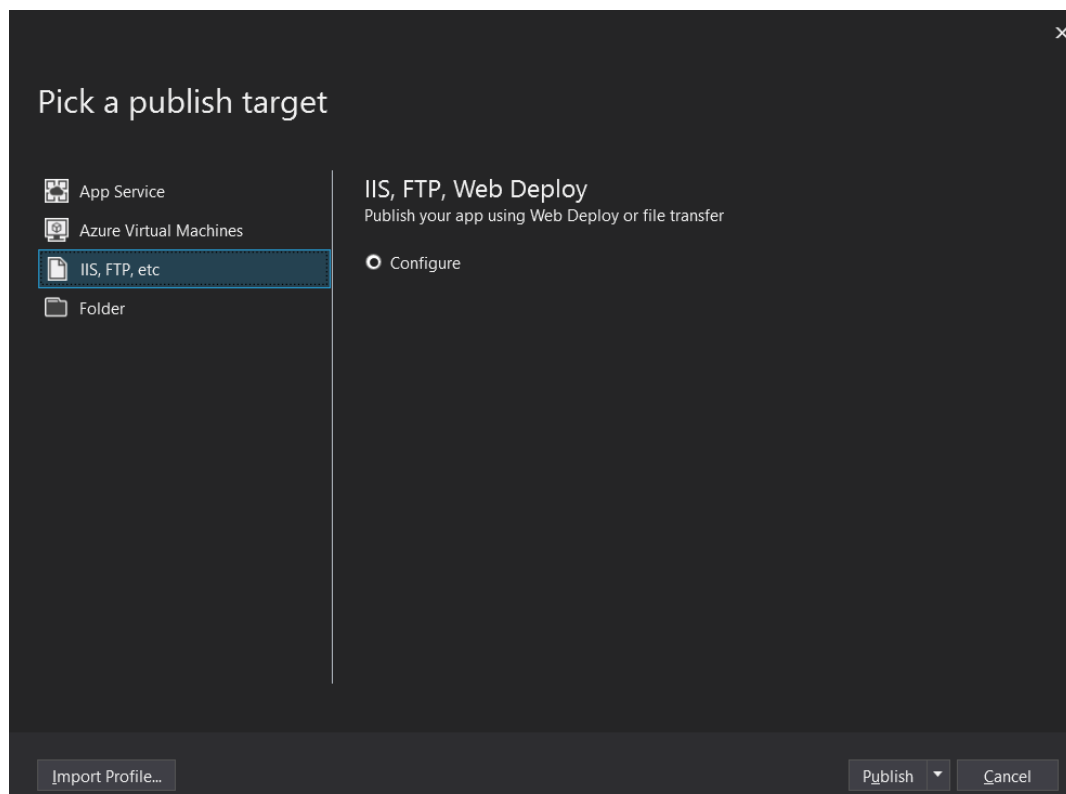
18. Kemudian kembali lagi ke project, selanjutnya klik kanan pada project dan pilih publish seperti berikut :



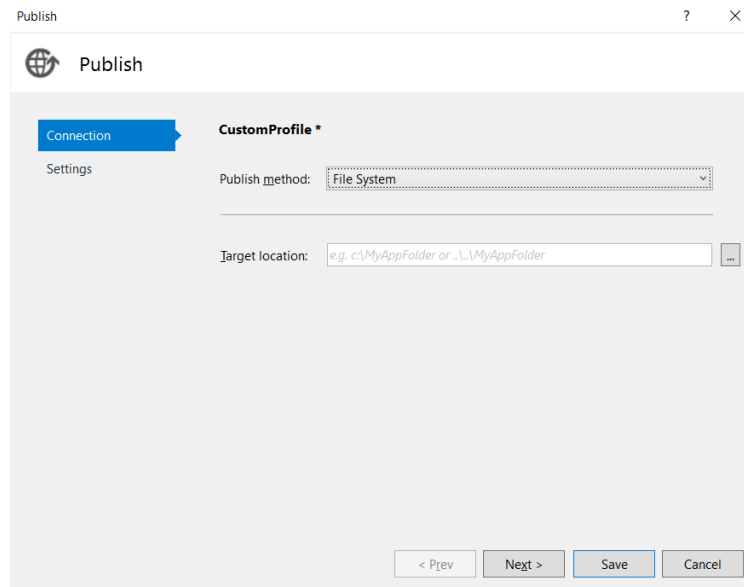
19. Selanjutnya klik start pada gambar berikut :



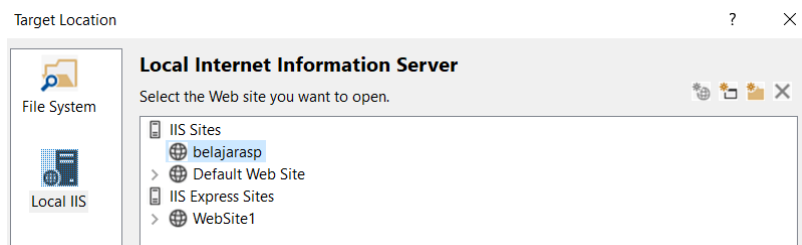
20. Setelah klik start, maka akan muncul beberapa menu. Pilih IIS,FTP,etc seperti gambar berikut :



21. Selanjutnya klik publish, kemudian pada field publish method pilih file system. Maka akan muncul gambar seperti berikut :



22. Klik pada target location, pilih menu Local IIS. Pada menu tersebut ada nama website yang kita buat tadi. Klik bagian itu seperti gambar berikut :



23.