

# PEMROGRAMAN WEB LANJUTAN

ALOYSIUS ARI WICAKSONO

# Topik

- ▶ HTML Tag Helper
- ▶ Submit Form

# Pre-Test

- ▶ Buatlah dropdown menggunakan HTML (berisi list jurusan MNP) dengan data berikut
  - ▶ ID : 1, Nama : EComLog
  - ▶ ID : 2, Nama : Animasi
  - ▶ ID : 3, Nama : Event Management
- ▶ Gunakan List<jurusan> di C#, lalu isilah dropdown tersebut dengan menggunakan looping
- ▶ Apa perbedaan dari GET dan POST?

# Pemrograman Web Lanjutan

HTML TAG HELPER

# HTML Tag Helper

- ▶ Jika kita Membuat combobox di HTML

```
<select id="cboJurusan" style="width:100px" class="cboEntry">  
  <option value="1" selected>EComLog</option>  
  <option value="2">Event</option>  
  <option value="3">Animasi</option>  
</select>
```

- ▶ Untuk menggunakan data dari server side :

```
<select id="cboJurusan" style="width:100px" class="cboEntry">  
  @foreach(var jurusan in Model.lstJurusan) {  
    <option value="@jurusan.ID">@jurusan.Nama</option>  
  }  
</select>
```

# HTML Tag Helper

- ▶ Cara tersebut terlalu banyak line yang dibutuhkan
- ▶ Diperlukan HTML Tag Helper untuk mempersingkat coding
- ▶ Dipersingkat jadi seperti ini :

```
@Html.DropDownList("cboJurusan", new SelectList(Model.lstJurusan, "ID", "Nama", "1"), null,  
new { @class="cboEntry", style="width:100px" })
```

# Checkbox

- ▶ Untuk Membuat checkbox bisa menggunakan Html CheckBox.

```
@Html.CheckBox("chkIsActive", isSelected, new { @class="chkEntry" })
```

# TextArea

- ▶ Untuk Membuat checkbox bisa menggunakan Html CheckBox.

```
@Html.TextArea("txtAddress","Gading Serpong", 2, 0, new { @class = "txtEntry", style = "width:100%" })
```



# Pemrograman Web Lanjutan

SUBMIT FORM

# GET vs POST

- ▶ Get : digunakan untuk mengambil data
- ▶ Data terlihat di url (query string)
- ▶ Post : digunakan untuk mengirim data
- ▶ Data tidak terlihat di url
- ▶ Digunakan untuk yang sifatnya mengubah data

# POST

- ▶ Pada ASP.Net Core, untuk Membuat method Post, buat function dengan nama `OnPost{NamaFunction}`
- ▶ Contoh : `OnPostSaveData`
- ▶ Untuk pengiriman data (rekomendasi) : gunakan AJAX Post

# POST

- Untuk memanggil OnPostSaveData, pengiriman di AJAX gunakan handler=SaveData

```
<script src="https://code.jquery.com/jquery-3.7.0.min.js" integrity="sha256-2PmVv0kuTBOenSvLm6bvfbSSHrUJ+3A7x6P5Ebd07/g=" crossorigin="anonymous"></script>
<script type="text/javascript">
    $(function () {
        $('#btnSave').click(function () {
            var data = { Nama: $('#txtNama').val(), NIM: $('#txtNIM').val(), Jurusan: $('#cboJurusan').val(), beasiswa: $('#chkBeasiswa').is(':checked'), Address: $('#txtAddress').val() };
            $.ajax({
                headers: {
                    "Access-Control-Allow-Origin": '*'
                },
                beforeSend: function (xhr) {
                    xhr.setRequestHeader("XSRF-TOKEN",
                        $('#input:hidden[name="__RequestVerificationToken"]').val());
                },
                crossDomain: true,
                url: '@Url.Content("~/")@Request.Path' + "?handler=SaveData",
                data: JSON.stringify(data),
                type: "POST",
                datatype: "json",
                contentType: "application/json; charset=utf-8",
                success: function (response) {
                    console.log(response);
                },
                error: function (XMLHttpRequest, textStatus, errorThrown) {
                    alert(XMLHttpRequest.status);
                }
            });
        });
    });
</script>
```

# POST

- ▶ Akan masuk ke dalam method `OnPostSaveData`.
- ▶ Gunakan debug untuk melihat isi dari `FormData`

```
6 references
public class MahasiswaEntryModel : PageModel
{
    private readonly ILogger<IndexModel> _logger;

    0 references
    public MahasiswaEntryModel(ILogger<IndexModel> logger)
    {
        _logger = logger;
    }

    public List<Jurusan> lstJurusan;
    0 references
    public void OnGet()
    {
        var context = new w6Context();
        lstJurusan = context.Jurusan.ToList();
    }

    0 references
    public IActionResult OnPostSaveData([FromBody] dynamic FormData)
    {
        bool result = true;
        string nama = FormData>Nama;
        return new JsonResult(new { result, nama });
    }
}
```

# Pemrograman Web Lanjutan

SEKIAN DAN TERIMA KASIH