

# MudBlazor Table Pada Blazor – Part 2

**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.*

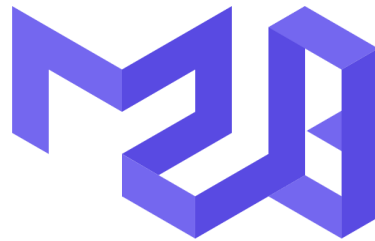
## Abstrak

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

Artikel ini melanjutkan dari artikel sebelumnya, yang membahas fungsi-fungsi yang ada pada MudBlazor Table (MudTable). Oleh karena itu 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> dan <http://junindar.blogspot.com/2022/07/mudblazor-table-pada-blazor-part-1.html> ).

Pada artikel sebelumnya telah dibahas bagaimana membuat fungsi Filtering, paging dan multi selection pada MudTable. Sedangkan untuk artikel ini kita akan membahas beberapa fitur lagi yang terdapat pada MudBlazor seperti Grouping dan kustomisasi header-footer. Perlu diketahui, pada artikel ini kita masih akan menggunakan blazor project yang telah kita buat pada artikel sebelumnya.

## - Grouping

Pada MudTable kita dapat menampilkan data secara group (grouping) dengan header dan footer pada setiap group. Dan juga menampilkan summary untuk masing-masing group pada table. Ikuti langkah-langkah berikut untuk membuat grouping pada MudTable.

Tambahkan sebuah Razor Component pada project lalu copy-kan sintaks dari “samplepaging.razor“. lalu ikuti langkah-langkah dibawah ini.

```
private bool _multiSelect = true;

private TableGroupDefinition<Book> _groupDefinition = new TableGroupDefinition<Book>
{
    GroupName = "Category",
    Expandable = true,
    IsInitiallyExpanded = true,
    Selector = (e) => e.Category>NamaCategory
};
```

Buat dua buah variable, yang pertama bertipe boolean dengan nama “\_multiSelect“ dan yang kedua “\_groupDefinition“ yang menggunakan class “TableGroupDefinition“. Pada “\_multiSelect“ akan digunakan untuk mengaktif dan non aktifkan fitur MultiSection. Lalu untuk “\_groupDefinition“ digunakan untuk mengatur dari proses grouping. Seperti “GroupName“ yang digunakan untuk menampilkan text pada group header. “Expandable“ digunakan untuk menampilkan *expand/collapse* control pada group. Lalu ada “IsInitiallyExpanded“ digunakan untuk mengatur *expand/collapse* control pada saat inisialisasi, dimana jika kita set menjadi true maka group akan di-*expand*. Dan terakhir adalah “Selector“ yang merupakan sebuah “func“ untuk memilih kolom “.GroupBy“ dengan menggunakan LINQ.

Lalu kita tambahkan property pada MudTable, sehingga seperti dibawah ini.

```
<MudTable Items="@Books" Hover="true" Loading="@_loading" @ref="table"
    Height="600px" GroupBy="@_groupDefinition" MultiSelection="_multiSelect"
    GroupHeaderStyle="background-color:var(--mud-palette-background-grey)"
    GroupFooterClass="mb-4"
    LoadingProgressColor="Color.Info">
```

“GroupBy” digunakan untuk menentukan parameter pengelompokan data (Grouping), dimana yang telah kita buat sebelumnya (\_groupDefinition). “MultiSelection” akan menggunakan value dari variable “\_multiSelect”. Jika true maka checkbox selection akan ditampilkan. “GroupHeaderStyle” dan “GroupFooterClass” untuk mendefinisikan

custom style yang akan digunakan pada GroupHeader (**GroupHeaderStyle**) maupun Footer (**GroupFooterClass**).

Selanjutnya kita tambahkan template untuk header pada group seperti dibawah.

```
<GroupHeaderTemplate>  
    <MudTh Class="mud-table-cell-custom-group"  
        colspan="5">@($){context.GroupName}: {context.Key}</MudTh>  
</GroupHeaderTemplate>
```

“context.GroupName” nama group yang telah kita set pada “\_groupDefinition.GroupName”, sedangkan “context.Key” adalah nilai dari GroupBy pada “\_groupDefinition.Selector”. Setelah selesai dengan header kita lanjutkan dengan template untuk footer pada group.

```
<GroupFooterTemplate>  
    <MudTh Class="mud-table-cell-custom-group mud-table-cell-custom-group-  
        footer" colspan="5">Number of Book: @context.Items.Count()</MudTh>  
</GroupFooterTemplate>
```

Pada footer group akan tampilkan jumlah data untuk masing-masing group dengan menggunakan sintaks“context.Items.Count()”. Pada template header dan footer diatas terdapat sebuah property class untuk style. Oleh karena itu kita perlu membuat css seperti dibawah.

```
<style>  
    .mud-table-cell-custom-group {  
        font-weight: 500;  
    }  
  
    .mud-table-cell-custom-group-footer {  
        padding-bottom: 50px;  
        text-align: right;  
    }  
</style>
```

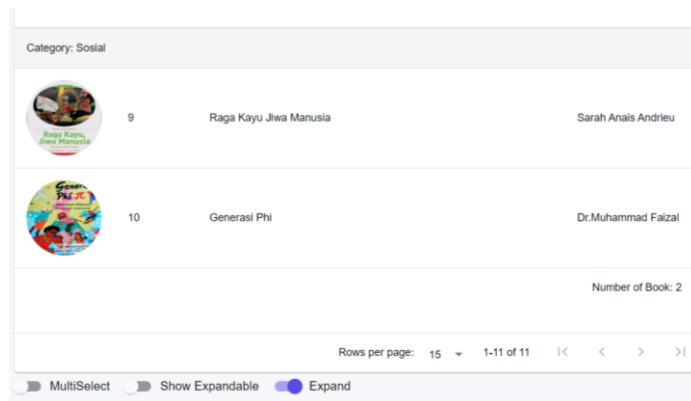
Jalankan program dan pastikan mendapatkan hasil seperti dibawah.

<input type="checkbox"/>	ID	Title	Author
<input type="checkbox"/>	12	Microsoft Blazor: Membangun Aplikasi Web Dengan Mudah dan Cepat	Junindar
Number of Book: 9			
▼ <input type="checkbox"/>	Category: Sosial		
<input type="checkbox"/>	9	Raga Kayu Jiwa Manusia	Sarah Anals Andrieu
<input type="checkbox"/>	10	Generasi Phi	Dr. Muhammad Faizal
Number of Book: 2			

Tambahkan 3 buah MudSwitch dengan mengetikkan sintaks seperti dibawah.

```
<MudSwitch @bind-Checked="_multiSelect"
           Color="Color.Primary">MultiSelect</MudSwitch>
<MudSwitch @bind-Checked="_groupDefinition.Expandable"
           Color="Color.Primary">Show Expandable</MudSwitch>
<MudSwitch @bind-Checked="_groupDefinition.IsInitiallyExpanded"
           Color="Color.Primary">Expand</MudSwitch>
```

disini kita dapat mengatur beberapa fungsi secara runtime, seperti apakah kita ingin menampilkan selection atau expand/collapse control pada table.



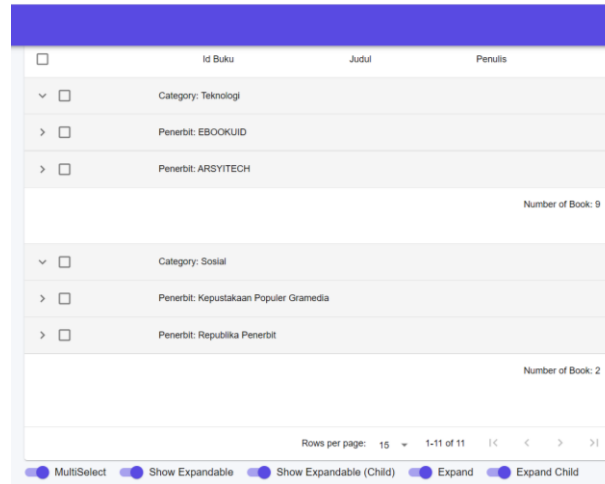
- Grouping Multi level

Pada latihan sebelumnya kita telah membuat group berdasarkan kolom Category. Dimana kita hanya menggunakan 1 kolom untuk mengelompokkan data. Pada latihan ini kita akan mengelompokkan data dengan menggunakan 2 kolom. Yang pertama adalah Category dan dilanjutkan dengan kolom Penerbit.

Buat sebuah Razor Component pada project, lalu copy seluruh sintaks pada latihan sebelumnya (grouping) kedalam razor component yang baru dibuat ini. Ganti sintaks pada “\_groupDefinition“ seperti dibawah.

```
private TableGroupDefinition<Book> _groupDefinition = new TableGroupDefinition<Book>
{
    GroupName = "Category",
    Expandable = true,
    IsInitiallyExpanded = true,
    Selector = (e) => e.Category>NamaCategory,
    InnerGroup = new TableGroupDefinition<Book>()
    {
        GroupName = "Penerbit",
        Expandable = true,
        IsInitiallyExpanded = true,
        Selector = (e) => e.Penerbit
    }
};
```

Dapat dilihat pada sintaks diatas terdapat property “InnerGroup” yang digunakan untuk mengatur pengelompokkan data untuk kolom kedua. Jalankan program untuk melihat hasilnya, pastikan seperti pada gambar dibawah.



#### - Custom Header – Footer

Pada MudTable kita dapat melakukan kustomisasi pada header maupun footer. Sebagai contoh kita akan menambahkan baris pada header dan melakukan merge untuk kolom-kolomnya. Sebelumnya kita ganti property-property pada MudTable seperti dibawah ini.

```
<MudTable Items="@Books" Hover="true" Loading="@_loading" @ref="table"
  Filter="new Func<Book,bool>(FilterTable)"
  @bind-SelectedItems="selectedItems" MultiSelection="_multiSelect"
  Striped="true" Bordered="true"
  CustomHeader="true" CustomFooter="true" HeaderComponent="table-head-bordered"
  FooterClass="table-foot-bordered"
  LoadingProgressColor="Color.Info">
```

Untuk melakukan kustomisasi pada header dan footer, kita tambahkan property CustomHeader dan CustomFooter menjadi true. Sehingga kita bisa menambah baris pada header maupun footer dengan menggunakan “MudTHedRow“ dan “MudTFootRow“. Lalu pada node “<HeaderContent>“ ganti sintaksnya menjadi seperti dibawah ini.

```
<HeaderContent>
  <MudTHeadRow IgnoreCheckbox="true" Class="header-centered">
    <MudTh colspan="5">Books Info</MudTh>
  </MudTHeadRow>
  <MudTHeadRow IgnoreCheckbox="true" Class="header-centered">
    <MudTh colspan="2">Image</MudTh>
    <MudTh colspan="3">Detail Data</MudTh>
  </MudTHeadRow>
  <MudTHeadRow IsCheckable="true">
    <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>
  </MudTHeadRow>
</HeaderContent>
```

Didalam HeaderContent, kita tambahkan 3 buah baris dengan menggunakan “MudTHeadRow“. Pada baris pertama memiliki satu kolom, sehingga kita perlu merge 5 kolom (colspan="5"). Untuk baris kedua, terdapat 2 kolom, kolom yang pertama “Image“ (merge 2 kolom) dan kolom kedua “DetailData“ (merge 2 kolom). Sedangkan baris terakhir adalah header untuk table yang akan ditampilkan. Terdapat 5 kolom pada baris terakhir ini yaitu selection, Gambar, Id Buku, Judul dan Penulis.






Pada baris pertama dan kedua terdapat property “IgnoreCheckbox="true“ yang berfungsi untuk multiselection aktif, jika value nya sama dengan “true“ maka tidak akan membuat kolom tambahan untuk checkbox. Sedangkan pada baris ketiga kita gunakan “IsCheckable="true“ untuk menampilkan checkbox (select/unselect all) pada header.

Lalu kita lanjutkan dengan menambahkan node “FooterContent“ seperti dibawah.

```
<FooterContent>
  <MudTFooterRow Class="bold-text">
    <MudTd></MudTd>
    <MudTd>Id Buku</MudTd>
    <MudTd>Judul</MudTd>
    <MudTd>Penulis</MudTd>
  </MudTFooterRow>
  <MudTFooterRow IsCheckable="true">
    <MudTd colspan="5">Selected: @selectedItems.Count</MudTd>
  </MudTFooterRow>
</FooterContent>
```

Konsepnya hampir sama seperti header, kita dapat menambahkan baris pada footer. Pada sintaks diatas kita tambahkan dua buah baris, dimana pada baris pertama sama seperti pada header di baris ketiga. Sedangkan untuk baris kedua pada footer terdapat sebuah kolom (merge 5 kolom), lalu kita gunakan “IsCheckable="true“ untuk menampilkan checkbox (select/unselect all). Pada kolom tersebut kita tampilkan

jumlah data yang dipilih pada table (selection). Jalankan program dan pastikan mendapatkan hasil seperti pada gambar dibawah ini.

Books Info				
Image	Detail Data			
<input type="checkbox"/>		Id Buku	Judul	Penulis
<input type="checkbox"/>		1	XAMARIN ANDROID - Mudah Membangun Aplikasi Mobile	Junindar
<input type="checkbox"/>		2	XAMARIN FORMS - Membangun Aplikasi Mobile Cross-Platform	Junindar
<input checked="" type="checkbox"/>		3	VISUAL BASIC 2015 - Cara Cepat Membangun Aplikasi Interaktif	Junindar
<input checked="" 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
		Id Buku	Judul	Penulis
<input type="checkbox"/>	Selected: 2			

Rows per page: 5 1-5 of 11 < > >>

- ChildRowContent

Pada MudTable kita dapat menambah child table, untuk membuat child table tersebut kita gunakan “<ChildRowContent>“. Dengan “<ChildRowContent>“ memungkinkan kita untuk menambahkan kontrol yang lebih banyak pada layout.

Pada latihan ini, kita akan menggunakan skenario menampilkan Buku (Book) yang berelasi dengan Kategori. Buat sebuah Razor Component pada project lalu ikuti langkah-langkah dibawah ini.

Tambahkan dua buah class seperti dibawah. Dimana class-class ini digunakan untuk menampilkan data pada MudTable.



```
public class CategoryVM
{
    public bool ShowDetails { get; set; }
    public int CategoryID { get; set; }
    public string NamaCategory { get; set; }
    public IList<BookVM> Books { get; set; }= new List<BookVM>();
}

public class BookVM
{
    public int BookID { get; set; }
    public string Judul { get; set; }
    public string Penulis { get; set; }
    public string Gambar { get; set; }
}
```

Dapat kita lihat sintaks diatas, pada class “CategoryVM” terdapat sebuah property “IList<BookVM> Books” dimana pada satu Category bisa terdapat lebih dari satu Buku (Book). Dan selanjutnya kita buat sebuah variable “\_loading” yang digunakan sebagai indikator animasi loading pada MudTable. Dan diikuti dengan membuat field “Categories” dengan menggunakan List<CategoryVM>.

```
private bool _loading;
private List<CategoryVM> Categories = new List<CategoryVM>();
```

Lalu deklarasikan interface dari ”IBookRepository” dan “ICategoryRepository”, detail sintaks dari kedua interface tersebut dapat dilihat pada project lampiran.

```
[Inject]
public IBookRepository BookRepository { get; set; }

[Inject]
public ICategoryRepository CategoryRepository { get; set; }

private async Task FillData()
{
    var categories = (await CategoryRepository.GetAll()).ToList();
    var books = (await BookRepository.GetAllBooks()).ToList();
    foreach (var cat in categories)
    {
        var catVM = new CategoryVM()
        {
            CategoryID=cat.CategoryID,NamaCategory=cat.NamaCategory,
            ShowDetails=false
        };

        foreach (var book in books.Where(c=>c.CategoryID==cat.CategoryID))
        {
            var bookVM = new BookVM()
            {
                BookID= book.BookID,
                Judul = book.Judul,
                Penulis = book.Penulis,
                Gambar = book.Gambar
            };
            catVM.Books.Add(bookVM);
        }

        Categories.Add(catVM);
    }
}
```

Tambahkan sebuah method “FillData“ seperti diatas. Pada sintaks diatas kita panggil method “GetAll“ (CategoryRepository) dan “GetAllBooks“ (BookRepository). Lalu kita *populate* data-data tersebut kedalam variable “Categories“ yang telah kita buat sebelumnya.

```
private void ShowBtnPress(int catid)
{
    CategoryVM tmpCat = Categories.First(f => f.CategoryID == catid);
    tmpCat.ShowDetails = !tmpCat.ShowDetails;
}
```

Selanjutnya buat sebuah method “ShowBtnPress” seperti diatas. Method ini digunakan untuk mendapatkan nilai “ShowDetails” dari Category yang dipilih. Jika true maka table child akan ditampilkan. Dan dilanjutkan dengan membuat override method (OnInitializedAsync) seperti dibawah.

```
protected override async Task OnInitializedAsync()
{
    _loading = true;
    await Task.Delay(1000);
    await FillData();
    _loading = false;
}
```

Setelah selesai dengan sintaks pada C#, kita lanjutkan dengan membuat sintaks pada MudTable. Berikut adalah property yang digunakan pada MudTable dilatihan ini. Untuk Items kita gunakan “Categories“ yang telah kita buat sebelumnya.

```
<MudTable Items="@Categories" Hover="true" Breakpoint="Breakpoint.Sm"
    Loading="@_loading" LoadingProgressColor="Color.Info">
```

Untuk header kita tambahkan 3 buah kolom seperti dibawah ini.

```
<HeaderContent>
    <MudTh></MudTh>
    <MudTh>Id Category</MudTh>
    <MudTh>Category Name</MudTh>
</HeaderContent>
```

Untuk RowTemplate, pada kolom pertama kita buah sebuah button (MudButton), dimana untuk event “OnClick” kita panggil method “ShowBtnPress” yang sebelumnya telah dibuat.

```
<RowTemplate>
    <MudTd>
        <MudButton Variant="Variant.Outlined" Size="Size.Small"
            OnClick="@(() => ShowBtnPress(context.CategoryID))">
            @((context.ShowDetails == true) ? "Hide" : "Show") Book Details
        </MudButton>
    </MudTd>
    <MudTd DataLabel="Nr">@context.CategoryID</MudTd>
    <MudTd DataLabel="Name">@context>NamaCategory</MudTd>
</RowTemplate>
```

Dan yang terakhir adalah dengan membuat ChildRowContent, dimana jika “ShowDetails” sama dengan true, maka akan menampilkan MudTable (Child table) yang berisikan data-data buku (Book) berdasarkan kategori yang dipilih atau diklik.



```
<ChildRowContent>
@if (context.ShowDetails)
{
    <MudTr>
        <td colspan="3">
            <MudCard Elevation="0">
                <MudCardHeader>
                    <CardHeaderContent>
                        <MudText Typo="Typo.body1">Book Details for
                            <strong>@context>NamaCategory</strong></MudText>
                    </CardHeaderContent>
                </MudCardHeader>
                <MudCardContent Class="pa-0">
                    <MudTable Items="@context.Books" Context="BookContext"
                        Hover="true" Breakpoint="Breakpoint.Sm" Elevation="0">
                        <HeaderContent>
                            <MudTh></MudTh>
                            <MudTh>Judul</MudTh>
                            <MudTh>Penulis</MudTh>
                        </HeaderContent>
                        <RowTemplate>
                            <MudTd DataLabel="Gambar">
                                
                            </MudTd>
                            <MudTd DataLabel="Address Line
                                1">@BookContext.Judul</MudTd>
                            <MudTd DataLabel="Address Line
                                2">@BookContext.Penulis</MudTd>
                        </RowTemplate>
                    </MudTable>
                </MudCardContent>
            </MudCard>
        </td>
    </MudTr>
}
</ChildRowContent>
```

Pada table ini data yang digunakan pada property Items adalah “context.Books“, dimana “Books“ adalah salah satu property pada class CategoryVM. Jalankan program dan pastikan mendapatkan hasil seperti dibawah.

Table With related data

	<b>Id</b>	<b>Category Name</b>
<a href="#">SHOW BOOK DETAILS</a>	1	Agama
<a href="#">SHOW BOOK DETAILS</a>	2	Bahasa
<a href="#">SHOW BOOK DETAILS</a>	3	Teknologi
<a href="#">HIDE BOOK DETAILS</a>	4	Sosial

Book Details for **Sosial**

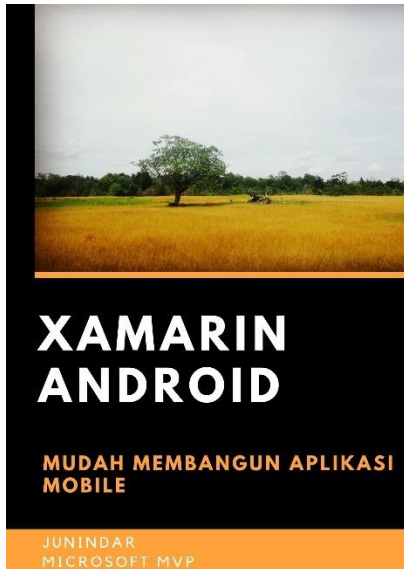
	<b>Judul</b>	<b>Penulis</b>
	Raga Kayu Jiwa Manusia	Sarah Anais Andrieu
	Generasi Phi	Dr. Muhammad Faizal

## **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-2.html>

## Referensi



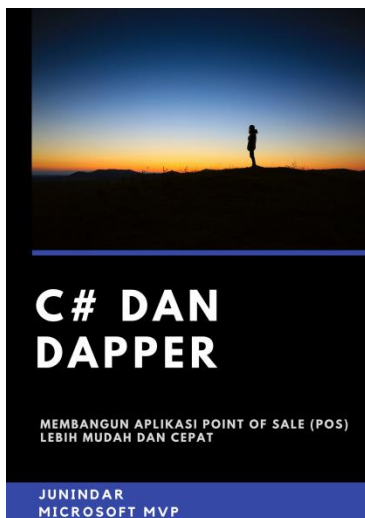
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



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



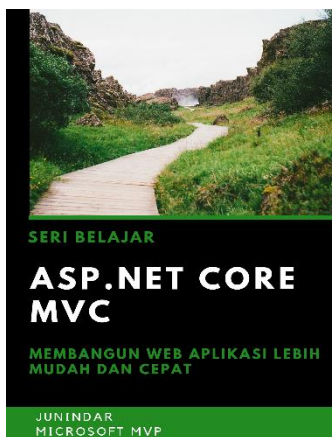
[https://play.google.com/store/books/details/Junindar\\_Xamarin\\_Forms?id=6Wg-DwAAQBAJ](https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ)



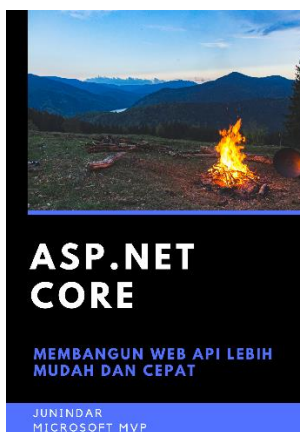
[https://play.google.com/store/books/details/Junindar\\_C dan Dapper Membangun Aplikasi POS Point?id=6TErDwAAQBAJ](https://play.google.com/store/books/details/Junindar_C_dan_Dapper_Membangun_Aplikasi_POS_Point?id=6TErDwAAQBAJ)



[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)



## 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”**.