

PRAKTIKUM MINGGU KE-10
MK PENGEMBANGAN APLIKASI WEB
FAKULTAS TEKNIK, PRODI TEKNIK INFORMATIKA
MATERI TENTANG GENERATE MVC
(MODEL, VIEW, CONTROLLER)

A. Tujuan Praktikum

Materi ini bertujuan untuk membuat tempat penguploadan suatu dokumen didalam website.

B. Materi

1. Upload file

C. Alat dan Bahan Praktikum

1. Laptop
2. Visual Studio 2017

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 10.

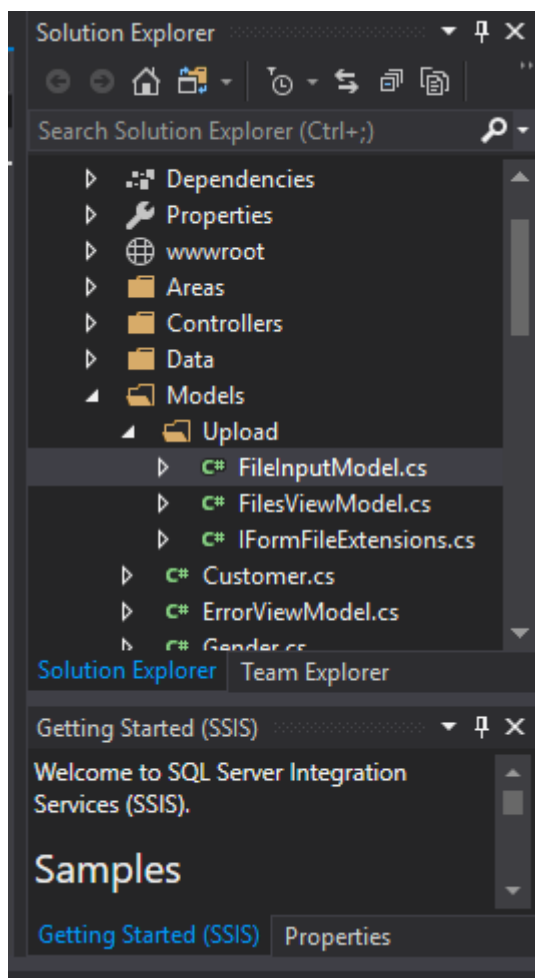
F. Penilaian

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

PERTEMUAN 10 – ASP.NET CORE MVC DENGAN DATABASE FIRST PART 7 : MEMBUAT FITUR UPLOAD FILE

Upload file adalah suatu proses pengiriman data atau file dari perangkat komputer atau perangkat lainnya yang memiliki koneksi internet dengan tujuan upload tertentu. Upload file sudah tidak asing dikalangan internet dan salahsatunya adalah website. Untuk membuat fitur upload file dengan asp.net core mvc lakukan langkah-langkah sebagai berikut.

1. Pada project rental kendaraan seperti pada gambar 10.1 buatlah folder baru dengan nama upload didalam folder models. Kemudian didalam folder upload buat lah 3 buah class dengan nama FileInputModel, FilesViewModel, dan IFormFileExtensions



Gambar 10. 1 Tampilan Folder *Models*

2. Pada class `FileInputModel` masukkan code seperti pada gambar 10.2.

```
namespace Rental_Kendaraan_KelasA.Models.Upload
{
    1 reference
    public class FileInputModel
    {
        4 references | 0 exceptions
        public IFormFile FileToUpload { get; set; }
    }
}
```

Gambar 10. 2 Menambahkan kode pada *Class FileInputModel*

3. Pada class `FilesViewModel` masukkan code seperti pada gambar 10.3.

```
namespace Rental_Kendaraan_KelasA.Models.Upload
{
    3 references
    public class FileDetails
    {
        1 reference | 0 exceptions
        public string Name { get; set; }
        1 reference | 0 exceptions
        public string Path { get; set; }
    }

    1 reference
    public class FilesViewModel
    {
        1 reference | 0 exceptions
        public List<FileDetails> Files { get; set; }
        = new List<FileDetails>();
    }
}
```

Gambar 10. 3 Menambahkan kode pada *Class FilesViewModel*

4. Pada class `IFormFileExtensions` masukkan code seperti pada gambar 10.4.

```
namespace Rental_Kendaraan_KelasA.Models.Upload
{
    0 references
    public static class IFormFileExtensions
    {
        3 references | 0 exceptions
        public static string GetFilename(this IFormFile file)
        {
            return ContentDispositionHeaderValue.Parse(
                file.ContentDisposition).FileName.ToString().Trim('');
        }

        0 references | 0 exceptions
        public static async Task<MemoryStream> GetFileStream(this IFormFile file)
        {
            MemoryStream filestream = new MemoryStream();
            await file.CopyToAsync(filestream);
            return filestream;
        }

        0 references | 0 exceptions
        public static async Task<byte[]> GetFileArray(this IFormFile file)
        {
            MemoryStream filestream = new MemoryStream();
            await file.CopyToAsync(filestream);
            return filestream.ToArray();
        }
    }
}
```

Gambar 10. 4 Menambahkan kode pada *Class IFormFileExtensions*

5. Kemudian buatlah sebuah controller dengan nama UploadController

dan masukkan code berikut

```
public class UploadController : Controller
{
    private readonly IFileProvider fileProvider;
    public UploadController(IFileProvider fileProvider)
    {
        this.fileProvider = fileProvider;
    }
    public IActionResult Index()
    {
        return View();
    }
    [HttpPost]
    public async Task<IActionResult> UploadFile(IFormFile file)
    {
        if (file == null || file.Length == 0)
            return Content("file not selected");

        var path = Path.Combine(
            Directory.GetCurrentDirectory(), "wwwroot",
            file.GetFilename());

        using (var stream = new FileStream(path, FileMode.Create))
        {
            await file.CopyToAsync(stream);
        }

        return RedirectToAction("Files");
    }
    [HttpPost]
    public async Task<IActionResult> UploadFiles(List<IFormFile> files)
    {
        if (files == null || files.Count == 0)
            return Content("files not selected");

        foreach (var file in files)
        {
            var path = Path.Combine(
                Directory.GetCurrentDirectory(), "wwwroot",
                file.GetFilename());

            using (var stream = new FileStream(path, FileMode.Create))
            {
                await file.CopyToAsync(stream);
            }
        }

        return RedirectToAction("Files");
    }
    [HttpPost]
    public async Task<IActionResult> UploadFileViaModel(FileInputModel
model)
    {
        if (model == null ||
            model.FileToUpload == null || model.FileToUpload.Length ==
0)
            return Content("file not selected");
    }
}
```

```

        var path = Path.Combine(
            Directory.GetCurrentDirectory(), "wwwroot",
            model.FileToUpload.GetFilename());

        using (var stream = new FileStream(path, FileMode.Create))
        {
            await model.FileToUpload.CopyToAsync(stream);
        }

        return RedirectToAction("Files");
    }

    public IActionResult Files()
    {
        var model = new FilesViewModel();
        foreach (var item in
            this.fileProvider.GetDirectoryContents(""))
        {
            model.Files.Add(
                new FileDetails { Name = item.Name, Path =
item.PhysicalPath });
        }
        return View(model);
    }

    public async Task<IActionResult> Download(string filename)
    {
        if (filename == null)
            return Content("filename not present");

        var path = Path.Combine(
            Directory.GetCurrentDirectory(),
            "wwwroot", filename);

        var memory = new MemoryStream();
        using (var stream = new FileStream(path, FileMode.Open))
        {
            await stream.CopyToAsync(memory);
        }
        memory.Position = 0;
        return File(memory, GetContentType(path),
            Path.GetFileName(path));
    }

    private string GetContentType(string path)
    {
        var types = GetMimeTypes();
        var ext = Path.GetExtension(path).ToLowerInvariant();
        return types[ext];
    }

    private Dictionary<string, string> GetMimeTypes()
    {
        return new Dictionary<string, string>
        {
            { ".txt", "text/plain" },
            { ".pdf", "application/pdf" },
            { ".doc", "application/vnd.ms-word" },
            { ".docx", "application/vnd.ms-word" },
            { ".xls", "application/vnd.ms-excel" },
            { ".xlsx", "application/vnd.openxmlformats
officedocument.spreadsheetml.sheet" },
        }
    }

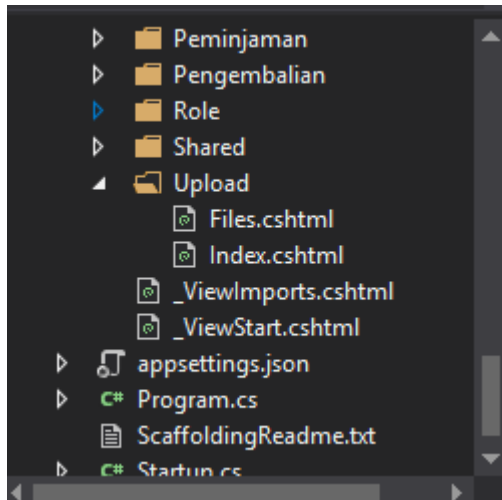
```

```

        {".png", "image/png"},
        {".jpg", "image/jpeg"},
        {".jpeg", "image/jpeg"},
        {".gif", "image/gif"},
        {".csv", "text/csv"}
    };
}

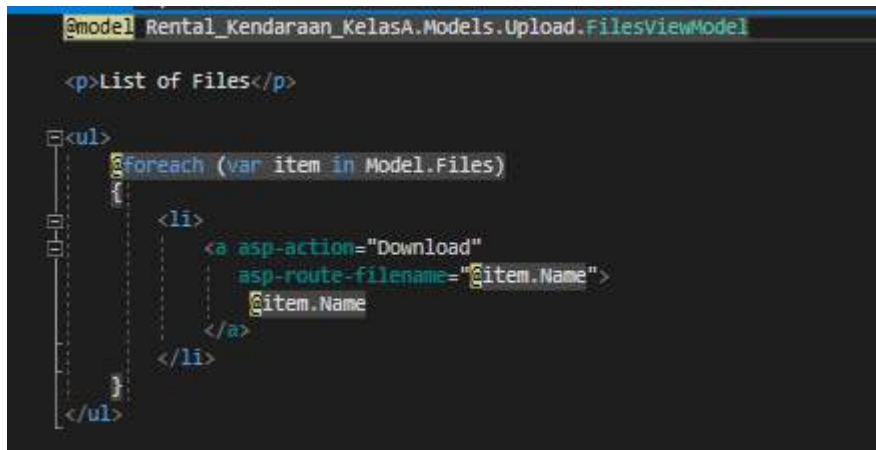
```

6. Selanjutnya adalah membuat view yaitu dengan cara membuat folder upload didalam folder view, seperti pada gambar 10.5.



Gambar 10. 5 Membuat Folder Upload Pada Views

7. Didalam folder upload buatlah 2 buah view dengan nama Files dan Index
 8. Pada view Files masukkan code berikut seperti pada gambar 10.6.



Gambar 10. 6 Menambahkan Kode Untuk Menampilkan *List Upload*

9. Kemudian pada views Index masukkan code berikut seperti pada gambar 10.7.

```

1 <form asp-controller="Upload" asp-action="UploadFile" method="post"
2   enctype="multipart/form-data">
3
4   <input type="file" name="file" />
5   <button type="submit">Upload File</button>
6 </form>
7
8 <form asp-controller="Upload" asp-action="UploadFiles" method="post"
9   enctype="multipart/form-data">
10
11   <input type="file" name="files" multiple />
12   <button type="submit">Upload Files</button>
13 </form>
14
15 <form asp-controller="Upload" asp-action="UploadFileViaModel" method="post"
16   enctype="multipart/form-data">
17
18   <input type="file" name="FileToUpload" />
19   <button type="submit">Upload File (model)</button>
20 </form>

```

Gambar 10. 7 Menambahkan Kode Pada Views Index

10. Kemudian masuk ke Startup.cs pada configureservices masukkan code berikut seperti pada gambar 10.8

```

public void ConfigureServices(IServiceCollection services)
{
    services.AddSingleton<IFileProvider>(
        new PhysicalFileProvider(
            Path.Combine(Directory.GetCurrentDirectory(), "wwwroot")));

    services.Configure<CookiePolicyOptions>(options =>
    {
        // This lambda determines whether user consent for non-essential cookies is needed for a given request.
        options.CheckConsentNeeded = context => true;
        options.MinimumSameSitePolicy = SameSiteMode.None;
    });

    services.AddDbContext<Models.rental_kendaraanContext>(options =>
        options.UseSqlServer(Configuration.GetConnectionString("Default")));
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);

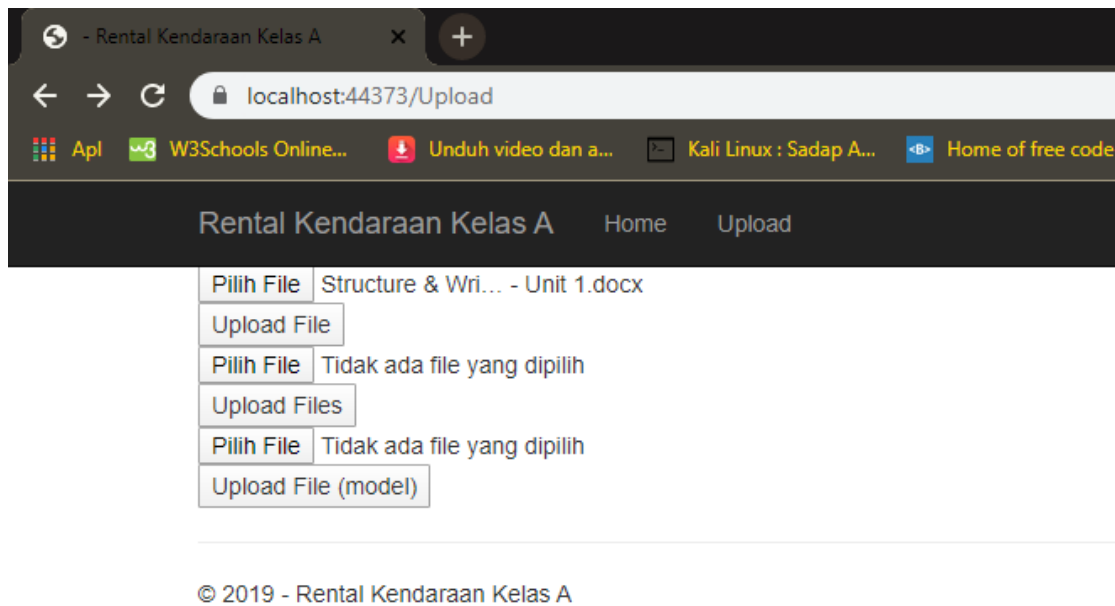
    //services.AddDefaultIdentity<IdentityUser>()
    // .AddEntityFrameworkStores<rental_kendaraanContext>();
    services.AddIdentity<IdentityUser, IdentityRole>().AddDefaultUI()
        .AddEntityFrameworkStores<rental_kendaraanContext>().AddDefaultTokenProviders();

    services.AddAuthorization(options => {
        options.AddPolicy("readonlypolicy",
            builder => builder.RequireRole("Admin", "Manager", "Kasir"));
        options.AddPolicy("writepolicy",
            builder => builder.RequireRole("Admin", "Kasir"));
        options.AddPolicy("editpolicy",
            builder => builder.RequireRole("Admin", "Kasir"));
        options.AddPolicy("deletepolicy",
            builder => builder.RequireRole("Admin", "Kasir"));
    });
    services.AddScoped<Peminjaman>();
    services.AddScoped<Pengembalian>();
}

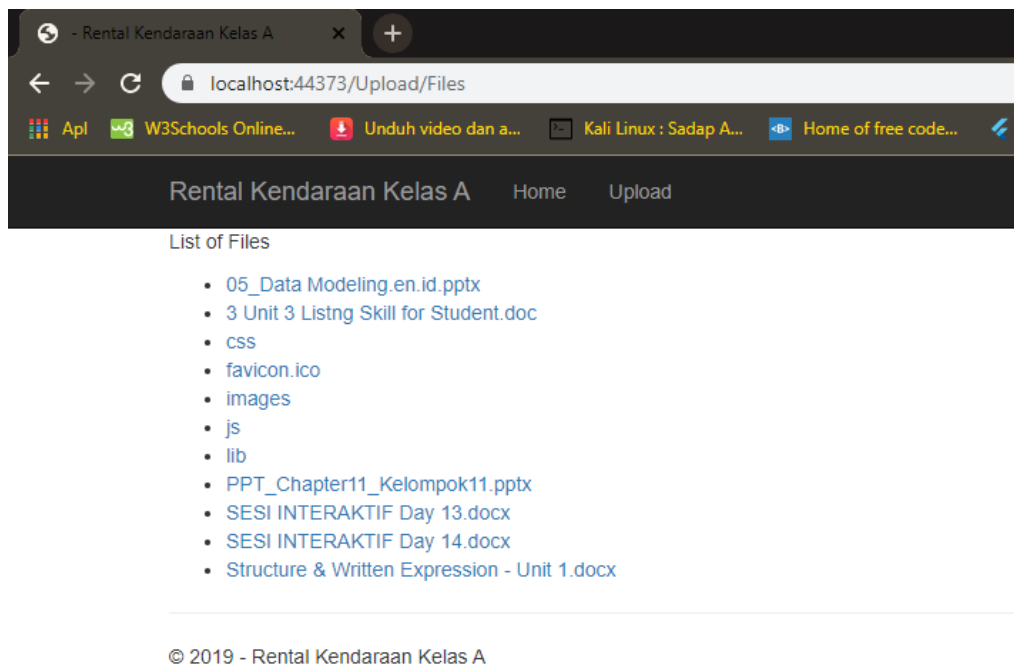
```

Gambar 10. 8 Menambahkan Kode Pada Method ConfigureServices

Hasil output seperti pada gambar 10.9 dan 10.10.



Gambar 10. 9 Tampilan *Upload File*



Gambar 10. 10 Tampilan *List Upload*