

PRAKTIKUM MINGGU KE-7
MK PENGEMBANGAN APLIKASI WEB
FAKULTAS TEKNIK, PRODI TEKNIK INFORMATIKA
MATERI TENTANG GENERATE MVC
(MODEL, VIEW, CONTROLLER)

A. Tujuan Praktikum

Materi ini bertujuan untuk memfilter dan mencari data agar mudah ditemukan

B. Materi

1. Membuat fitur Filter
2. Membuat fitur search data

C. Alat dan Bahan Praktikum

1. Laptop
2. Kode Editor (Visual Studio 2017)
3. DBMS SQL Server

D. Metode Praktikum

1. Penayangan slide pada proyektor didepan kelas dan pada laptop setiap mahasiswa. Penjelasan terinci setiap slide materi kepada mahasiswa.
2. Mahasiswa beserta Dosen, Asisten/Instruktur melakukan koding bersama untuk membuat program sederhana dengan materi sesuai chapter tersebut.
3. Dosen, Asisten/Instruktur memberikan tugas diakhir sesi dengan studi kasus tertentu guna mereview materi yang telah disampaikan.

E. Tugas

1. Tugas dibuat oleh setiap individu mahasiswa.
2. Pengumpulan tugas dilakukan melalui situs <https://learning.eng.umsida.ac.id/> pada Assignment Tugas 8 mata kuliah Pengembangan Aplikasi Web.
3. Tugas dikumpulkan maksimal sebelum perkuliahan Minggu ke 9.

F. Penilaian

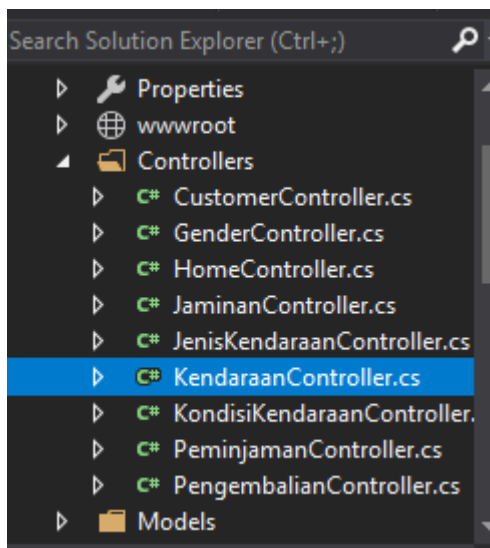
1. Penilaian tugas akan diberikan dari skala 1-100.
2. Tugas yang terlambat akan mengalami pengurangan nilai 10-50% sesuai dengan keterlambatan hari.

PERTEMUAN 7 – ASP.NET CORE MVC DENGAN DATABASE FIRST PART 5 : MEMBUAT FITUR FILTER & SEARCH DATA

a. Membuat Fitur Filter

Salah satu cara untuk mempermudah pencarian data adalah dengan adanya fitur filter data, didalam asp.net core filter data dapat dibuat dengan langkah-langkah sebagai berikut:

1. Pada project rental kendaraan buka folder controllers dan buka kendaraancontroller.cs, seperti pada gambar 8.1.



Gambar 8. 1 Tampilan Folder *Controllers*

2. Kemudian modifikasi code index menjadi seperti pada gambar 8.2 berikut:

```
// GET: Kendaraan
3 references | 0 requests | 0 exceptions
public async Task<IActionResult> Index(string ktsd, string searchString)
{
    //buat list menyimpan ketersediaan
    var ktsdList = new List<string>();
    //query mengambil data
    var ktsdQuery = from d in _context.Kendaraan orderby d.Ketersediaan select d.Ketersediaan;

    ktsdList.AddRange(ktsdQuery.Distinct());

    //untuk nampilkan di view
    ViewBag.ktsd = new SelectList(ktsdList);

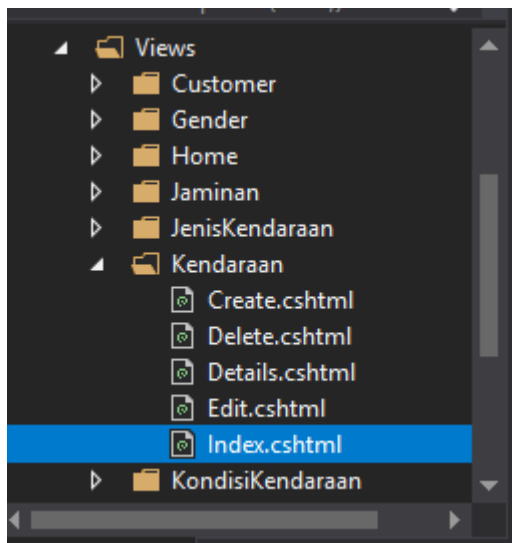
    //panggil db context
    var menu = from m in _context.Kendaraan.Include(k => k.IdJenisKendaraanNavigation) select m;

    //untuk memilih dropdownlist ketersediaan
    if (!string.IsNullOrEmpty(ktsd))
    {
        menu = menu.Where(x => x.Ketersediaan == ktsd);
    }

    return View(await menu.ToListAsync());
}
```

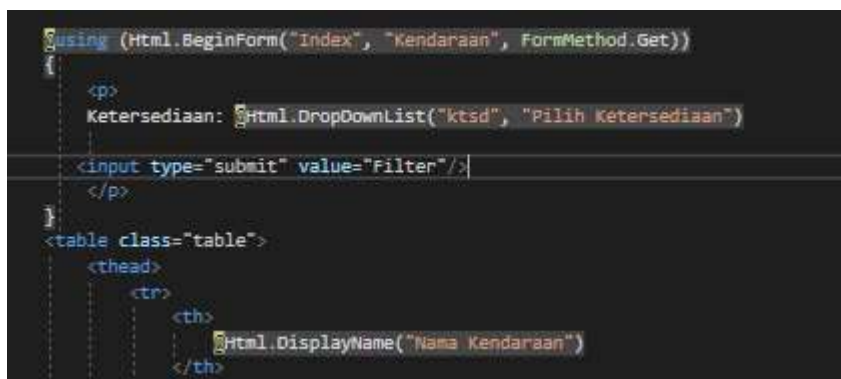
Gambar 8. 2 Menambahkan Fungsi Untuk Membuat Filter

3. Kemudian masuk ke folder views kemudian kendaraan dan index.cshtml, seperti pada gambar 8.3.



Gambar 8. 3 Tampilan Folder Views

4. Kemudian masukkan kode seperti pada gambar 8.4 berikut tepat diatas code table.



Gambar 8. 4 Menambahkan *Dropdown List* di Views

Output yang dihasilkan seperti pada gambar 8.5.

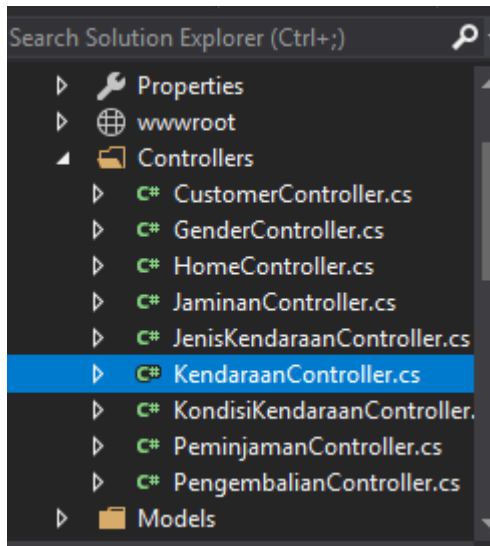


Gambar 8. 5 Tampilan *Dropdown List*

b. Membuat Fitur Search Data

Salah satu cara untuk mempermudah pencarian data adalah dengan adanya fitur filter data, didalam asp.net core filter data dapat dibuat dengan langkah-langkah sebagai berikut:

1. Pada project rental kendaraan buka folder controllers dan buka kendaraancontroller.cs, seperti pada gambar 8.6.



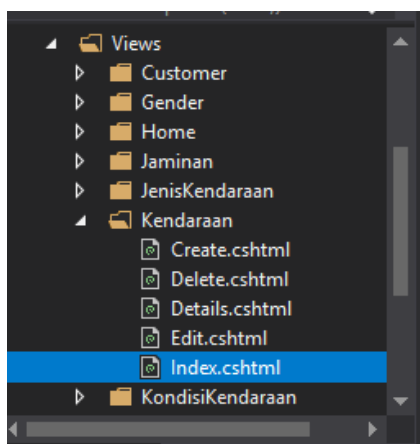
Gambar 8. 6 Tampilan Folder *Controllers*

2. Kemudian tambahkan code seperti pada gambar 8.5 berikut didalam index get kendaraan.

```
//untuk search data
if (!string.IsNullOrEmpty(searchString))
{
    menu = menu.Where(s => s.NoPolisi.Contains(searchString) || s>NamaKendaraan.Contains(searchString)
    || s.NoStnk.Contains(searchString));
}
```

Gambar 8. 7 Menambahkan Fungsi Untuk Mencari Data

3. Kemudian masuk ke folder views kemudian kendaraan dan index.cshtml, seperti pada gambar 8.8.



Gambar 8. 8 Tampilan Folder *Views*

4. Kemudian modifikasi kode filter data menjadi seperti pada gambar 8.9 berikut ini tepat diatas code table.

“Cari Data : @Html.TextBox("searchString")”

```
<a asp-action="Create">Tambah Data Baru</a>

using (Html.BeginForm("Index", "Kendaraan", FormMethod.Get))
{
    <p>
        Ketersediaan: @Html.DropDownList("ktsd", "Pilih Ketersediaan")
        Cari Data : @Html.TextBox("searchString")
        <input type="submit" value="Filter"/>
    </p>
}
```

Gambar 8. 9 Menambahkan *Textbox Search*

Output yang dihasilkan seperti pada gambar 8.10.

Tambah Data Kendaraan

Tambah Data Baru

Ketersediaan: Cari Data:

Nama Kendaraan	No Polisi	No STNK	Ketersediaan	Jenis Kendaraan	
Honda Scoopy	AB 1122 AB	61717171	Tersedia	Motor	Ubah Detail Hapus
Jazz	A 6712 AB	1292911	Tersedia	Mobil	Ubah Detail Hapus
Honda Mobilio	B 3211 AB	1212121	Tersedia	Mobil	Ubah Detail Hapus

© 2019 - Rental Kendaraan HAW19

Gambar 8. 10 Tampilan Mencari Data

Tugas:

Menerapkan fitur filter dan search data pada setiap index didalam project rental kendaraan.

Upload file ke E-Learning dalam bentuk pdf setiap tampilan index disetiap view pada rental kendaraan.

PRAKTIKUM MINGGU KE-8
MK PENGEMBANGAN APLIKASI WEB
FAKULTAS TEKNIK, PRODI TEKNIK INFORMATIKA
MATERI TENTANG GENERATE MVC
(MODEL, VIEW, CONTROLLER)

A. Tujuan Praktikum

Materi ini bertujuan untuk mempersingkat halaman web serta mempermudah pencarian data dan pengurutan data.

B. Materi

1. Fitur Paging Page
2. Fitur Sort Data

C. Alat dan Bahan Praktikum

1. Laptop
2. Kode Editor (visual studio)

D. Metode Praktikum

1. Penayangan slide pada proyektor didepan kelas dan pada laptop setiap mahasiswa. Penjelasan terinci setiap slide materi kepada mahasiswa.
2. Mahasiswa beserta Dosen, Asisten/Instruktur melakukan koding bersama untuk membuat program sederhana dengan materi sesuai chapter tersebut.
3. Dosen, Asisten/Instruktur memberikan tugas diakhir sesi dengan studi kasus tertentu guna mereview materi yang telah disampaikan.

E. Tugas

1. Tugas dibuat oleh setiap individu mahasiswa.
2. Pengumpulan tugas dilakukan melalui situs <https://learning.eng.umsida.ac.id/> pada Assignment Tugas 9 mata kuliah Pengembangan Aplikasi Web.
3. Tugas dikumpulkan maksimal sebelum perkuliahan Minggu ke 9.

F. Penilaian

1. Penilaian tugas akan diberikan dari skala 1-100.
2. Tugas yang terlambat akan mengalami pengurangan nilai 10-50% sesuai dengan keterlambatan hari.

PERTEMUAN 8 – ASP.NET CORE MVC DENGAN DATABASE FIRST PART 6 : MEMBUAT FITUR PAGING PAGE & SORT DATA

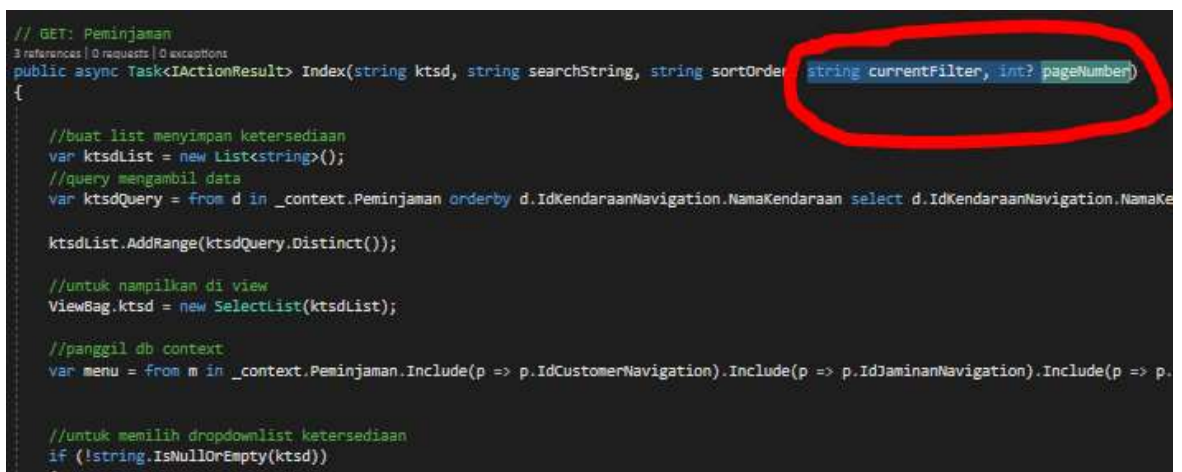
a. Fitur Paging Page

Paging page adalah teknik untuk menampilkan data ke dalam beberapa halaman yang berbeda guna untuk mengurangi kebosanan user ketika harus scrool begitu banyak halaman. Oleh karena itu untuk memecahkan masalah tersebut maka teknik paging page merupakan solusi yang paling tepat.

berikut merupakan langkah-langkah membuat paging page:

Dalam hal ini halaman akan dibatasi oleh 5 data saja.

1. Pada project rental kendaraan buka folder controller dan buka PeminjamanController, seperti pada gambar 9.1.
2. Masukkan parameter didalam method `public async Task<IActionResult> Index(string currentFilter, int? pageNumber,`



```
// GET: Peminjaman
public async Task<IActionResult> Index(string ktsd, string searchString, string sortOrder, string currentFilter, int? pageNumber)
{
    //buat list menyimpan ketersediaan
    var ktsdList = new List<string>();
    //query mengambil data
    var ktsdQuery = from d in _context.Peminjaman orderby d.IdKendaraanNavigation.NamaKendaraan select d.IdKendaraanNavigation.NamaKe

    ktsdList.AddRange(ktsdQuery.Distinct());

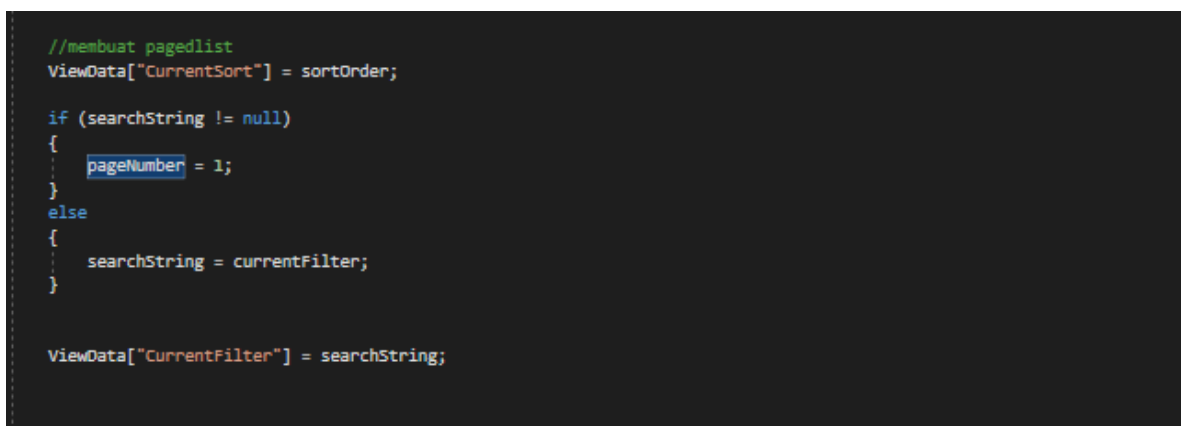
    //untuk nampilkan di view
    ViewBag.ktsd = new SelectList(ktsdList);

    //panggil db context
    var menu = from m in _context.Peminjaman.Include(p => p.IdCustomerNavigation).Include(p => p.IdJaminanNavigation).Include(p => p.

    //untuk memilih dropdownlist ketersediaan
    if (!string.IsNullOrEmpty(ktsd))
    {
```

Gambar 9. 1 Menambahkan Parameter di *Method Index*

3. Kemudian masukkan code berikut, seperti pada gambar 9.2.



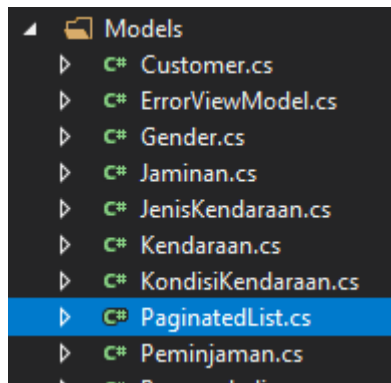
```
//membuat pagedlist
ViewData["CurrentSort"] = sortOrder;

if (searchString != null)
{
    pageNumber = 1;
}
else
{
    searchString = currentFilter;
}

ViewData["CurrentFilter"] = searchString;
```

Gambar 9. 2 Menambahkan Fungsi Untuk Membuat *PagedList*

4. Kemudian didalam folder model buat class baru dengan nama “PaginatedList”, seperti pada gambar 9.3.



Gambar 9. 3 Membuat *Class PaginatedList* didalam Folder *Models*

5. Kemudian masukkan code berikut, seperti pada gambar 9.4.

```
8 references
public class PaginatedList<T> : List<T>
{
    5 references | 0 exceptions
    public int PageIndex { get; private set; }
    1 reference | 0 exceptions
    public int TotalPages { get; private set; }

    1 reference | 0 exceptions
    public PaginatedList(List<T> items, int count, int pageIndex, int pageSize)
    {
        PageIndex = pageIndex;
        TotalPages = (int)Math.Ceiling(count / (double)pageSize);

        this.AddRange(items);
    }

    1 reference | 0 exceptions
    public bool HasPreviousPage
    {
        get
        {
            return (PageIndex > 1);
        }
    }

    1 reference | 0 exceptions
    public bool HasNextPage
    {
        get
        {
            return (PageIndex < TotalPages);
        }
    }

    1 reference | 0 exceptions
    public static async Task<PaginatedList<T>> CreateAsync(IQueryable<T> source, int pageIndex, int pageSize)
    {
        var count = await source.CountAsync();
        var items = await source.Skip((pageIndex - 1) * pageSize).Take(pageSize).ToListAsync();
        return new PaginatedList<T>(items, count, pageIndex, pageSize);
    }
}
```

Gambar 9. 4 Menambahkan Fungsi pada *Class PaginatedList*

6. Kemudian kembali ke dalam peminjamancontroller

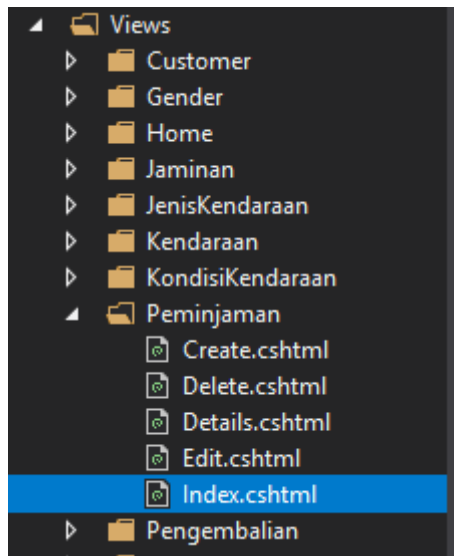
Selanjutnya memasukkan code berikut, seperti pada gambar 9.5.

```
//definisi jumlah data pada halaman
int pageSize = 5;

return View(await PaginatedList<Peminjaman>.CreateAsync(menu.AsNoTracking(), pageNumber ?? 1, pageSize));
```

Gambar 9. 5 Menambahkan kode jumlah data

7. Selanjutnya masuk ke index view peminjaman di dalam project rental kendaraan, seperti pada gambar 9.6.



Gambar 9. 6 Tampilan Folder Views

8. Ganti code

“@model IEnumerable<Rental_Kendaraan_KelasA.Models.Peminjaman>”

Dengan

“@model PaginatedList<Rental_Kendaraan_KelasA.Models.Peminjaman>”

9. Dibawah table Masukkan code berikut, seperti pada gambar 9.7.

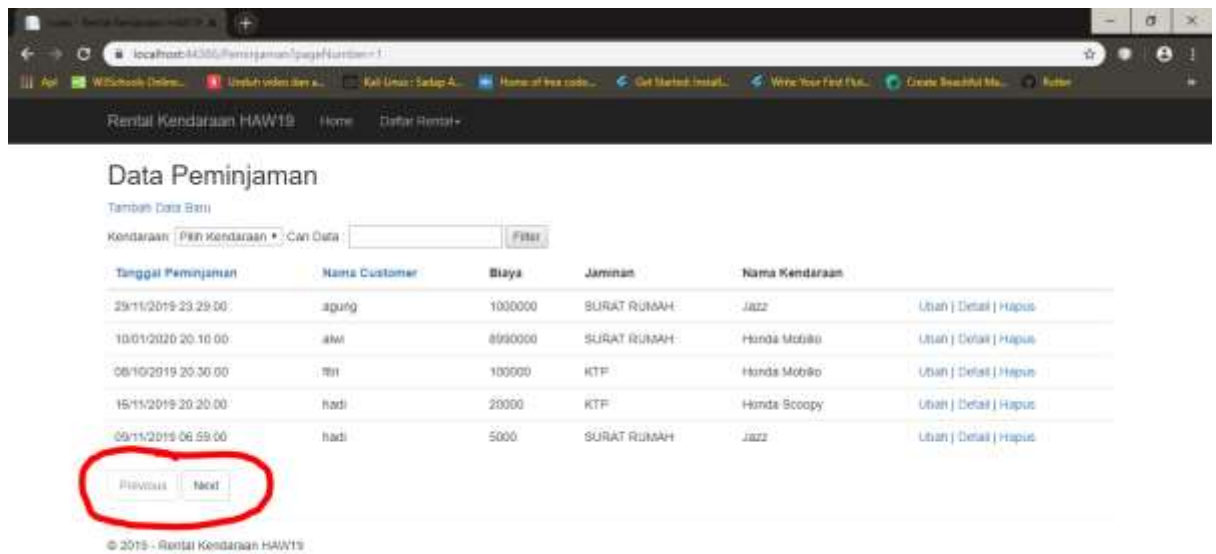
```
</tbody>
</table>

var prevDisabled = !Model.HasPreviousPage ? "disabled" : "";
var nextDisabled = !Model.HasNextPage ? "disabled" : "";

<a asp-action="Index"
  asp-route-sortOrder="@ViewData["CurrentSort"]"
  asp-route-pageNumber="@Model.PageIndex - 1"
  asp-route-currentFilter="@ViewData["CurrentFilter"]"
  class="btn btn-default @prevDisabled">
  Previous
</a>
<a asp-action="Index"
  asp-route-sortOrder="@ViewData["CurrentSort"]"
  asp-route-pageNumber="@Model.PageIndex + 1"
  asp-route-currentFilter="@ViewData["CurrentFilter"]"
  class="btn btn-default @nextDisabled">
  Next
</a>
```

Gambar 9. 7 Menambahkan Kode di Views

Output yang dihasilkan seperti pada gambar 9.8.



Gambar 9. 8 Tampilan *Button PagedList*

b. Fitur Sort Data

1. Pada rental kendaraan buka folder controller dan buka PeminjamanController
2. Masukkan parameter seperti pada gambar 9.9 didalam method `public async Task<IActionResult> Index` string sortOrder.

```
// GET: Peminjaman
// 3 references | 0 requests | 0 exceptions
public async Task<IActionResult> Index(string ktsd, string searchString, string currentFilter, int? pageNumber, string sortOrder)
{
    //buat list menyimpan ketersediaan
    var ktsdList = new List<string>();
    //query mengambil data
    var ktsdQuery = from d in _context.Peminjaman orderby d.IdKendaraanNavigation>NamaKendaraan select d.IdKendaraanNavigation>NamaKend
    ktsdList.AddRange(ktsdQuery.Distinct());

    //untuk nampilkan di view
    ViewBag.ktsd = new SelectList(ktsdList);
}
```

Gambar 9. 9 Menambahkan Parameter di *Method Index*

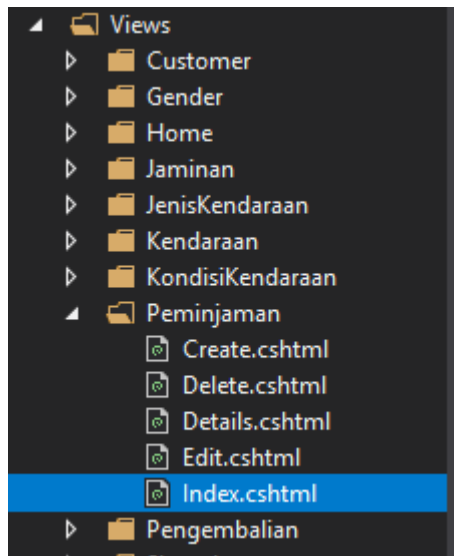
3. Kemudian masukkan code sorting berikut, seperti pada gambar 9.10.

```
//untuk sorting
ViewData["NameSortParam"] = String.IsNullOrEmpty(sortOrder) ? "name_desc" : "";
ViewData["DateSortParam"] = sortOrder == "Date" ? "date_desc" : "Date";

switch (sortOrder)
{
    case "name_desc":
        menu = menu.OrderByDescending(s => s.IdCustomerNavigation>NamaCustomer);
        break;
    case "Date":
        menu = menu.OrderBy(s => s.TglPeminjaman);
        break;
    case "date_desc":
        menu = menu.OrderByDescending(s => s.TglPeminjaman);
        break;
    default: //name ascending
        menu = menu.OrderBy(s => s.IdCustomerNavigation>NamaCustomer);
        break;
}
```

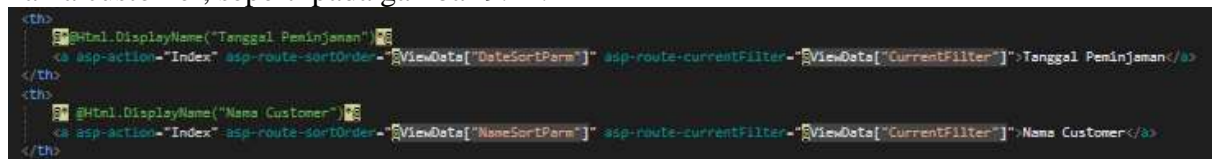
Gambar 9. 10 Menambahkan Fungsi Untuk Membuat *Sorting*

4. Selanjutnya masuk ke index view peminjaman di dalam project rental kendaraan, seperti pada gambar 9.11.



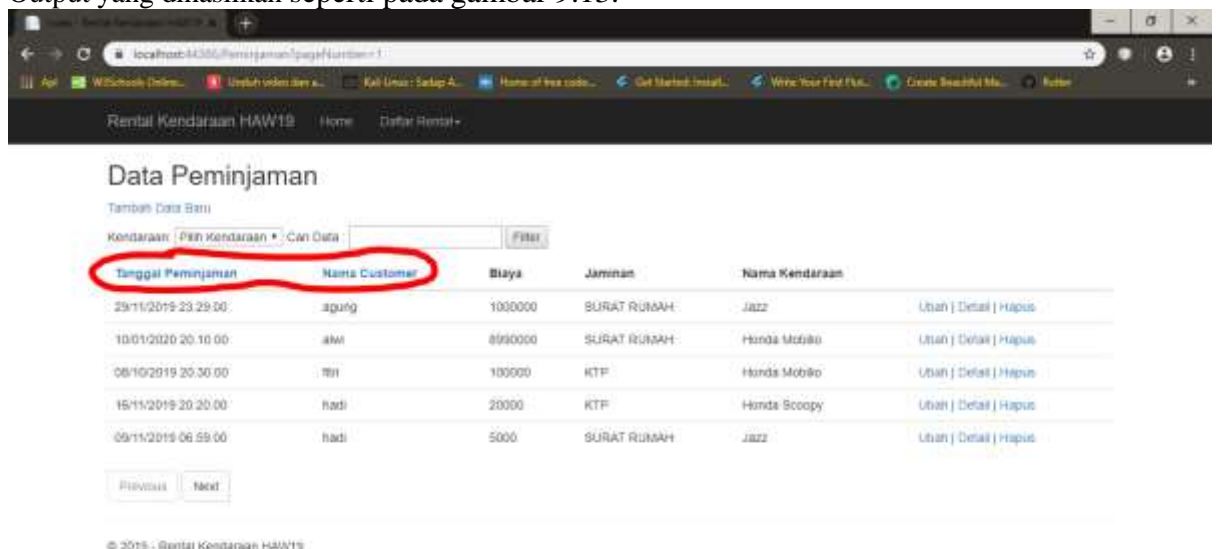
Gambar 9. 11 Tampilan Folder Views

5. Kemudian ubah tag <th> yang bagian tanggal dan ubah tag <th> pada bagian nama customer, seperti pada gambar 9.12.



Gambar 9. 12 Mengubah tag pada Views

Output yang dihasilkan seperti pada gambar 9.13.



Gambar 9. 13 Tampilan Fitur Sorting

Tugas:

Pada project rental kendaraan, berikan paging page dan sort data pada setiap inputan yang ada didalam view sesuai ketentuan yang telah ditentukan.

Upload file ke E-Learning dalam bentuk pdf setiap tampilan validasi disetiap view pada rental kendaraan.