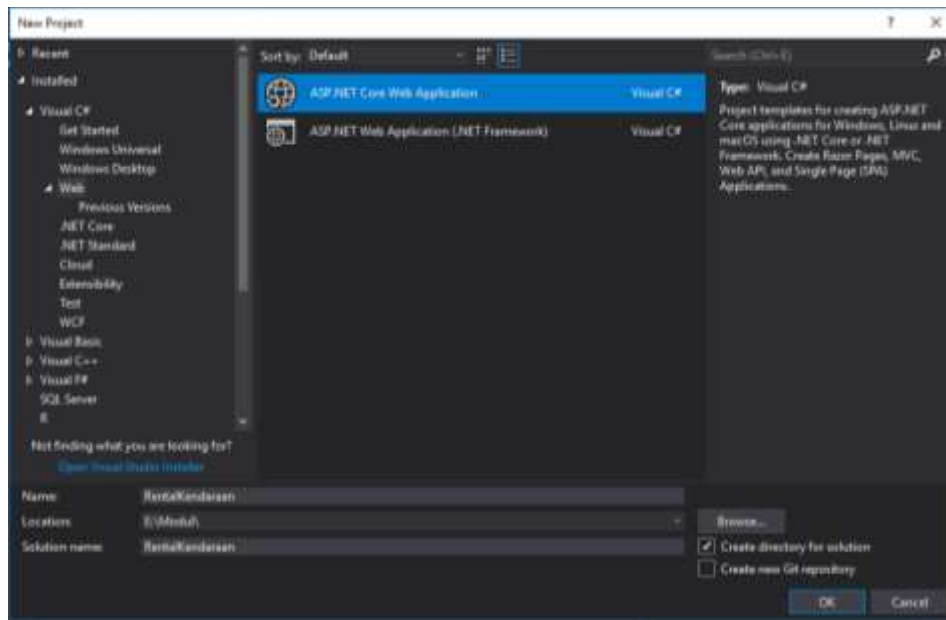


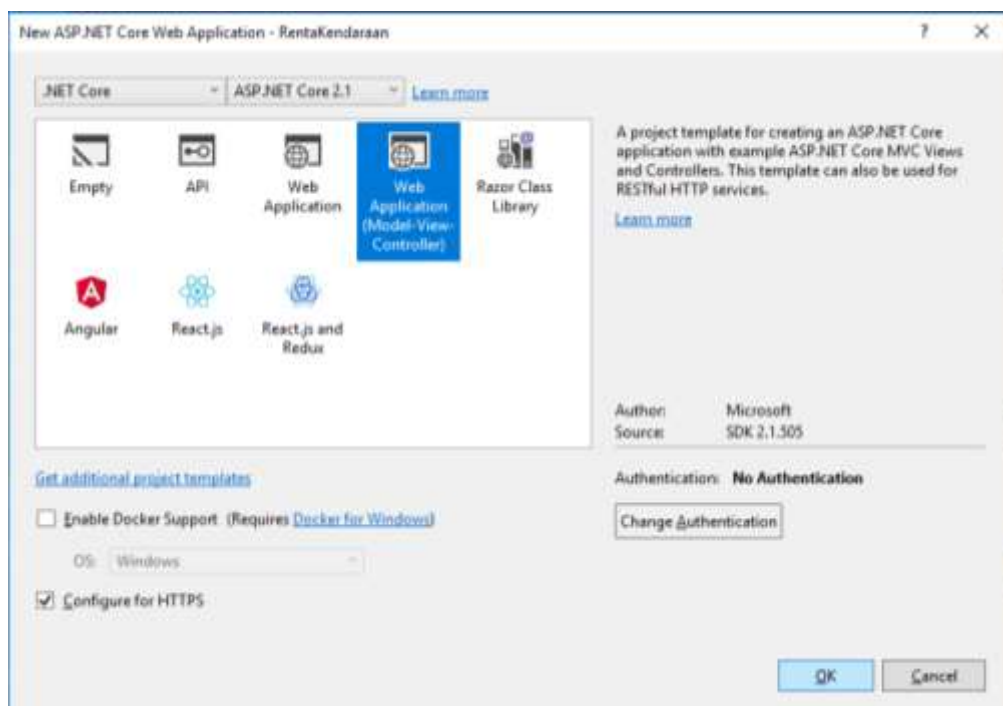
PERTEMUAN 4 – ASP.NET CORE MVC DENGAN DATABASE FIRST PART 2 : GENERATE MVC (MODEL, VIEW, CONTROLLER)

1. Buat sebuah project baru **ASP.NET Core Web Application**, seperti pada gambar 5.1. Klik File > New > Project > Pilih ASP.Net Core Web Application > Beri Nama “RentalKendaraan_NIM” > OK



Gambar 5. 1 Membuat Proyek Baru

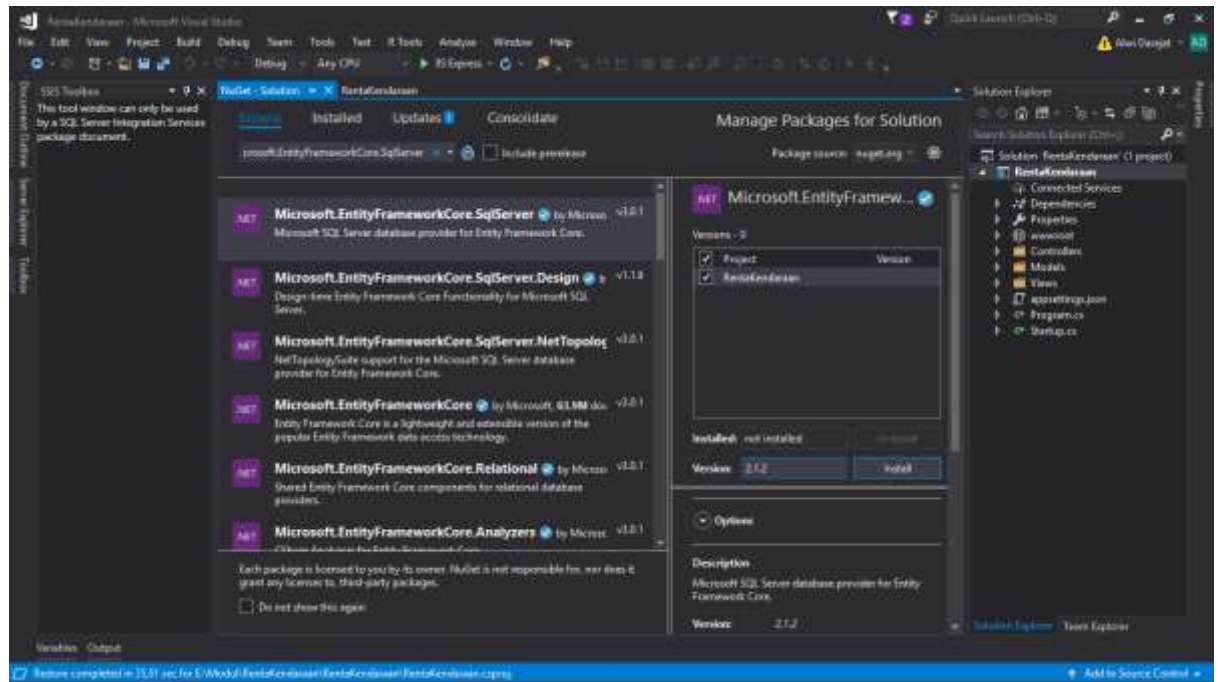
2. Kemudian pilih **Web Application (Model-View-Controller)** dengan **No Authentication** > OK, seperti seperti pada gambar 5.2.



Gambar 5. 2 Pilih Web Application

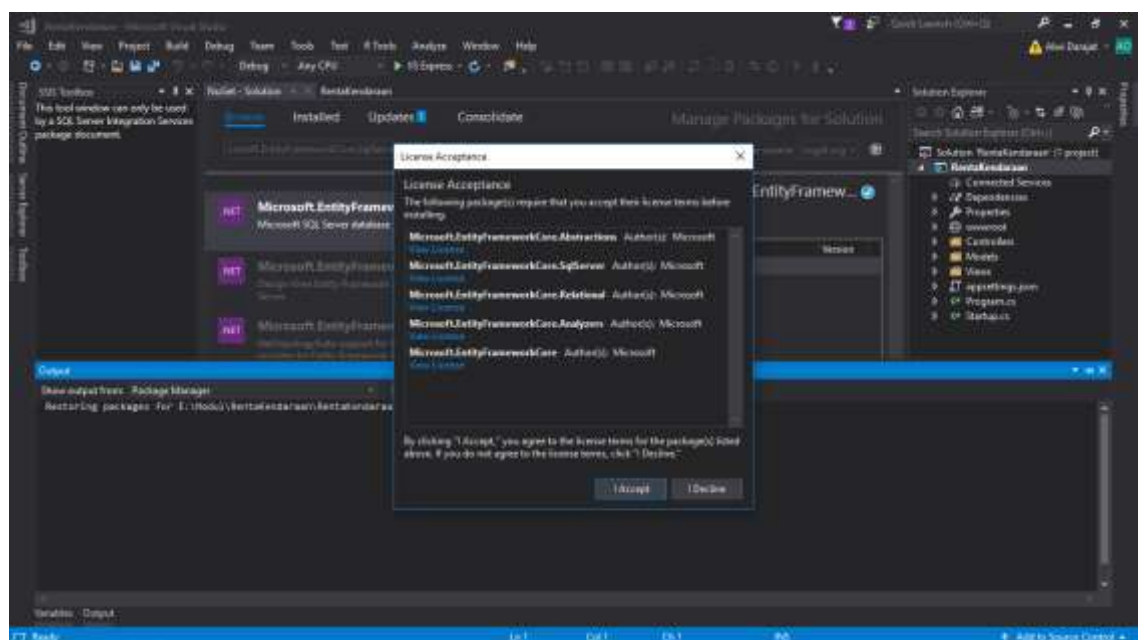
3. Jika project sudah terbuat proyek tetap dapat dijalankan dengan menekan **F5** dengan tampilan default, selanjutnya masuk ke menu **Tools->Nuguet Package Manager->Nuguet - Solution** dan install tiga buah references seperti gambar 5.3 dibawah ini :

➤ **Install-Package Microsoft.EntityFrameworkCore.SqlServer**
Checklist Project > Pilih Version 2.1.2 > Install



Gambar 5. 3 Menambahkan Package Nuguet

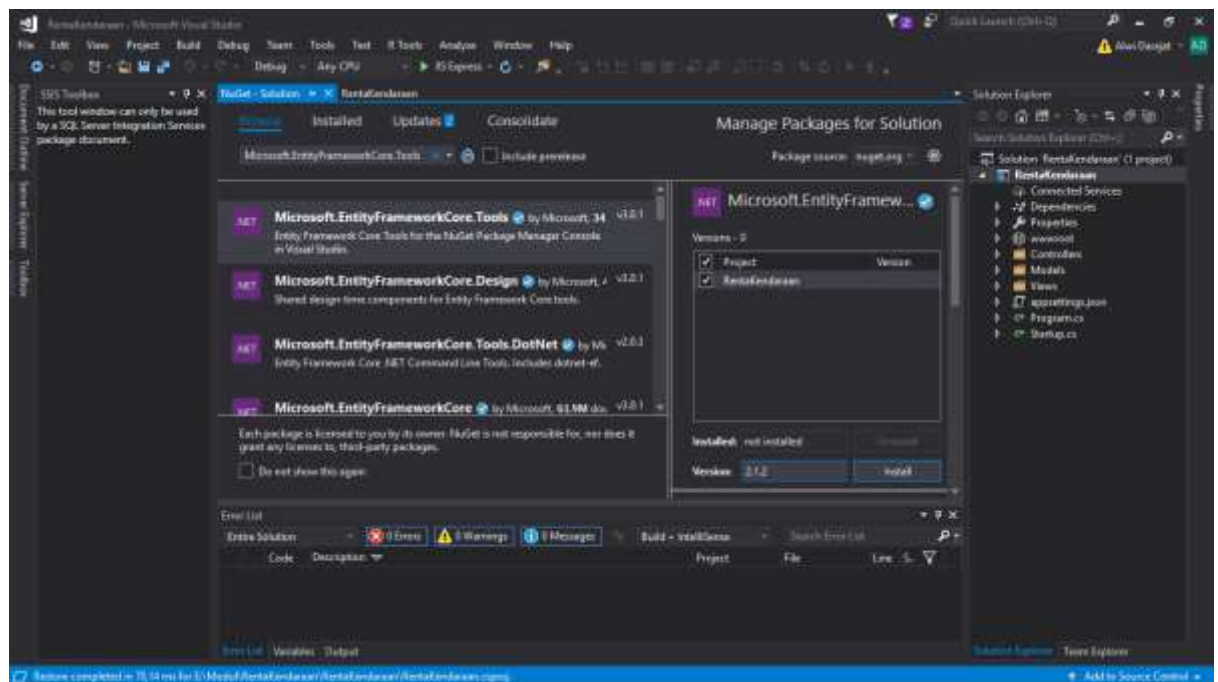
Klik I Accept



Gambar 5. 4 Klik I Accept

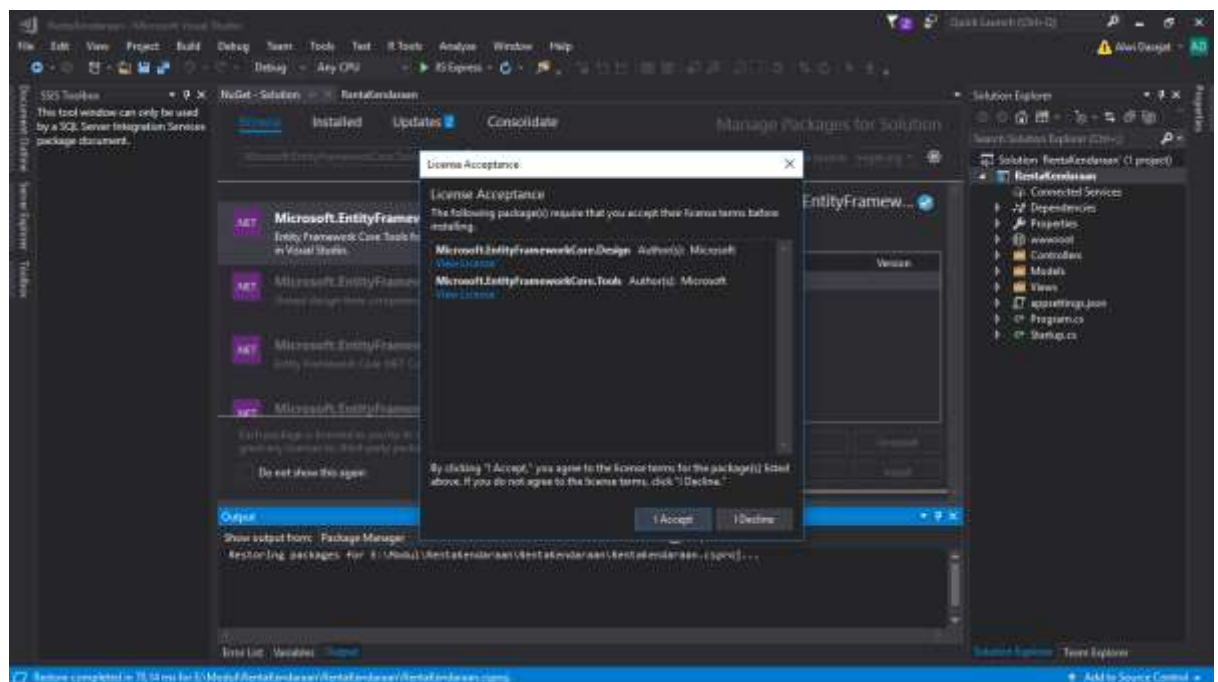
➤ Install-Package Microsoft.EntityFrameworkCore.Tools

Checklist Project > Pilih Version 2.1.2 > Install



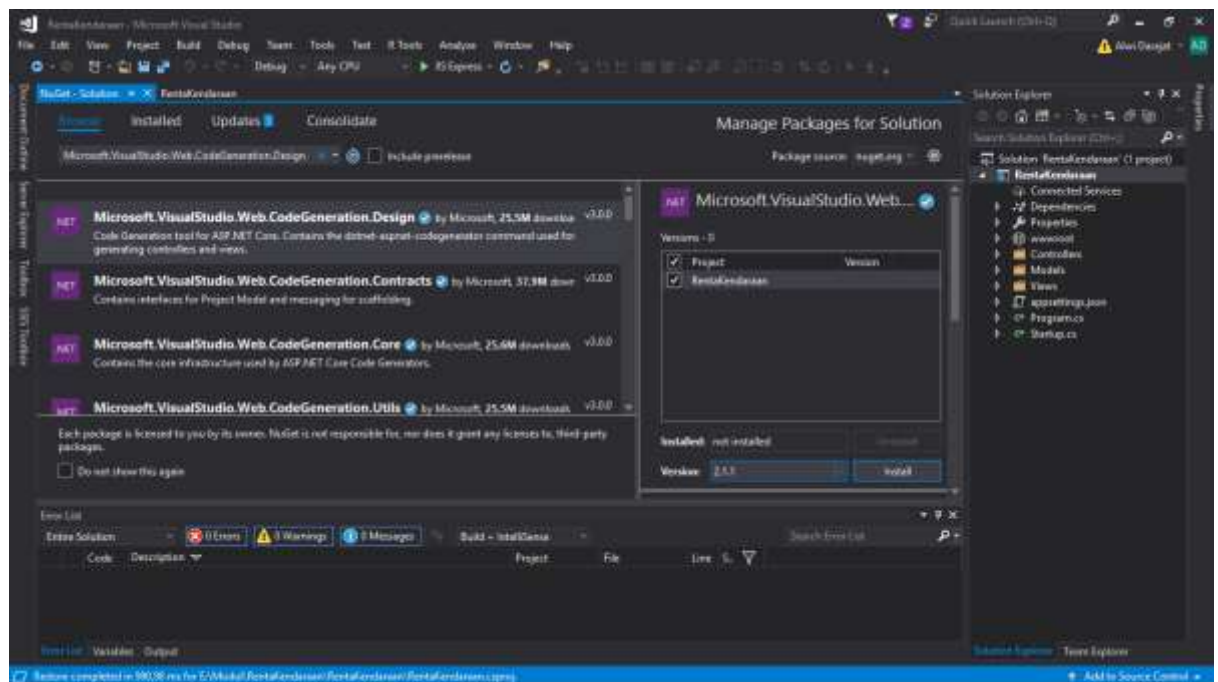
Gambar 5. 5 Menambahkan Package Nugget

Klik I Accept



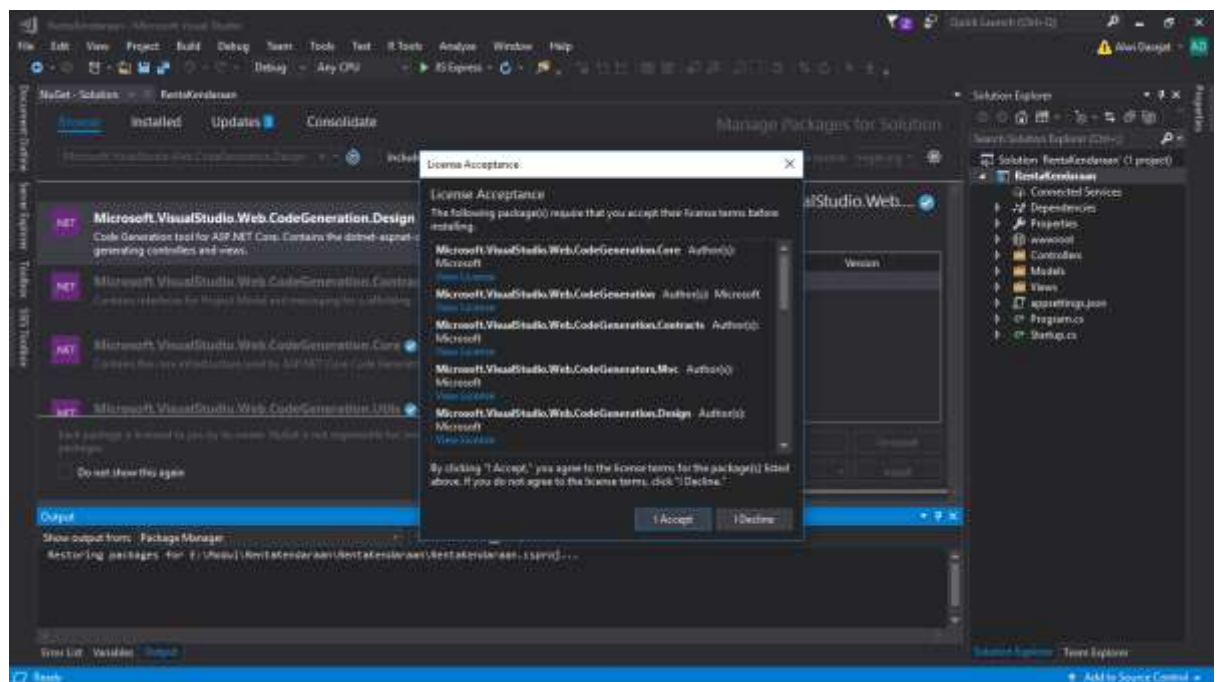
Gambar 5. 6 Klik I Accept

- Install-Package Microsoft.VisualStudio.Web.CodeGeneration.Design
Checklist Project > Pilih Version 2.1.1 > Install



Gambar 5. 7 Menambahkan *Package Nugget*

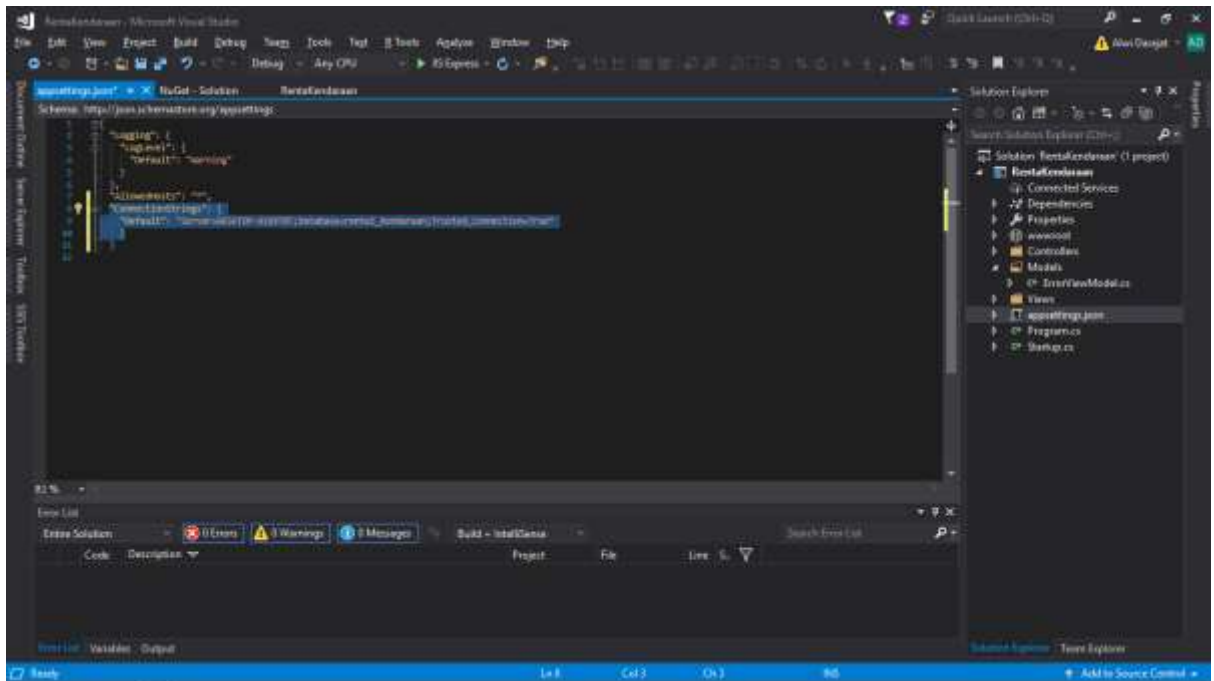
Klik I Accpet



Gambar 5. 8 Klik *I Accpet*

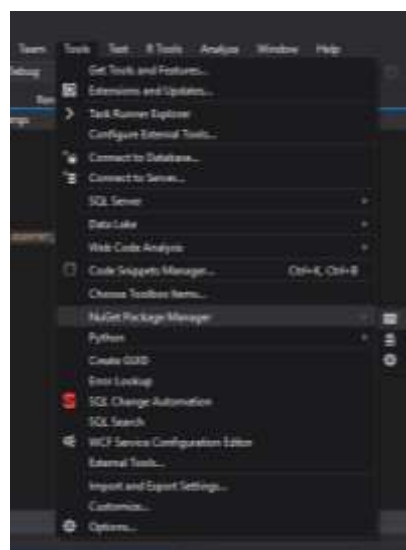
4. Jika sudah sekarang kita mengatur appsettings.json (koneksi database)
Masukan di dalam appsettings.json sesuai dengan nama server dan database di SQLServer anda, seperti pada gambar 5.9. (Pastikan tidak ada error karena ke Enter)

```
"ConnectionStrings": {  
    "Default": "Server=HAIDARALDIW;Database=rental_kendaraan;  
    Trusted_Connection=True"  
}
```



Gambar 5. 9 Mengubah *appsettings.json*

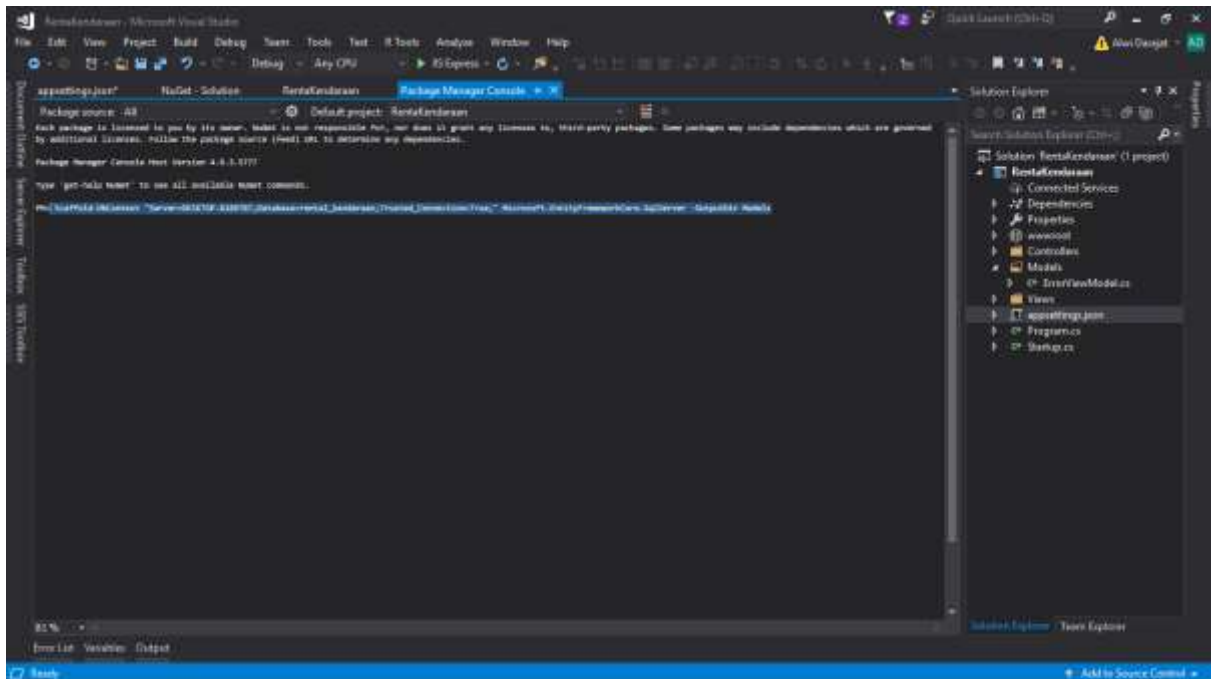
5. Kemudian buat model menggunakan package manager console, seperti gambar 5.10.
Klik Tools > Nuget Package Manager > Package Manager Console



Gambar 5. 10 Membuat Model Dengan *Package Manager Console*

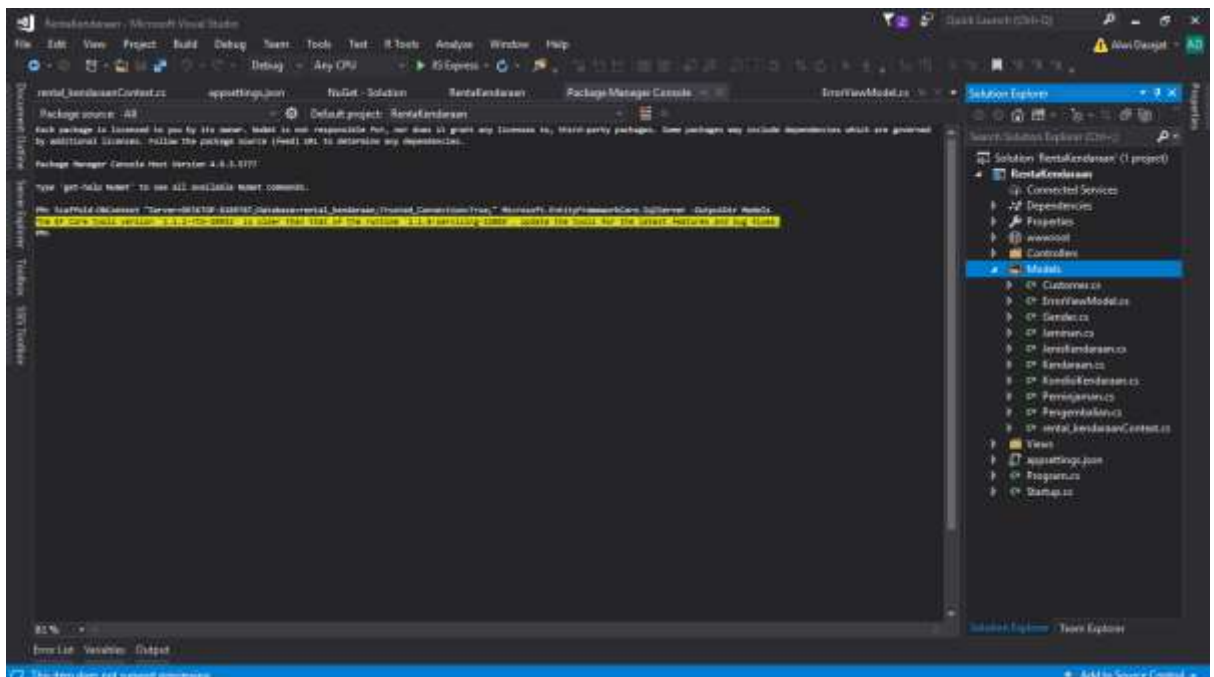
Masukan komen seperti gambar 5.11 dibawah ini (Sesuaikan Server dan Database sesuai yang anda buat):

Scaffold-DbContext "Server=HAIDARALDIW;Database=rental_kendaraan;Trusted_Connection=True;"
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models



Gambar 5. 11 Memasukkan *command* untuk membuat model

Kemudian tekan Enter, tunggu hingga models muncul di dalam folder Models. Seperti gambar 5.12 dibawah ini.



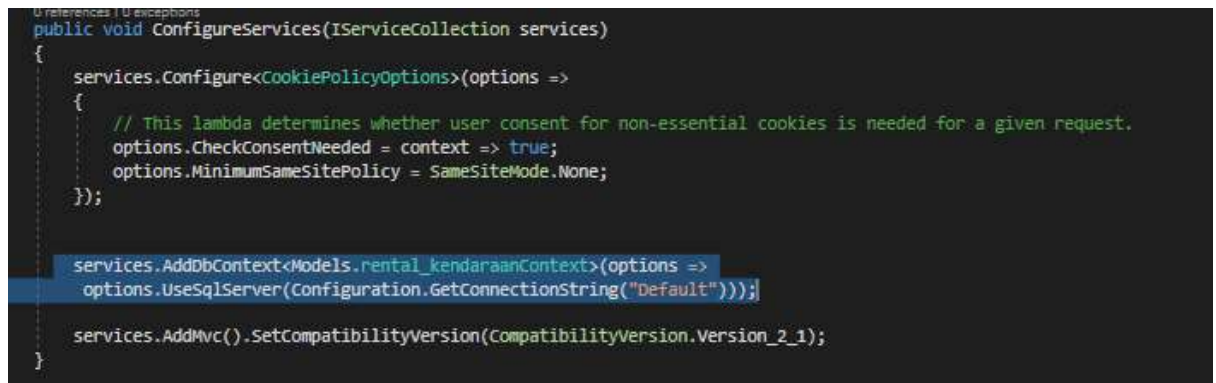
Gambar 5. 12 Hasil Dari Pembuatan Model

6. Hapus code method OnConfiguring di dalam rental_kendaraanContext:

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    if (!optionsBuilder.IsConfigured)
    {
        #warning To protect potentially sensitive information in your connection
        string, you should move it out of source code. See
        http://go.microsoft.com/fwlink/?LinkId=723263 for guidance on storing
        connection strings.
        optionsBuilder.UseSqlServer("Server=DESKTOP-
        8189TBT;Database=rental_kendaraan;Trusted_Connection=True;");
    }
}
```

7. Kemudian Tambahkan Code Berikut pada method ConfigureServices dalam Startup.cs, seperti pada gambar 5.13:

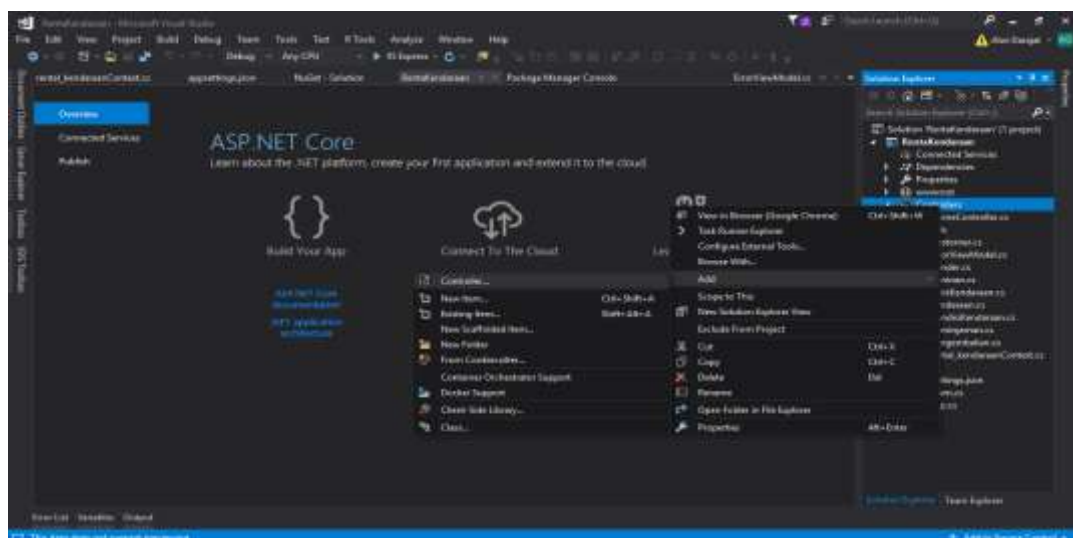
```
services.AddDbContext<Models.rental_kendaraanContext>(options =>
options.UseSqlServer(Configuration.GetConnectionString("Default")));
```



Gambar 5. 13 Menambahkan code di *ConfigureServices*

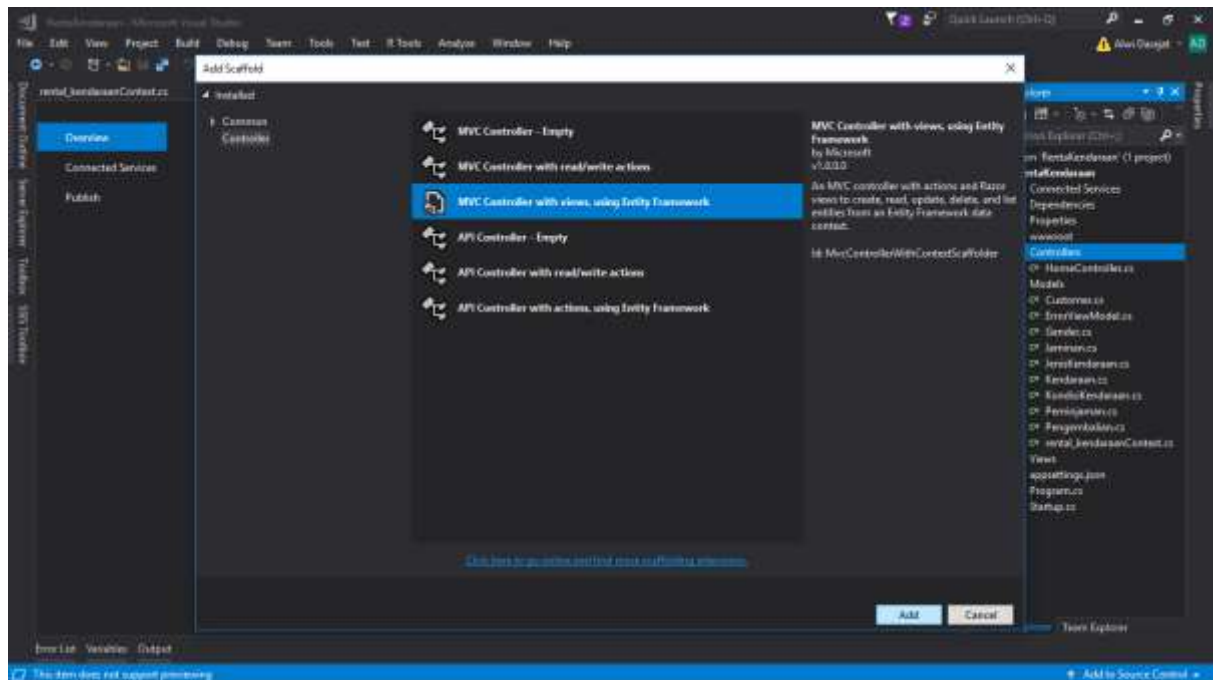
8. Selanjutnya membuat Controller dan View, seperti pada gambar 5.14.

Klik kanan di folder Controller > Add > Controller.



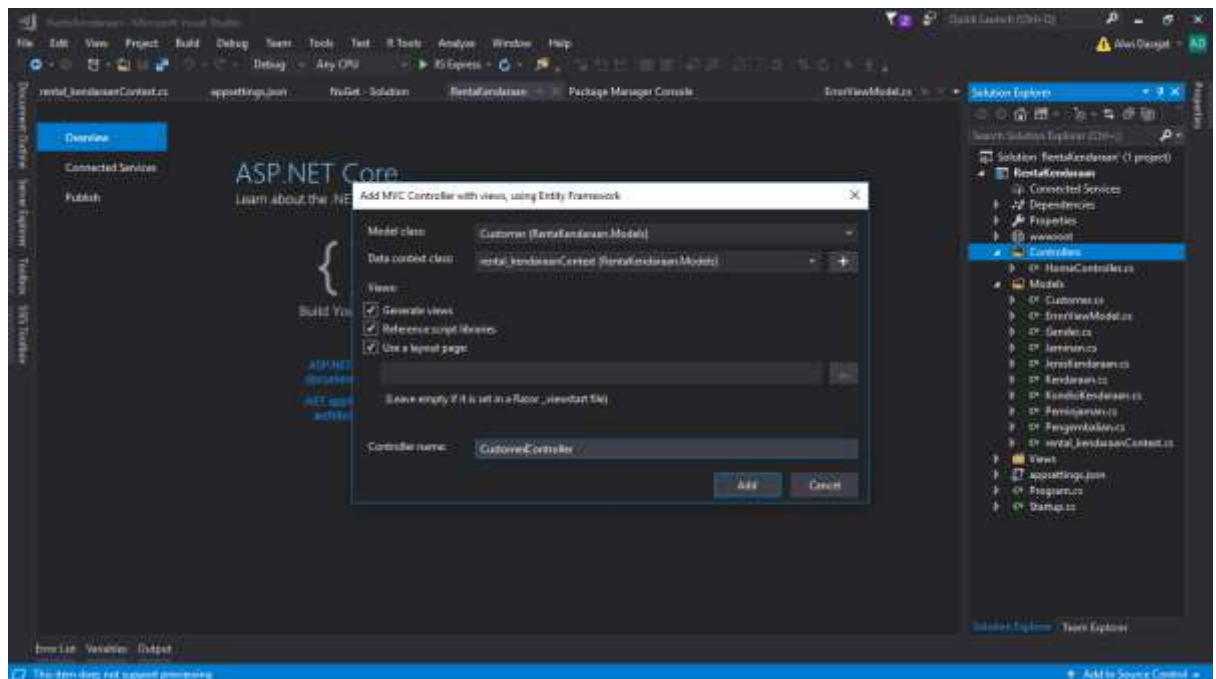
Gambar 5. 14 Menambahkan *Controller*

Pilih MVC Controller with views, using Entity Framework. Klik OK, seperti pada gambar 5.15.



Gambar 5. 15 Pilih MVC Controller with views, using Entity Framework

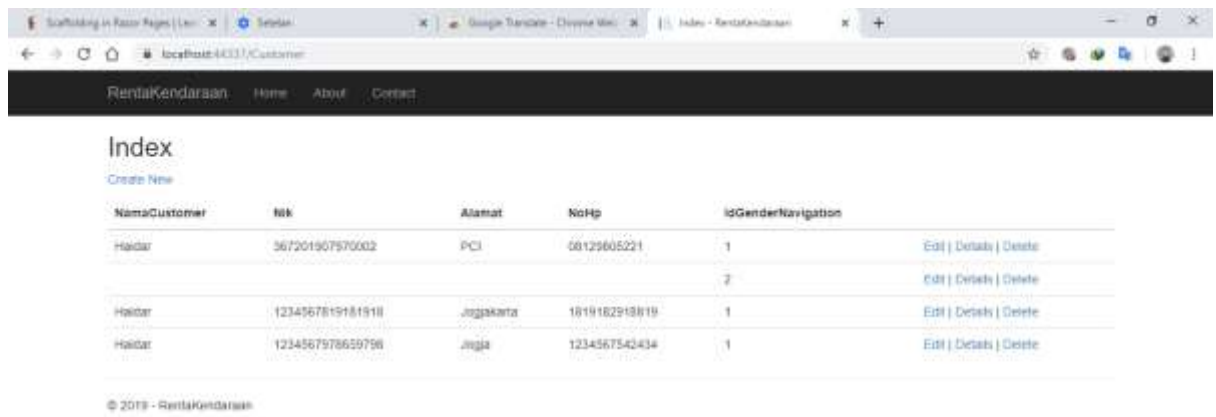
Pada kali ini seperti pada gambar 5.16 akan membuat Controller dan view untuk Customer, Pilih Model Class Customer > Pilih Data context class : rental_kendaraanContext > Add.



Gambar 5. 16 Membuat Controller dan View Customer

Jika Controller dan view Customer berhasil dibuat seperti pada gambar 5.17, silahkan running Index.cshtml dengan cara: klik kanan di Index.cshtml > pilih view in Browser.

Tampilan Index.cshtml



Gambar 5. 17 Tampilan *Website* Ketika Dijalankan

Untuk Model yang lain seperti Gender, Jaminan, JenisKendaraan dan lain-lain cara membuat Controller dan View nya sama saja dengan cara diatas.

Silahkan buat MVC untuk semua Model yang sudah dibuat!