

# MudBlazor Table Pada Blazor – Part 1

**Junindar, ST, MCPD, MOS, MCT, MVP**

*junindar@gmail.com*

<http://junindar.blogspot.com>

## ***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.*

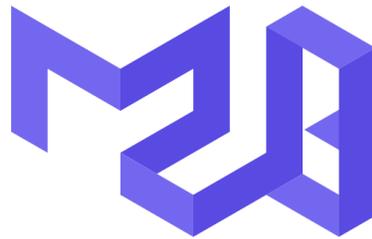
## Abstrak

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

Untuk artikel ini akan dijelaskan bagaimana menggunakan MudBlazor Table pada sebuah project, sebelum melanjutkan artikel ini pastikan telah mengikuti artikel sebelumnya disini <http://junindar.blogspot.com/2021/12/mudblazor-component-pada-blazor-part-1.html> . Pada artikel sebelumnya telah dijelaskan bagaimana menambahkan dan melakukan konfigurasi MudBlazor pada project. Sehingga pada artikel ini pembahasan hanya fokus bagaimana menggunakan MudBlazor Table beserta fungsi-fungsinya.

Untuk memudahkan dalam memahami artikel ini, kita buat terlebih dahulu sebuah project Blazor App. Lalu tambahkan (install) MudBlazor dan lakukan konfigurasi pada project seperti yang telah dijelaskan pada artikel sebelumnya (<http://junindar.blogspot.com/2021/12/mudblazor-component-pada-blazor-part-1.html>).

Sedangkan untuk pengolahan datanya kita gunakan Entity Framework seperti pada atikel berikut (<http://junindar.blogspot.com/2020/02/create-read-update-dan-delete-crud-pada.html>). Untuk selanjutnya ikuti langkah-langkah dibawah ini.

- Buka file index dan hapus seluruh sintaks yang ada. Pada file ini akan kita gunakan untuk menampilkan data pada MudBlazor Table. Ketikkan sintaks dibawah ini untuk menambahkan komponen MudBlazor Table pada halaman Index.

```
<MudTable Items="@Books" Hover="true" Breakpoint="Breakpoint.Sm"
Loading="@_loading" LoadingProgressColor="Color.Info">
  <HeaderContent>
    <MudTh></MudTh>
    <MudTh>Id Buku</MudTh>
    <MudTh>Judul</MudTh>
    <MudTh>Penulis</MudTh>
  </HeaderContent>
  <RowTemplate>
    <MudTd DataLabel="Gambar">
      
    </MudTd>
    <MudTd DataLabel="BookID">@context.BookID</MudTd>
    <MudTd DataLabel="Judul">@context.Judul</MudTd>
    <MudTd DataLabel="Penulis">@context.Penulis</MudTd>
  </RowTemplate>
</MudTable>

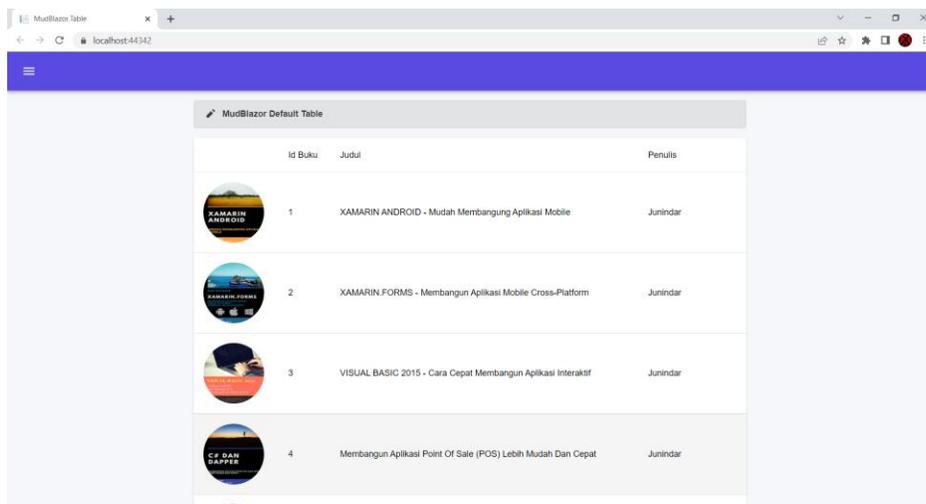
@code {
  private bool _loading;
  [Inject]
  public IBookRepository BookRepository { get; set; }
  private IEnumerable<Book> Books = new List<Book>();

  protected override async Task OnInitializedAsync()
  {
    _loading = true;
    await Task.Delay(1000);
    Books = (await BookRepository.GetAllBooks()).ToList();
    _loading = false;
  }
}
```

Pada MudTable terdapat property “Items” (IEnumerable) yang digunakan untuk menampilkan data pada table. MudTable akan membuat 1 baris untuk 1 item. Disini

untuk “Items” kita gunakan property “Books” yang akan digunakan untuk menampung data dari Database. Sintaks untuk memanggil fungsi pengambilan data dari database terdapat pada method “OnInitializedAsync”.

Pada MudTable, terdapat Node “HeaderContent” dan “RowTemplate”. HeaderContent digunakan sebagai Header dari table, sedangkan untuk RowTemplate digunakan untuk menampilkan item-item pada setiap baris. Untuk item-item tersebut kita gunakan “DataLabel”. Jalankan program untuk melihat hasilnya.

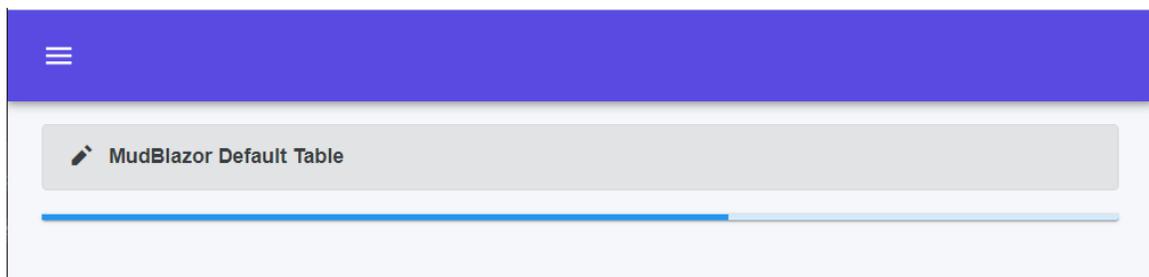


Untuk kolom pertama, data yang kita tampilkan adalah gambar dari buku dengan menggunakan sintaks seperti berikut.

```
<MudTd DataLabel="Gambar">  
    
</MudTd>
```

Perlu diketahui Data pada kolom gambar adalah berupa text yaitu nama dari file gambar tersebut. Sedangkan untuk menampilkan data dari item kita gunakan sintaks “{@context>NamaProperty”.

Pada MudTable terdapat property untuk menampilkan animasi loading (**Loading**) dan pengaturan warna pada loading tersebut (**LoadingProgressColor**).



Pada latihan ini kita gunakan variable “\_loading” yang bertipe boolean (true/false). Dapat dilihat pada method “OnInitializedAsync”, value dari “\_loading” kita set menjadi true terlebih dahulu, sehingga animasi loading akan muncul sebelum data ditampilkan. Lalu disini kita gunakan sintaks “await Task.Delay(1000);” untuk melakukan delay sebelum memanggil fungsi “BookRepository.GetAllBooks()”.Setelah itu baru kita ganti value dari “\_loading” menjadi false, untuk menghilangkan animasi loading pada layar.

Note : Sintaks “await Task.Delay(1000);” digunakan hanya sebagai latihan untuk menampilkan animasi loading sebelum data tampil pada layar.

#### - Paging, Filtering dan Sorting

Selain menampilkan data MudTable komponen mendukung proses paging, filtering dan sorting. Tambahkan sebuah Razor Component untuk membuat proses-proses tersebut, lalu copy seluruh sintaks yang ada pada file Index.razor dan selanjutnya akan kita modifikasi seperti berikut.

Buat sebuah function dengan nama “FilterFunc” seperti dibawah. Return value dari fungsi ini bertipe boolean dan memiliki dua buah parameter objek “Book” dan string (searchString). Terdapat 3 buah kondisi pada fungsi ini, yang pertama jika nilai dari “searchString” adalah kosong/empty maka return valuenya adalah true. Dan jika terdapat nilai pada searchString, maka nilai tersebut akan digunakan untuk mencari data pada property Penulis dan Judul. Jika ditemukan maka bernilai true, dan jika tidak maka bernilai false.

```
private bool FilterFunc(Book book, string searchString)
{
    if (string.IsNullOrEmpty(searchString))
        return true;
    if (book.Penulis.Contains(searchString,
        StringComparison.OrdinalIgnoreCase))
        return true;
    if (book.Judul.Contains(searchString,
        StringComparison.OrdinalIgnoreCase))
        return true;

    return false;
}
private string searchString = "";
private bool FilterTable(Book book) => FilterFunc(book, searchString);
```

Setelah membuat fungsi diatas, kita tambahkan dua buah variable “searchString” dan “FilterTable”. Variable “FilterTable” digunakan untuk menampung nilai dari fungsi yang telah kita buat diatas. Dan nilai ini akan digunakan pada MudTable.

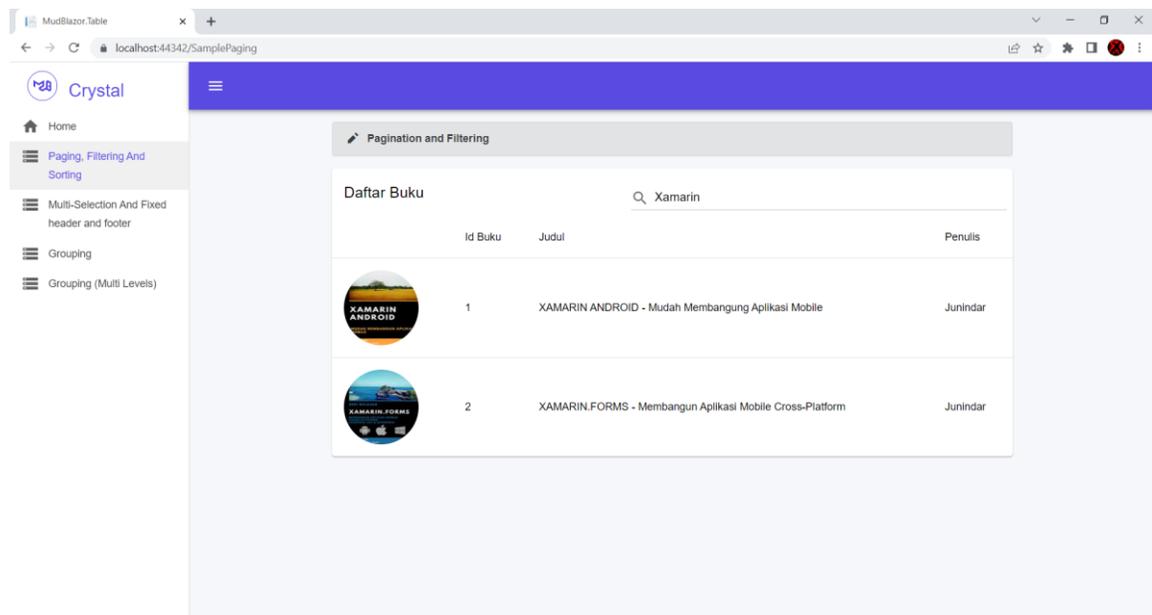
Selanjutnya pada MudTable tambahkan property “Filter”. Property ini menggunakan sebuah function, dimana hasil dari function tersebut akan digunakan untuk menentukan apakah item akan ditampilkan pada table atau tidak. Berikut sintaks untuk menggunakan property filter.

```
“Filter=new Func<Book,bool>(FilterTable)”
```

Masih pada MudTable, kita akan menambahkan Toolbar pada table, dimana pada toolbar ini akan terdapat Textbox (MudTextField) untuk proses pencarian. Dan nilai dari textbox ini akan di *binding* dengan variable “searchString”.

```
<ToolBarContent>  
  <MudText Typo="Typo.h6">Daftar Buku</MudText>  
  <MudSpacer />  
  <MudTextField @bind-Value="searchString" Placeholder="Search"  
    Adornment="Adornment.Start"  
    AdornmentIcon="@Icons.Material.Filled.Search" IconSize="Size.Medium"  
    Class="mt-0"></MudTextField>  
</ToolBarContent>
```

Lalu jalankan program dan lakukan pencarian data untuk melihat hasilnya.



Untuk langkah selanjutnya kita akan menambahkan paging pada table. Untuk menampilkan daftar pilihan page size pada MudTable, kita ketikkan sintaks seperti dibawah.

```
<PagerContent>  
  <MudTablePager />  
</PagerContent>
```

Jalankan program untuk melihat hasilnya, pastikan mendapatkan hasil seperti gambar dibawah ini.

	9	Raga Kayu Jiwa Manusia	Sarah Anais Andrieu
	10	Generasi Phi	Dr. Muhammad Faizal

Default pilihan dari page size adalah “10, 25, 50 dan 100”, dan menampilkan 10 data per page. Lalu bagaimana jika kita ingin mengganti page size dan default pilihannya. Untuk melakukan hal tersebut, kita perlu melakukan beberapa penambahan sintaks pada razor component.

Buat sebuah variable seperti pada sintaks C# dibawah ini.

```
private int[] pageSizeOptions = new int[] { 5, 10, 15 };
```

Variable ini digunakan untuk menampilkan daftar pilihan page size dalam menampilkan data pada table. Sedangkan untuk menampilkan daftar page size diatas pada MudTable, kita ketikkan sintaks seperti dibawah.

```
<PagerContent>  
  <MudTablePager PageSizeOptions="pageSizeOptions" />  
</PagerContent>
```

Pada property “PageSizeOption” kita gunakan variable “pageSizeOptions” yang telah kita buat sebelumnya. Lalu ketikkan sintaks dibawah untuk mengganti default valuenya.

```
private MudTable<Book> table;  
  
protected override Task OnAfterRenderAsync(bool firstRender)  
{  
  table.SetRowsPerPage(5);  
  return base.OnAfterRenderAsync(firstRender);  
}
```

Dan tambahkan sintaks beriku pada MudTable.

```
“@ref="table””
```

Jalankan program dan pastikan perubahan yang telah kita buat diatas berfungsi dengan baik.

	3	VISUAL BASIC 2015 - Cara Cepat Membangun Aplikasi Interaktif	Junindar
	4	Membangun Aplikasi Point Of Sale (POS) Lebih Mudah Dan Cepat	Junindar
	5	Visual Studio LightSwitch Learning By Doing	Junindar

Rows per page: 5 10 15 1-5 of 11

Setelah berhasil dengan latihan diatas, latihan selanjutnya adalah dengan menambahkan fungsi sorting pada kolom. Untuk melakukan hal tersebut kita perlu mengganti sintaks pada “**HeaderContent**”, seperti dibawah ini.

```
<HeaderContent>
  <MudTh></MudTh>
  <MudTh>
    <MudTableSortLabel SortBy="new Func<Book, object>(x=>x.BookID)">Id
    Buku</MudTableSortLabel></MudTh>
    <MudTh>
      <MudTableSortLabel SortBy="new Func<Book,
      object>(x=>x.Judul)">Judul</MudTableSortLabel>
    </MudTh>
    <MudTh>
      <MudTableSortLabel SortBy="new Func<Book,
      object>(x=>x.Penulis)">Penulis</MudTableSortLabel>
    </MudTh>
  </HeaderContent>
```

dan pastikan mendapatkan hasil seperti dibawah ini.

Pagination and Filtering

Daftar Buku 🔍 Search

	Id Buku ↑	Judul	Penulis
	1	XAMARIN ANDROID - Mudah Membangun Aplikasi Mobile	Junindar
	2	XAMARIN.FORMS - Membangun Aplikasi Mobile Cross-Platform	Junindar

- Multi Selection dan Fixed header & footer

Setelah selesai dengan fungsi Paging, Filtering dan Sorting, kita akan lanjutkan dengan membuat Multi Selection dan Fixed header & footer.

Tambahkan sebuah Razor Component lalu copy sintaks dari latihan sebelumnya (Paging, Filtering dan Sorting). Untuk menambahkan fungsi MultiSelection pada MudTable, kita hanya perlu menambah property “**MultiSelection="true"**”. Setelah menambahkan sintaks tersebut maka pada table akan terdapat checkbox pada setiap item dan header. Sedangkan untuk menambahkan checkbox pada footer kita perlu menambahkan sintaks seperti dibawah ini.

```
<FooterContent>  
  <MudTd colspan="5">Select All</MudTd>  
</FooterContent>
```

Sehingga akan mendapatkan hasil seperti dibawah ini.

The screenshot shows a web application interface with a table titled "Daftar Buku". At the top right of the table area is a search bar with a magnifying glass icon and the text "Search". The table has five columns: a checkbox column, "Id Buku", "Judul", and "Penulis". There are five rows of book data. The first two rows have their checkboxes checked, while the last three are unchecked. At the bottom of the table, there is a "Select All" checkbox.

<input type="checkbox"/>	Id Buku	Judul	Penulis	
<input checked="" type="checkbox"/>	1	XAMARIN ANDROID - Mudah Membangun Aplikasi Mobile	Junindar	
<input checked="" type="checkbox"/>	2	XAMARIN FORMS - Membangun Aplikasi Mobile Cross-Platform	Junindar	
<input type="checkbox"/>	3	VISUAL BASIC 2015 - Cara Cepat Membangun Aplikasi Interaktif	Junindar	
<input type="checkbox"/>	4	Membangun Aplikasi Point Of Sale (POS) Lebih Mudah Dan Cepat	Junindar	
<input type="checkbox"/>	5	Visual Studio LightSwitch Learning By Doing	Junindar	
<input type="checkbox"/>	Select All			

Setelah berhasil membuat Multi Selection, lalu bagaimana untuk mendapatkan value dari item-item yang dipilih. Untuk membuat fungsi tersebut ikuti langkah-langkah dibawah ini.

Tambahkan sebuah variable “selectedItem1” seperti dibawah ini.

```
private HashSet<Book> selectedItems1 = new HashSet<Book>();
```

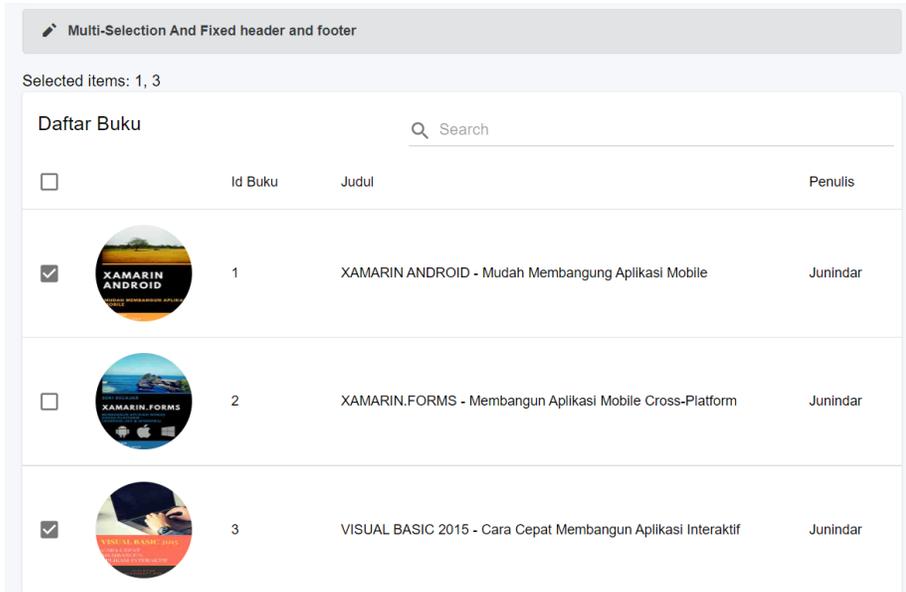
Variable ini digunakan untuk menampung item-item yang dipilih. Selanjutnya pada MudTable tambahkan sintaks berikut ”**@bind-SelectedItems="selectedItems1"**”.

Disini kita melakukan *binding* property SelectedItems ke variable “selectedItems1”.

Sedangkan untuk menampilkan “ID” dari item, ketikkan sintaks dibawah ini.

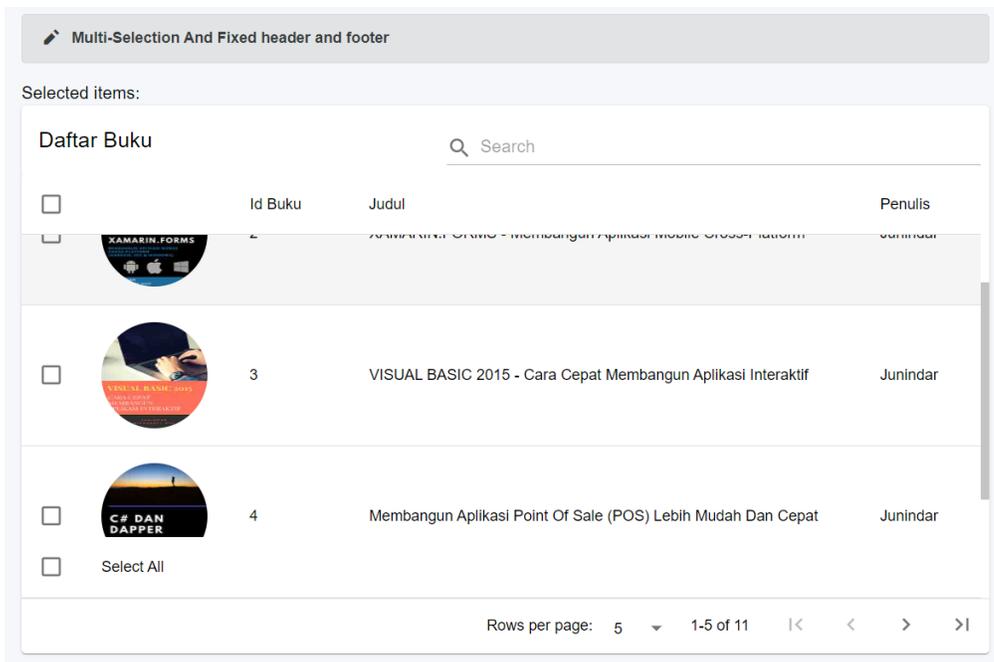
```
<MudText Inline="true">Selected items: @(@(selectedItems1==null ? "" : string.Join(", ", selectedItems1.OrderBy(x=>x.BookID).Select(x=>x.BookID)))</MudText>
```

Lalu jalankan program untuk mendapatkan hasil seperti dibawah.



Terakhir untuk membuat Fixed header & footer, kita hanya perlu menambahkan sintaks berikut pada MudTable.

```
FixedHeader="true" FixedFooter="true" Height="400px"
```

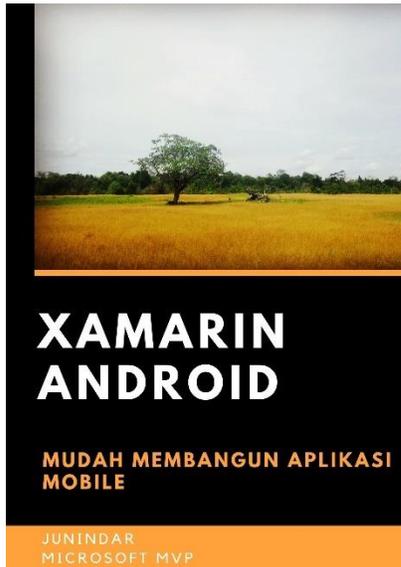


## **Penutup**

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/2022/07/mudblazor-table-pada-blazor-part-1.html>

## Referensi



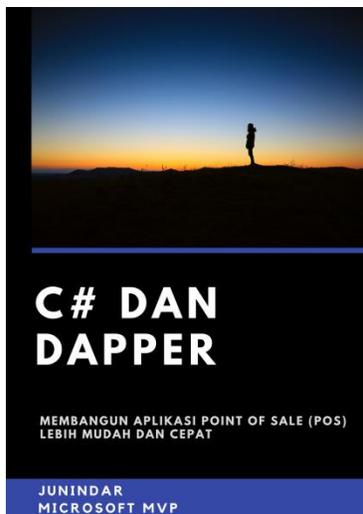
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



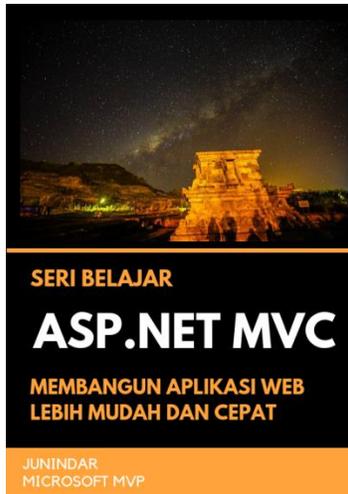
<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>



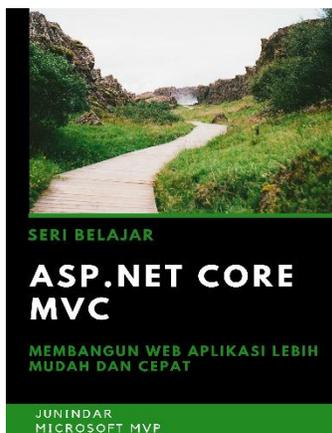
[https://play.google.com/store/books/details/Junindar\\_Xamarin\\_Forms?id=6Wg-DwAAQBAJ](https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ)



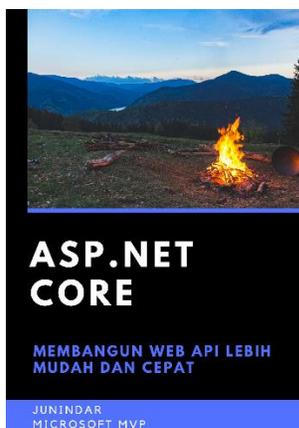
[https://play.google.com/store/books/details/Junindar\\_C dan Dapper Membangun Aplikasi POS Point?id=6TErDwAAQBAJ](https://play.google.com/store/books/details/Junindar_C_dan_Dapper_Membangun_Aplikasi_POS_Point?id=6TErDwAAQBAJ)



[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_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_MVC?id=xEe5DwAAQBAJ)



[https://play.google.com/store/books/details/Junindar\\_ASP\\_NET\\_CORE?id=COUWEAAQBAJ](https://play.google.com/store/books/details/Junindar_ASP_NET_CORE?id=COUWEAAQBAJ)



<https://play.google.com/store/books/details?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”**.