

MudBlazor Input Component

Pada Blazor – Part 1

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Junindar, ST, MCPD, MOS, MCT, MVP

junindar@gmail.com

<http://junindar.blogspot.com>

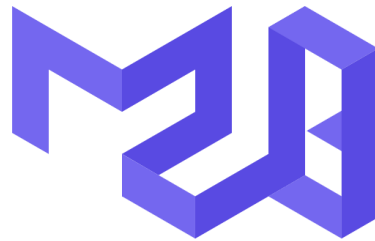
Abstrak

Blazor adalah Web Framework yang bersifat Open Source dimana aplikasi Web yang bersifat client-side interactive dapat dikembangkan dengan menggunakan .Net (C#) dan HTML. Pada saat ini C# biasa digunakan untuk melakukan proses back-end dari aplikasi web. Dengan menggunakan fitur baru dari ASP.NET Core yaitu Blazor, kita dapat membangun interactive WEB dengan menggunakan C# dan .NET .

Code .Net berjalan pada WebAssembly, yang artinya kita dapat menjalankan “NET“ didalam browser (Client) tanpa harus menginstall plugin seperti Silverlight, Java maupun Flash.

Pendahuluan

MudBlazor adalah sebuah material design component framework yang dibangun khusus untuk Blazor (<https://mudblazor.com/>). Terdapat banyak komponen pada MudBlazor seperti chart, grid dan lain-lain untuk membantu dalam membangun aplikasi web dengan menggunakan blazor. Seluruh komponen pada MudBlazor dibangun dengan menggunakan C# tanpa javascript kecuali jika sangat diperlukan. Dan dokumentasi untuk penggunaan MudBlazor ini sangat lengkap sehingga membantu para developer untuk menggunakannya.



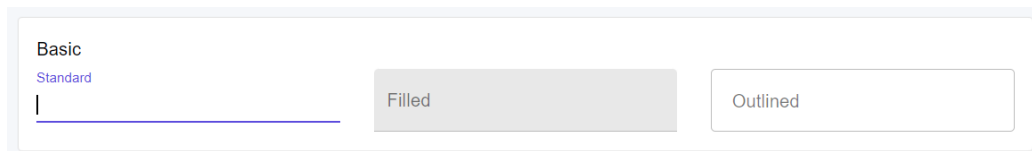
Blazor Components For faster and easier web development

Artikel ini membahas komponen-komponen Input yang terdapat pada MudBlazor. Komponen-komponen tersebut sering digunakan dalam sebuah aplikasi web. Beberapa komponen yang akan dijelaskan pada artikel ini adalah Text Field, Numeric Field dan Select. Dikarenakan artikel ini masih melanjutkan fitur-fitur yang ada pada MudBlazor, maka sebelum memulai latihan-latihan pada artikel ini, pastikan telah menyelesaikan latihan-latihan pada artikel sebelumnya yang dapat dilihat disini (<http://junindar.blogspot.com/2021/12/mudblazor-component-pada-blazor-part-1.html>), (<http://junindar.blogspot.com/2022/07/mudblazor-table-pada-blazor-part-1.html>), (<http://junindar.blogspot.com/2022/07/mudblazor-table-pada-blazor-part-2.html>) dan (<http://junindar.blogspot.com/2022/08/mudblazor-table-pada-blazor-part-3.html>).

- Text Field

Text Field adalah salah satu komponen atau control yang paling sering digunakan dalam sebuah aplikasi web. Text Field yang lebih dikenal dengan Text Box adalah sebuah komponen untuk menerima informasi yang diberikan pengguna. Untuk sintaks pada MudBlazor, Text Field dikenal dengan nama “**MudTextField**“. Terdapat beberapa fitur pada Text Field yang dapat kita gunakan untuk membuat sebuah aplikasi, berikut adalah fitur-fiturnya.

1. Basic



```
<MudTextField @bind-Value="TextValue" Label="Standard"
Variant="Variant.Text" ></MudTextField>
```

```
<MudTextField @bind-Value="TextValue" Label="Filled"
Variant="Variant.Filled"></MudTextField>
```

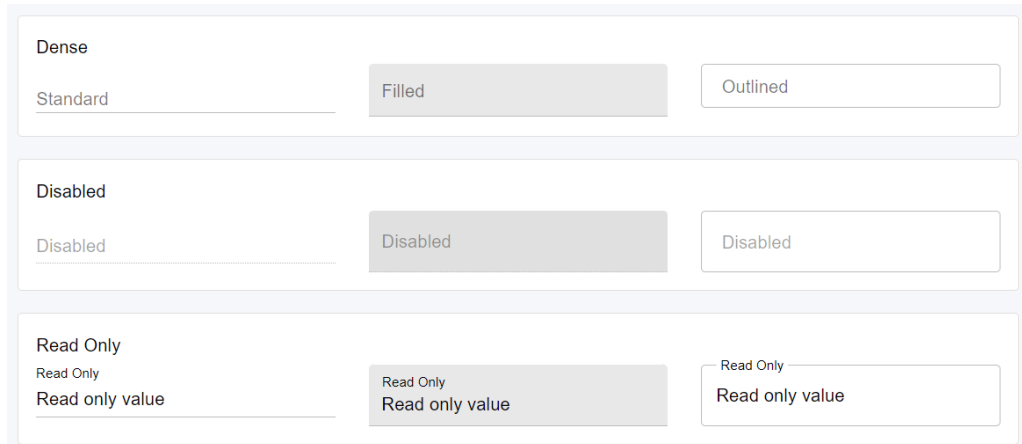
```
<MudTextField @bind-Value="TextValue" Label="Outlined"
Variant="Variant.Outlined"></MudTextField>
```

```
@code {
    public string TextValue { get; set; }
}
```

Terdapat 3 jenis atau varian Text Field pada MudBlazor yaitu Text, Filled dan Outlined. Untuk menggunakan varian-varian diatas, kita gunakan property “**Variant**“ lalu pilih varian yang akan digunakan (**Variant.Text**). Jika tidak secara spesifik menentukan varian apa yang akan digunakan, maka secara otomatis Text Field akan menggunakan varian “**Variant.Text**“.

Selanjutnya adalah property Label, dimana jika Text Field tidak memiliki nilai, maka text dari label akan tampil didalam Text Field, tetapi jika memiliki nilai maka akan ditampilkan di sisi kiri atas Text Field. Dan untuk “mengikat“ suatu nilai dari Text Field kita gunakan property “**@bind-Value**“ dan pastikan kita buat terlebih dahulu variable atau property pada C#.

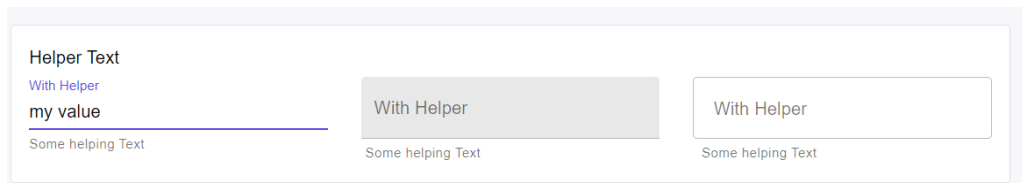
2. Margin, Disabled dan ReadOnly



```
<MudTextField @bind-Value="TextValue" Label="Standard" Variant="Variant.Text" Margin="Margin.Dense"></MudTextField>
```

Seperti textbox pada umumnya, pada MudTextField terdapat fitur untuk mengatur margin, disabled dan ReadOnly. Untuk mengatur vertical spacing agar lebih kecil kita dapat menggunakan property Margin dan atur nilainya menjadi Margin.Dense. Sedangkan untuk menggunakan fitur Disabled dan ReadOnly, kita cukup menambahkan property “Disabled=true” dan “ReadOnly=true” .

3. Helper Text

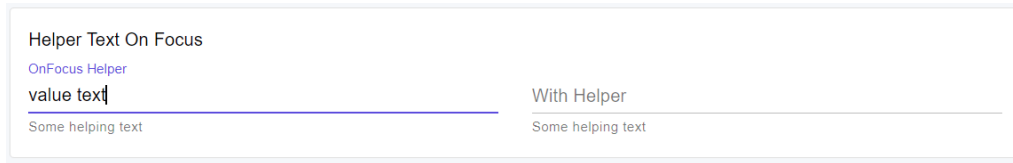


Helper Text adalah salah fitur yang sangat membantu pengguna dalam pengoperasian suatu aplikasi. Karena sesuai dengan namanya Helper Text biasa digunakan untuk memberikan informasi tambahan atau petunjuk kepada pengguna. Sebagai contoh, maksimal jumlah karakter pada sebuah Text Field.

```
<MudTextField @bind-Value="TextValue" Label="With Helper" HelperText="Some helping Text" Variant="Variant.Text" />
```

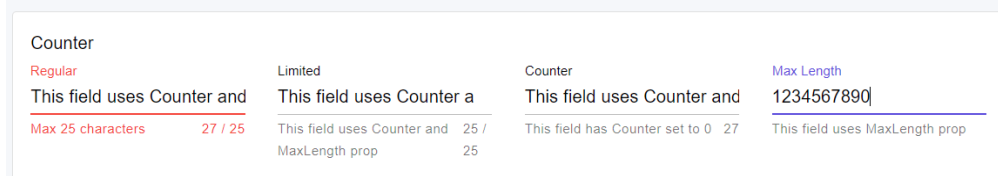
Property yang digunakan untuk menampilkan helper text adalah “HelperText” diikuti dengan text yang ingin ditampilkan. Helper Text akan ditampilkan dibawah dari Text Field.

Jika kita ingin menampilkan Helper Text hanya pada saat Text Field dalam keadaan fokus, maka kita gunakan property “`HelperTextOnFocus="true"`”.



4. Counter

Counter digunakan untuk menampilkan jumlah karakter yang terdapat pada sebuah TextField. Untuk menampilkannya kita gunakan property Counter dan masukkan nilai dari counter yang diinginkan. Posisi dari counter adalah disebelah kanan bawah dari Text Field.



```
< MudTextField T="string" Counter="25" HelperText="This field uses Counter prop"
Immediate="true" Validation="@((new Func<string, IEnumerable<string>>(MaxCharacters))"
Label="Regular" Variant="Variant.Text" />
```

```
< MudTextField T="string" Counter="25" MaxLength="25"
HelperText="This field uses Counter and MaxLength prop"
Immediate="true" Label="Limited" Variant="Variant.Text" />
```

```
< MudTextField T="string" Counter="0" HelperText="This field has Counter set to 0"
Immediate="true" Label="Counter" Variant="Variant.Text" />
```

```
< MudTextField T="string" MaxLength="10" HelperText="This field uses MaxLength prop"
Immediate="true" Label="Max Length" Variant="Variant.Text" />
```

```
@code {
    private IEnumerable<string> MaxCharacters(string ch)
    {
        if (!string.IsNullOrEmpty(ch) && 25 < ch?.Length)
            yield return "Max 25 characters";
    }
}
```

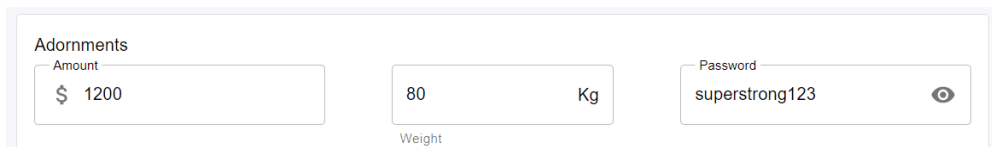
Perhatikan sintask diatas, pada Text Field pertama nilai dari Counter adalah 25, lalu kita gunakan property Validation untuk mengecek jika jumlah karakter melebihi 25 maka akan muncul pesan error pada Helper Text dengan tulisan berwarna merah. Pastikan buat terlebih dahulu method MaxCharacters seperti diatas. Lalu terdapat property “`Immediate`”, dimana jika bernilai true maka akan mengganti value pada saat diketik, jika false nilai diganti hanya jika di enter.

Untuk Text Field kedua, selain menggunakan Counter kita tambahkan property “MaxLength“, sehingga pengguna tidak dapat memasukkan data melebihi jumlah nilai MaxLength.

Jika nilai dari Counter bernilai 0, maka hanya akan ditampilkan jumlah dari karakter pada Text Field saja.

5. Adornments

Adornment adalah Kustom Text atau icon yang terdapat di awal atau akhir pada sebuah Text Field.



```
< MudTextField @bind-Value="Amount" Label="Amount" Variant="Variant.Outlined" Adornment="Adornment.Start" AdornmentIcon="@Icons.Material.Filled.AttachMoney"/>
```

```
< MudTextField @bind-Value="Weight" HelperText="Weight" Variant="Variant.Outlined" Adornment="Adornment.End" AdornmentText="Kg" Class="mx-8" />
```

```
< MudTextField @bind-Value="Password" Label="Password" Variant="Variant.Outlined" InputType="@PasswordInput" Adornment="Adornment.End" AdornmentIcon="@PasswordInputIcon" OnAdornmentClick="ShowPassword" AdornmentAriaLabel="Show Password" />
```

```
@code {  
    public double? Amount { get; set; }  
    public int? Weight { get; set; }  
    public string Password { get; set; } = "superstrong123";  
    bool isShow;  
    InputType PasswordInput = InputType.Password;  
    string PasswordInputIcon = Icons.Material.Filled.VisibilityOff;  
  
    void ShowPassword()  
    {  
        if (isShow)  
        {  
            isShow = false;  
            PasswordInputIcon = Icons.Material.Filled.VisibilityOff;  
            PasswordInput = InputType.Password;  
        }  
        else  
        {  
            isShow = true;  
            PasswordInputIcon = Icons.Material.Filled.Visibility;  
            PasswordInput = InputType.Text;  
        }  
    }  
}
```

Untuk mengatur posisi dari Text atau icon kita gunakan property “Adornment“ lalu pilih posisi yang diinginkan. Seperti berikut “Adornment.Start“ atau “Adornment.End“.

Jika ingin menampilkan Text, kita gunakan property “`AdornmentText`” sedangkan untuk icon property nya adalah “`AdornmentIcon`”.

Lalu perhatikan sintaks pada Text Field ketiga, untuk property “`InputType`”, “`AdornmentIcon`” dan “`OnAdornmentClick`” nilainya menggunakan field dari C#. Seperti pada aplikasi pada umumnya, jika Text Field adalah sebuah Password maka kita tidak akan menampilkan password yang dimasukkan oleh pengguna. Oleh karena itu kita perlu mengganti tipe dari Text Field menjadi “`InputType.Password`”. lalu untuk icon, kita deklarasikan sebuah string dengan nama “`PasswordInputIcon`” yang nilainya diambil dari “`Icons.Material.Filled.VisibilityOff`”. dan terakhir kita buat sebuah method “`ShowPassword`” untuk mengganti tipe dari Text Field menjadi Password maupun Text.

Sedangkan untuk mengganti warna pada text maupun icon, kita gunakan property “`AdornmentColor`” dan masukkan value dari color seperti berikut. “`Color.Info`”.

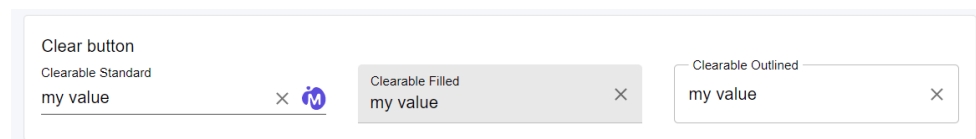
```
< MudTextField @bind-Value="Weight" Label="Weight" Variant="Variant.Outlined" Adornment="Adornment.End" AdornmentText="Kg" AdornmentColor="Color.Info" />
```



6. Clear Button

Clear Button berfungsi untuk menghapus nilai yang terdapat pada Text Field. Untuk menambahkan fitur ini, kita gunakan property “`Clearable`” dengan valuenya adalah true.

```
< MudTextField @bind-Value="TextValue" Label="Clearable Filled" Variant="Variant.Filled" Clearable="true" Immediate="true" />
```



7. Input Type

Untuk TextField pada MudBlazor, terdapat beberapa tipe yang bisa kita gunakan sesuai dengan keperluan. Beberapa tipe yang terdapat pada TextField adalah sebagai berikut.

- Color

```
<MudTextField T="string" Label="Color" InputType="InputType.Color" />
```

- Date

```
<MudTextField T="DateTime?" Format="yyyy-MM-dd" Label="Date" InputType="InputType.Date" />
```

- DateTimeLocal

```
<MudTextField T="DateTime?" Format="s" Label="DateTimeLocal" InputType="InputType.DateTimeLocal" />
```

- Month

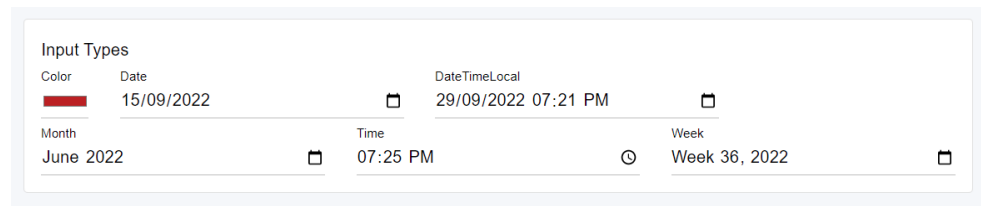
```
<MudTextField T="string" Label="Month" InputType="InputType.Month"/>
```

- Time

```
<MudTextField T="string" Label="Time" InputType="InputType.Time" />
```

- Week

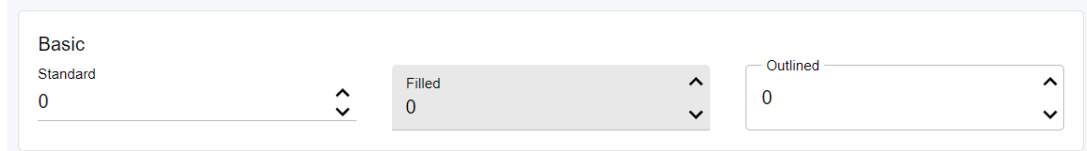
```
<MudTextField T="string" Label="Week" InputType="InputType.Week"/>
```



- Numeric Field

Sama seperti Text Field yang telah dijelaskan sebelumnya, Numeric Field merupakan komponen yang digunakan untuk menerima informasi yang diberikan oleh pengguna. Tetapi untuk komponen ini dibatasi hanya informasi bertipe numerik yang dapat dimasukkan. Berikut merupakan beberapa fitur yang terdapat pada Numeric Field pada MudBlazor.

1. Basic



Pada komponen ini terdapat atribut Min/Max, dimana kita dapat mengatur batas dari nilai yang dapat dimasukkan oleh pengguna. Jika kita tidak mengatur nilai dari Min/Max maka pengguna dapat memasukkan nilai sesuai dengan range dari tipe datanya. Sebagai contoh jika integer maka pengguna dapat memasukkan nilai dari -2147483648 sampai 2147483647.

Selain Min/Max, terdapat juga atribut Step, dimana kita dapat mengatur penambahan atau pengurangan nilai, jika pengguna menggunakan spin button pada Numeric Field.

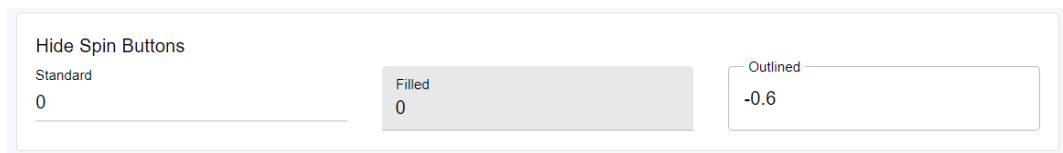
```
<MudNumericField @bind-Value="IntValue" Label="Standard" Variant="Variant.Text" Min="0" Max="10" />
```

```
<MudNumericField @bind-Value="DoubleValue" Label="Filled" Variant="Variant.Filled" Min="0.0" />
```

```
<MudNumericField @bind-Value="DecimalValue" Label="Outlined" Variant="Variant.Outlined" Step=".2M" />
```

```
@code {  
    public int IntValue { get; set; }  
    public double DoubleValue { get; set; }  
    public decimal DecimalValue { get; set; }  
}
```

Untuk menghilangkan spin button pada Numeric Field, kita dapat menggunakan atribut “HideSpinButtons=true”, sehingga mendapatkan hasil seperti pada gambar dibawah ini.



2. Setting Culture dan Format

Dengan menggunakan atribut “Format” dan “Culture“, kita dapat mengatur CultureInfo dan format secara khusus sesuai dengan keperluan kita masing-masing.

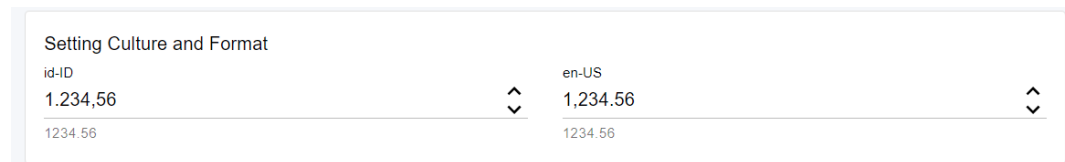
```
<MudNumericField Immediate="false" Label="id-ID" Format="N2" Culture="@_id"
T="double?" @bind-Value="_valueId" HelperText="@_valueId.ToString()" />

<MudNumericField Immediate="false" Label="en-US" Format="N2" Culture="@_en"
T="double?" @bind-Value="_valueEn" HelperText="@_valueEn.ToString()" />

@code {

    public CultureInfo _id = CultureInfo.GetCultureInfo("id-ID");
    public CultureInfo _en = CultureInfo.GetCultureInfo("en-US");

    public double? _valueId = 1234.56;
    public double? _valueEn = 1234.56;
}
```



3. Value Type vs Nullable

Jika kita melakukan “binding“ sebuah variable seperti integer, double dan lain-lain. Untuk pertama kali Numeric Field tidak akan kosong, meskipun pengguna belum memasukkan nilai karena variable-variable diatas selalu memiliki default value. Sebagai contoh untuk integer default value-nya 0.

Jadi jika kita ingin Numeric Field ini kosong, selama tidak ada yang dimasukkan atau setelah valuenya dihapus, maka tipe data yang kita gunakan adalah Nullable. (int?, double?). Pastikan kita tidak mengatur untuk atribu Min, Max, dan Step jika menggunakan tipe data ini.

```
<MudNumericField @bind-Value="intValue" Label="Enter an int"
Variant="Variant.Outlined" />

<MudNumericField @bind-Value="doubleValue" Label="Enter a double" Format="F1"
Variant="Variant.Outlined" />

<MudNumericField @bind-Value="nullableInt" Label="Enter an int"
Variant="Variant.Outlined" />

<MudNumericField @bind-Value="nullableDouble" Label="Enter a double" Format="F1"
Variant="Variant.Outlined" />

@code {

    int intValue;
    double doubleValue;
    int? nullableInt;
    double? nullableDouble;
}
```



4. Mouse Wheel

Pada Numeric Field terdapat sebuah fitur dimana pengguna dapat mengubah nilai dengan menggunakan Shift + MouseWheel. Dan terdapat sebuah atribut InvertMouseWheel untuk membalikkan proses MouseWheel ke atas dan ke bawah. Jika bernilai true, maka jika scroll pada mouse keatas nilai akan bertambah, jika kebawah maka akan berkurang. Dan kebalikkannya jika bernilai false.

```
<MudNumericField @bind-Value="@_normal" InvertMouseWheel="@_invertMouseWheel"
Label="Use Shift + Mouse Wheel" Variant="Variant.Outlined" />

<MudCheckBox Class="mt-3" @bind-Checked="@_invertMouseWheel" Label="Revert Mouse
Wheel" Color="Color.Primary" />

@code {
    double _normal = 0;
    bool _invertMouseWheel = false;
}
```



- Select

Select adalah sebuah komponen yang memungkinkan pengguna memberikan informasi untuk memilih dari daftar opsi. Seperti pada Text Field komponen ini juga memiliki varian untuk tampilannya.

1. Basic

Daftar opsi pada Select bersifat generik, sehingga kita dapat menggunakan value dari tipe apapun. Seperti contoh dibawah, terdapat 3 buah Select yang menggunakan tipe yang berbeda-beda. (string, double dan object)

The screenshot shows a Blazor form with three dropdown menus. The first dropdown is labeled 'Coffee' and has 'Coffee' selected. The second dropdown is labeled 'Price' and has '0' selected. The third dropdown is labeled 'Pizza' and has 'Pizza' selected.

```
<MudSelect T="string" Label="Coffee" AnchorOrigin="Origin.BottomCenter">
  <MudSelectItem Value="@(" Cappuccino")" />
  <MudSelectItem Value="@(" Cafe Latte")" />
  <MudSelectItem Value="@(" Espresso")" />
</MudSelect>

<MudSelect T="double" Label="Price" Variant="Variant.Outlined"
AnchorOrigin="Origin.BottomCenter">
  <MudSelectItem T="double" Value="4.50" />
  <MudSelectItem T="double" Value="4.99" />
  <MudSelectItem T="double" Value="3.60" />
</MudSelect>

<MudSelect T="Pizza" Label="Pizza" Variant="Variant.Filled"
AnchorOrigin="Origin.BottomCenter">
  <MudSelectItem Value="@("new Pizza("Cardinale"))" />
  <MudSelectItem Value="@("new Pizza("Diavolo"))" />
  <MudSelectItem Value="@("new Pizza("Margarita"))" />
  <MudSelectItem Value="@("new Pizza("Spinaci"))" />
  <MudSelectItem Value="@("new Pizza("Spinaci"))" />
</MudSelect>

@code {
  public class Pizza
  {
    public Pizza(string name)
    {
      Name = name;
    }
    public readonly string Name;
    public override string ToString() => Name;
  }
}
```

2. Multiselect

Pada komponen ini kita dapat memilih lebih dari satu pilihan. Untuk menggunakan fitur ini, kita perlu menggunakan atribut “MultiSelection=true”.

The screenshot shows a Blazor Multiselect component. The dropdown menu is open, showing 'Alaska, American Samoa' as the selected value. Below the dropdown, the 'Value:' field shows '2 states have been selected'. The 'SelectedValues: HashSet<string>' field shows a list containing 'Alaska' and 'American Samoa'.

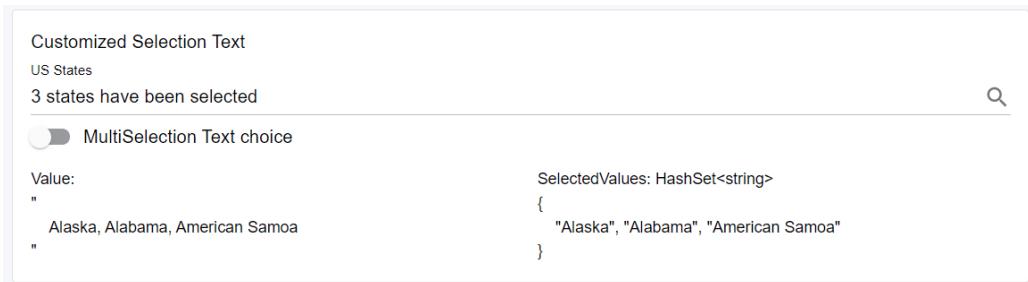
Pada contoh diatas, akan ditampilkan daftar dari opsi-opsi yang telah dipilih dari pengguna. Disini kita juga dapat melakukan kustomisasi text dari opsi-opsi yang telah dipilih tersebut. Sehingga pada Select tidak menampilkan opsi-opsi yang dipilih. Disini kita gunakan atribut “MultiSelectionTextFunc”.

```
<MudSelect MultiSelectionTextFunc="@((new Func<List<string>,
string>(GetMultiSelectionTextChoice))" MultiSelection="true"
@bind-Value="value" @bind-SelectedValues="options" T="string" Label="US States"
AdornmentIcon="@Icons.Material.Filled.Search" AnchorOrigin="Origin.BottomCenter">
@foreach (var state in states)
{
    <MudSelectedItem T="string" Value="@state">@state</MudSelectedItem>
}
</MudSelect>

<MudSwitch @bind-Checked="multiselectionTextChoice" Class="mud-width-full"
Color="Color.Primary">MultiSelection Text choice</MudSwitch>

@code {
    private string GetMultiSelectionTextChoice(List<string> selectedValues)
    {
        if (multiselectionTextChoice)
        {
            return $"Selected state{(selectedValues.Count > 1 ? "s" : "")}:
            {string.Join(", ", selectedValues.Select(x => x))}";
        }
        else
        {
            return $"{{selectedValues.Count}} state{(selectedValues.Count > 1 ?
            "s have" : " has")} been selected";
        }
    }

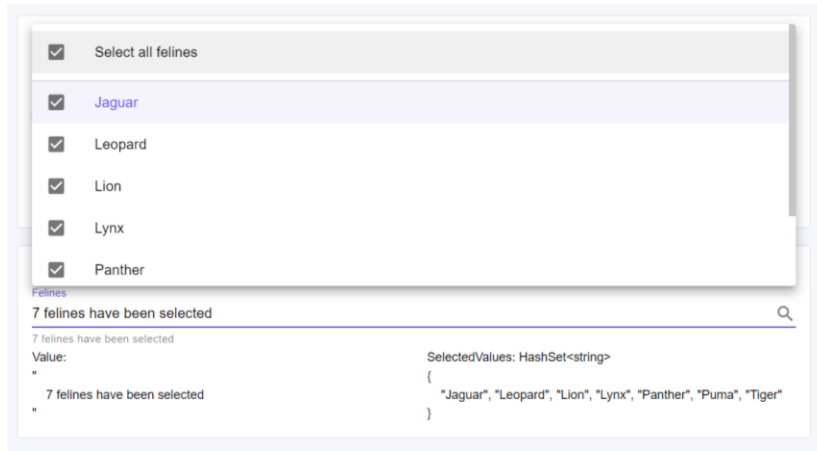
    private bool multiselectionTextChoice;
}
```



3. Select All

Untuk memudahkan pengguna dalam memilih seluruh daftar opsi pada select, kita dapat menggunakan atribut `SelectAll="true"`. Lalu untuk mengatur text pada checkbox select all, kita gunakan atribut `SelectAllText`.

```
<MudSelect T="string" MultiSelection="true" SelectAll="true"  
SelectAllText="Select all felines" HelperText="@value"  
@bind-Value="value" @bind-SelectedValues="options2"  
MultiSelectionTextFunc="@(new Func<List<string>, string>(GetMultiSelectionText))"  
Label="Felines" AdornmentIcon="@Icons.Material.Filled.Search">  
    @foreach (var feline in felines)  
    {  
        <MudSelectedItem T="string" Value="@feline">@feline</MudSelectedItem>  
    }  
</MudSelect>
```

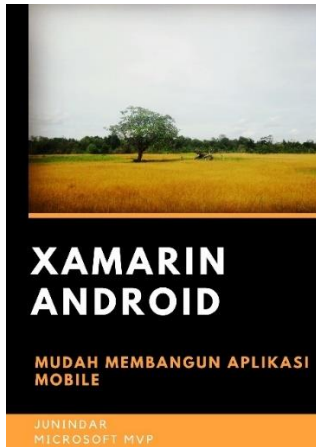


Penutup

Sedangkan untuk memudahkan dalam memahami isi artikel, maka penulis juga menyertakan dengan full source code project latihan ini, dan dapat di download disini

<http://junindar.blogspot.com/2022/10/mudblazor-input-component-pada-blazor.html>

Referensi



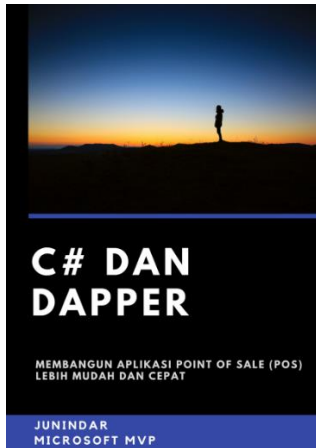
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



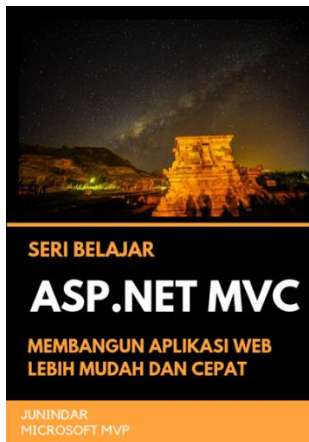
<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>



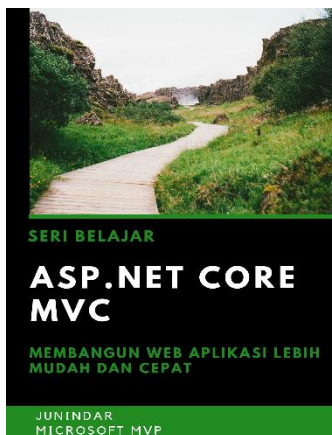
https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ



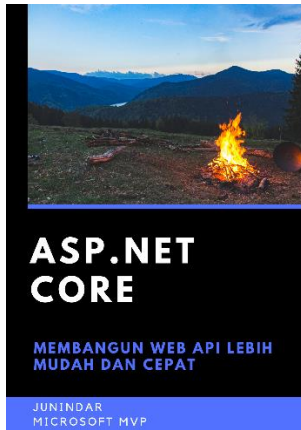
https://play.google.com/store/books/details/Junindar_C_dan_Dapper_Membangun_Aplikasi_POS_Point?id=6TErDwAAQBAJ



https://play.google.com/store/books/details/Junindar_ASP_NET_MVC_Membangun_Aplikasi_Web_Lebih?id=XLlyDwAAQBAJ



https://play.google.com/store/books/details/Junindar_ASP_NET_CORE_MVC?id=xEe5DwAAQBAJ



https://play.google.com/store/books/details/Junindar_ASP_NET_CORE?id=COUWEAAQBAJ



https://play.google.com/store/books/details/Junindar_Microsoft_Blazor_Membangun_Aplikasi_Web_D?id=HKZhEAAAQBAJ

Biografi Penulis.



Junindar Lahir di Tanjung Pinang, 21 Juni 1982. Menyelesaikan Program S1 pada jurusan Teknik Inscreenatika di Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN-Bandung). Junindar mendapatkan Award Microsoft MVP VB pertanggal 1 oktober 2009 hingga saat ini. Senang mengutak-atik computer yang berkaitan dengan bahasa pemrograman. Keahlian, sedikit mengerti beberapa bahasa pemrograman seperti : VB.Net, C#, SharePoint, ASP.NET, VBA. Reporting: Crystal Report dan Report Builder. Database: MS Access, MY SQL dan SQL Server. Simulation / Modeling Packages: Visio Enterprise, Rational Rose dan Power Designer. Dan senang bermain gitar, karena untuk bisa menjadi pemain gitar dan seorang programmer sama-sama membutuhkan seni. Pada saat ini bekerja di salah satu Perusahaan Consulting dan Project Management di Malaysia sebagai Senior Consultant. Memiliki beberapa sertifikasi dari Microsoft yaitu Microsoft Certified Professional Developer (MCPD – SharePoint 2010), MOS (Microsoft Office Specialist) dan MCT (Microsoft Certified Trainer) Mempunyai moto hidup: **“Jauh lebih baik menjadi Orang Bodoh yang giat belajar, dari pada orang Pintar yang tidak pernah mengimplementasikan ilmunya”**.