

# MudBlazor Dialog dan Snackbar Pada Blazor

## ***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>

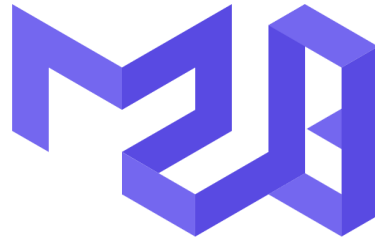
## Abstrak

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

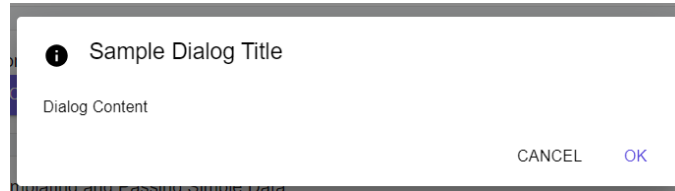
MudBlazor adalah sebuah material design component framework yang dibangun khusus untuk Blazor (<https://mudblazor.com/> ). Terdapat banyak komponen pada MudBlazor seperti chart, grid dan lain-lain untuk membantu dalam membangun aplikasi web dengan menggunakan blazor. Seluruh komponen pada MudBlazor dibangun 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 melanjutkan dari artikel sebelumnya, yang membahas komponen-komponen yang terdapat pada MudBlazor. Komponen-komponen tersebut sering digunakan dalam sebuah aplikasi web. Beberapa komponen yang akan dijelaskan pada artikel ini adalah Dialog dan Snackbar. 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. <https://junindar.blogspot.com/2022/12/mudblazor-input-component-picker-pada.html> dan <https://junindar.blogspot.com/2023/01/mudblazor-chart-component-pada-blazor.html> .

## Dialog



Dialog adalah jendela kecil yang meminta pengguna untuk membuat keputusan atau memasukkan informasi tambahan. Dialog biasanya tidak memenuhi layar dan digunakan untuk mengharuskan pengguna untuk melakukan tindakan sebelum bisa melanjutkan. Pada MudBlazor untuk menggunakan dialog pada aplikasi, kita dapat menggunakan komponen MudDialog. Sebelumnya kita harus melakukan beberapa langkah sebelum menggunakan MudDialog. Tambahkan “<MudDialogProvider />” pada MainLayout.razor (dibawah MudLayout element). Selanjutnya buka \_Imports.razor dan ketikkan sintak seperti berikut “@inject IDialogService \_dialogService”.

Setelah selesai dengan langkah-langkah diatas, kita sudah bisa menggunakan komponen MudDialog. Berikut akan dijelaskan beberapa fitur pada MudDialog.

### - Basic

Pertama-tama kita buat sebuah Razor Component, pada folder “Shared” dengan nama “SampleDialog.razor”. Lalu ketikkan sintaks dibawah ini.

```
<MudDialog>
  <DialogContent>
    Dialog Content
  </DialogContent>
  <DialogActions>
    <MudButton OnClick="Cancel">Cancel</MudButton>
    <MudButton Color="Color.Primary" OnClick="Submit">Ok</MudButton>
  </DialogActions>
</MudDialog>
@code {
  [CascadingParameter] MudDialogInstance MudDialog { get; set; }

  void Submit() => MudDialog.Close(DialogResult.Ok(true));
  void Cancel() => MudDialog.Cancel();
}
```

Lalu untuk memanggil MudDialog atau Razor Component yang telah kita buat diatas, ketikkan sintaks dibawah ini.

```
<MudButton @onclick="OpenDialog" Variant="Variant.Filled" Color="Color.Primary">
    Open Simple Dialog
</MudButton>
@code {
    private void OpenDialog()
    {
        var options = new DialogOptions { CloseOnEscapeKey = true };
        _dialogService.Show<SampleDialog>("Simple Dialog", options);
    }
}
```

#### - Configuration

Terdapat beberapa cara untuk mengganti setting pada MudDialog. Yang pertama kita lakukan secara global, dimana kita hanya cukup mengganti pada “MudDialogProvider“, seperti pada contoh dibawah.

```
<MudDialogProvider
FullWidth="true"
MaxWidth="MaxWidth.ExtraSmall"
CloseButton="true"
DisableBackdropClick="true"
NoHeader="true"
Position="DialogPosition.Center"
CloseOnEscapeKey="true"/>
```

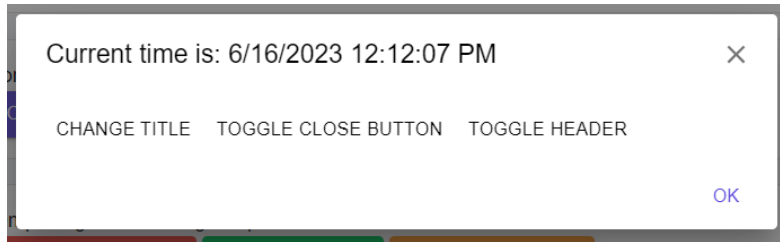
Dengan cara ini seluruh dialog akan memiliki setting yang sama. Selanjutnya cara yang kedua dengan mengganti setting per-dialog. Dengan menggunakan DialogOptions, kita dapat mengganti fitur pada MudDialog, seperti FullWidth, NoHeader, Position dan lain-lain.

```
DialogOptions maxWidth = new() { MaxWidth = MaxWidth.Medium, FullWidth = true };
DialogOptions closeButton = new DialogOptions() { CloseButton = true };
DialogOptions noHeader = new DialogOptions() { NoHeader = true };
DialogOptions disableBackdropClick = new DialogOptions() { DisableBackdropClick = true };
DialogOptions fullScreen = new DialogOptions() { FullScreen = true, CloseButton = true };
DialogOptions topCenter = new DialogOptions() { Position = DialogPosition.TopCenter };

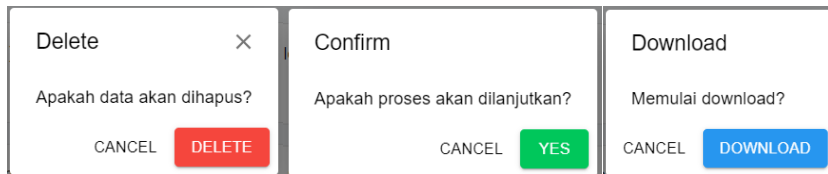
private void OpenDialogConfiguration(DialogOptions options)
{
    _dialogService.Show<SampleDialog>("Custom Options Dialog", options);
}
```

Kita dapat juga mengganti setting pada option dari dialog itu sendiri. Untuk detail dapat dilihat pada project lampiran.

```
void ChangeTitle()
{
    MudDialog.SetTitle("Current time is: " + DateTime.Now);
}
void ToggleCloseButton()
{
    MudDialog.Options.CloseButton = !(MudDialog.Options.CloseButton ?? false);
    MudDialog.SetOptions(MudDialog.Options);
}
void ToggleHeader()
{
    MudDialog.Options.NoHeader = !(MudDialog.Options.NoHeader ?? false);
    MudDialog.SetOptions(MudDialog.Options);
}
```



- Template & Passing Data



Kita dapat menggunakan Paramater pada MudDialog, sehingga hanya cukup dengan satu dialog tetapi dapat digunakan untuk berbagai tujuan.

```
<MudDialog>
  <DialogContent>
    <MudText>@ContentText</MudText>
  </DialogContent>
  <DialogActions>
    <MudButton OnClick="Cancel">Cancel</MudButton>
    <MudButton Color="@Color" Variant="Variant.Filled"
      OnClick="Submit">@ButtonText</MudButton>
  </DialogActions>
</MudDialog>
@code {
  [CascadingParameter] MudDialogInstance MudDialog { get; set; }
  [Parameter] public string ContentText { get; set; }
  [Parameter] public string ButtonText { get; set; }
  [Parameter] public Color Color { get; set; }
  void Submit() => MudDialog.Close(DialogResult.Ok(true));
  void Cancel() => MudDialog.Cancel();
}
```

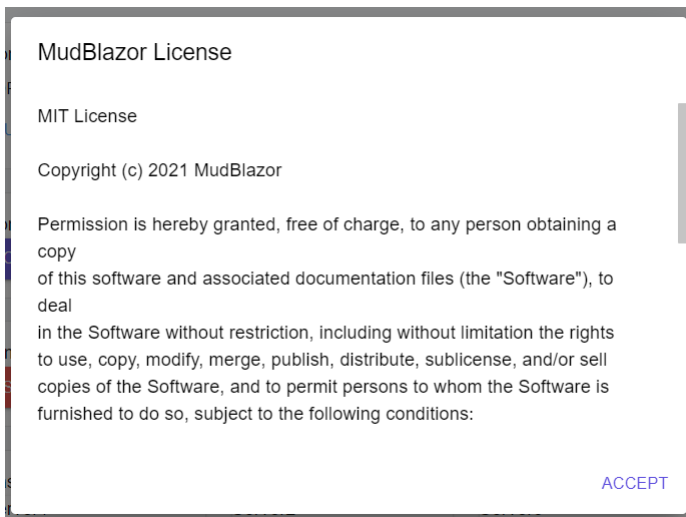
Dengan membuat dialog seperti diatas, beberapa data akan menjadi dynamic, seperti DialogContent, Color button dan Button Text.

```
private void DialogParam1()
{
  var parameters = new DialogParameters();
  parameters.Add("ContentText", "Apakah data akan dihapus?");
  parameters.Add("ButtonText", "Delete");
  parameters.Add("Color", Color.Error);
  var options = new DialogOptions()
  { CloseButton = true, MaxWidth = MaxWidth.ExtraSmall };
  _dialogService.Show<SampleDialogParameter>("Delete", parameters, options);
}
```

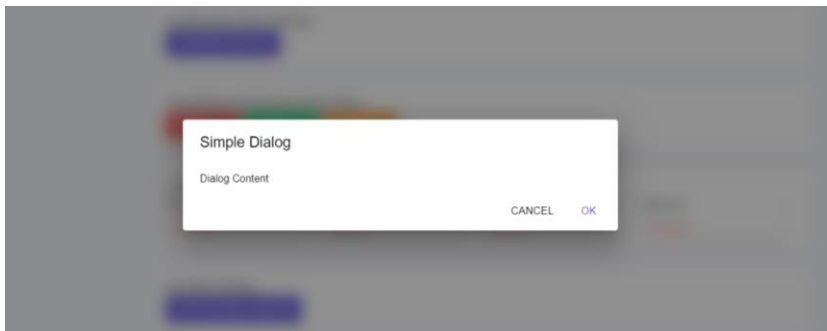
- Scrollable Dialog

Content pada MudDialog dan diatur nilai dari Height dan Width. Sehingga jika data yang ditampilkan besar dan melebihi dari nilai Height dan Width, maka pada content akan terdapat vertical/horizontal scroll.

```
<MudDialog DisableSidePadding="true">  
  <DialogContent>  
    <MudContainer Style="max-height: 300px; overflow-y: scroll">  
      @if (Loading)  
      {  
        <MudProgressCircular Indeterminate="true"></MudProgressCircular>  
      }  
      else  
      {  
        <MudText Style="white-space: pre-wrap;">@LicenseText</MudText>  
      }  
    </MudContainer>  
  </DialogContent>  
  <DialogActions>  
    <MudButton Color="Color.Primary" onClick="Ok">Accept</MudButton>  
  </DialogActions>  
</MudDialog>
```



- Blurry Dialog



Pada MudDialog, kita dapat mengganti background dengan menggunakan “ClassBackground“. Terlebih dahulu kita buat sebuah class pada css. Lalu masukkan nama class tersebut pada ClassBackground, berikut contoh penggunaannya.

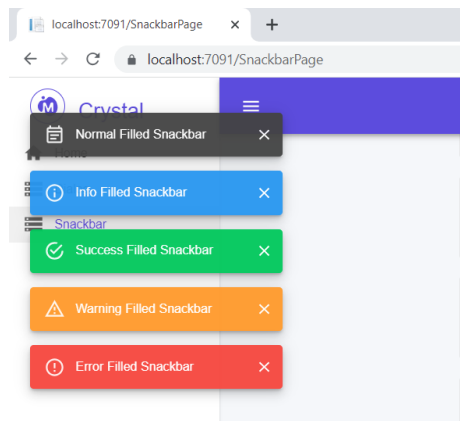
```
<MudButton @onclick="OpenDialogBlur" Variant="Variant.Filled"
Color="Color.Primary">
    Open Simple Dialog
</MudButton>

<style>
    .my-custom-class {backdrop-filter: blur(10px);}
</style>

private void OpenDialogBlur()
{
    var options = new DialogOptions { ClassBackground = "my-custom-class" };
    _dialogService.Show<SampleDialog>("Simple Dialog", options);
}
```

## Snackbar

Snackbar atau biasa juga disebut dengan Toast adalah sebuah alert untuk menampilkan sebuah pesan secara dinamik pada pengguna didalam aplikasi web.



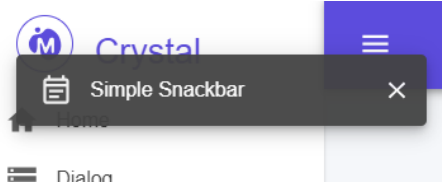
Snackbar pada MudBlazor menggunakan `ISnackbar Service` dan `"MudSnackbarProvider"`, Sebelumnya kita harus melakukan beberapa langkah sebelum menggunakan Snackbar. Tambahkan `"<MudSnackbarProvider/>"` pada `MainLayout.razor` (dibawah `MudLayout` element). Selanjutnya buka `_Imports.razor` dan ketikkan sintak seperti berikut `"@inject ISnackbar _snackBar"`. Setelah selesai dengan langkah-langkah diatas, kita sudah bisa menggunakan komponen Snackbar pada aplikasi. Berikut akan dijelaskan beberapa fitur pada Snackbar.

### - Basic

Berikut adalah cara paling sederhana dalam menggunakan Snackbar.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary"
@onclick="@(() => _snackBar.Add("Simple Snackbar"))">
    Open Snackbar
</MudButton>
```

Untuk menampilkan simple Snackbar kita gunakan sintaks berikut. “\_snackBar.Add("Simple Snackbar)“. Cukup dengan memanggil method “Add“ dan isi paramater message dari method tersebut.



- HTML pada isi pesan

Untuk isi dari pesan kita dapat menggunakan sintaks HTML. Pesan snackbar dirender sebagai HTML menggunakan MarkupString.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary" @onClick="@((C) =>
_snackBar.Add($"<ul><li>Item {++count}</li><li>Item {++count}</li></ul>"))">
    Open Snackbar
</MudButton>
```

Pada sintaks diatas, dapat dilihat isi pesan pada Snackbar menggunakan HTML. Sehingga menghasilkan tampilan seperti pada gambar dibawah.

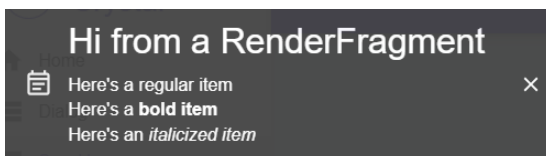


- RenderFragment

Selain menyetel pesan Snackbar dengan string, Anda juga dapat menggunakan RenderFragments untuk membuat Snackbar dengan konten yang lebih kompleks.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary"
OnClick="@OnClickRenderFragment">Show a RenderFragment Snackbar</MudButton>
```

```
private void OnClickRenderFragment()
{
    _snackBar.Add
    (
        @<div>
            <h3>Test RenderFragment</h3>
            <ul>
                <li>Here's a regular item</li>
                <li>Here's a <strong>bold item</strong></li>
                <li>Here's an <em>italicized item</em></li>
            </ul>
        </div>
    );
}
```

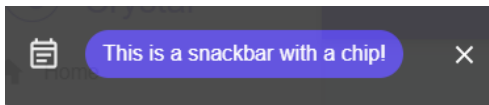




- Custom component

Selain menggunakan HTML dan RenderFragment, kita juga bisa menambahkan custom component pada snackbar. Pada contoh dibawah dapat dilihat, untuk pesan di snackbar kita menggunakan komponen “MudChip“.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary"
OnClick="@OnClickCustom">Show a snackbar with a custom component</MudButton>
private void OnClickCustom()
{
    _snackBar.Add<MudChip>(new Dictionary<string, object>() {
        { "Text", "This is a snackbar with a chip!" },
        { "Color", Color.Primary }
    });
}
```



- Severity

Secara default snackbar akan menggunakan normal Severity dan warna background. Untuk mengganti Severity pada Snackbar, cukup dengan menambahkan severity enum setelah pesan. Terdapat 5 pilihan yaitu Normal, Info, Success, Warning dan Error. Untuk detail nya ada pada sintaks dibawah ini.

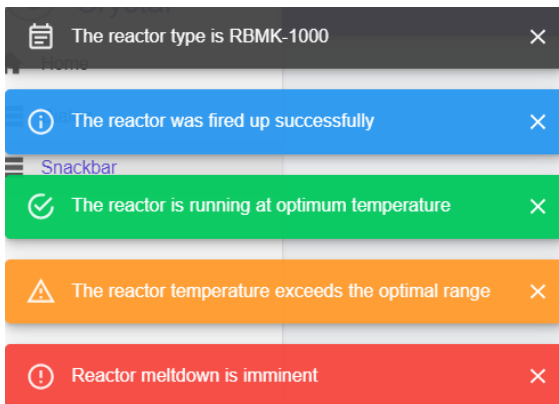
```
<MudButton Color="Color.Dark"
@onclick="@(() => _snackBar.Add("The reactor type is RBMK-1000", Severity.Normal))">Normal
Snackbar</MudButton>

<MudButton Color="Color.Info" @onclick="@(() => _snackBar.Add("The reactor was fired up
successfully", Severity.Info))">Info Snackbar</MudButton>

<MudButton Color="Color.Success" @onclick="@(() => _snackBar.Add("The reactor is running at
optimum temperature", Severity.Success))">Success Snackbar</MudButton>

<MudButton Color="Color.Warning" @onclick="@(() => _snackBar.Add("The reactor temperature
exceeds the optimal range", Severity.Warning))">Warning Snackbar</MudButton>

<MudButton Color="Color.Error" @onclick="@(() => _snackBar.Add("Reactor meltdown is
imminent", Severity.Error))">Error Snackbar</MudButton>
```



## - Configuration

Seperti MudDialog, untuk Snackbar kita juga dapat melakukan konfigurasi dengan 2 cara yaitu.

### 1. Global

Jika kita melakukan konfigurasi secara global maka berefek pada seluruh snackbar di aplikasi. Kita dapat melakukannya pada program.cs atau Startup.cs bergantung pada project yang kita miliki.

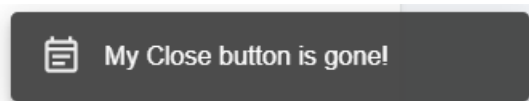
```
builder.Services.AddMudServices(config =>
{
    config.SnackbarConfiguration.PositionClass = Defaults.Classes.Position.BottomLeft;

    config.SnackbarConfiguration.PreventDuplicates = false;
    config.SnackbarConfiguration.NewestOnTop = false;
    config.SnackbarConfiguration.ShowCloseIcon = true;
    config.SnackbarConfiguration.VisibleStateDuration = 10000;
    config.SnackbarConfiguration.HideTransitionDuration = 500;
    config.SnackbarConfiguration.ShowTransitionDuration = 500;
    config.SnackbarConfiguration.SnackbarVariant = Variant.Filled;
});
```

### 2. Per Snackbar

Sedangkan dengan cara ini hanya berefek pada 1 snackbar yang telah ganti konfigurasinya. Sebagai contoh, pada Global setting kita set ShowCloseIcon adalah true. Dan pada Snackbar ini kita ganti menjadi false, maka untuk Snackbar ini akan menampilkan pesan tanpa close button.

```
<MudButton @onClick="@(() => _snackBar.Add("My Close button is gone!",
Severity.Normal, config => { config.ShowCloseIcon = false; })))"
Variant="Variant.Filled" Color="Color.Primary">
    Open Modified Snackbar
</MudButton>
```



## - Position

Secara bawaan posisi Snackbar pada MudBlazor akan tampil pada kiri bawah layar. Tetapi kita dapat mengubah posisi dari snackbar, dengan mengganti konfigurasi dari "PositionClass" seperti berikut " \_snackBar.Configuration.PositionClass = position;"

```
<MudButton @onclick="@(() => ChangePosition("Top-Start",  
Defaults.Classes.Position.TopStart))" Color="Color.Primary">Top-Start</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Top-Left",  
Defaults.Classes.Position.TopLeft))" Color="Color.Primary">Top-Left</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Top-Center",  
Defaults.Classes.Position.TopCenter))" Color="Color.Primary">Top-Center</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Top-Right",  
Defaults.Classes.Position.TopRight))" Color="Color.Primary">Top-Right</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Top-End",  
Defaults.Classes.Position.TopEnd))" Color="Color.Primary">Top-End</MudButton>  
<MudDivider />  
<MudButton @onclick="@(() => ChangePosition("Bottom-Start",  
Defaults.Classes.Position.BottomStart))" Color="Color.Default">Bottom-  
Start</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Bottom-Left",  
Defaults.Classes.Position.BottomLeft))" Color="Color.Default">Bottom-Left</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Bottom-Center",  
Defaults.Classes.Position.BottomCenter))" Color="Color.Default">Bottom-  
Center</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Bottom-Right",  
Defaults.Classes.Position.BottomRight))" Color="Color.Default">Bottom-  
Right</MudButton>  
<MudButton @onclick="@(() => ChangePosition("Bottom-End",  
Defaults.Classes.Position.BottomEnd))" Color="Color.Default">Bottom-End</MudButton>  
  
void ChangePosition(string message, string position)  
{  
    _snackBar.Clear();  
    _snackBar.Configuration.PositionClass = position;  
    _snackBar.Add(message, Severity.Normal);  
}
```

#### - Variants

Terdapat 3 buah variant snackbar pada mudblazor, yang pertama Text, Filled dan Outlined. Untuk menggunakan ataupun mengganti variant pada snackbar kita gunakan "SnackbarVariant" pada konfigurasi. Seperti pada contoh dibawah.

```
<MudButton @onclick="@(() => ChangeVariant("Text Snackbar", Variant.Text))"  
Color="Color.Primary">Open Text Snackbar Variants</MudButton>  
<MudButton @onclick="@(() => ChangeVariant("Filled Snackbar", Variant.Filled))"  
Color="Color.Secondary">Open Filled Snackbar Variants</MudButton>  
<MudButton @onclick="@(() => ChangeVariant("Outlined Snackbar", Variant.Outlined))"  
Color="Color.Tertiary">Open Outlined Snackbar Variants</MudButton>  
  
void ChangeVariant(string message, Variant variant)  
{  
    _snackBar.Configuration.SnackbarVariant = variant;  
    _snackBar.Configuration.MaxDisplayedSnackbars = 10;  
    _snackBar.Add($"Normal {message}", Severity.Normal);  
    _snackBar.Add($"Info {message}", Severity.Info);  
    _snackBar.Add($"Success {message}", Severity.Success);  
    _snackBar.Add($"Warning {message}", Severity.Warning);  
    _snackBar.Add($"Error {message}", Severity.Error);  
}
```

#### - CloseAfterNavigation

Secara default Snackbar akan terus muncul sampai pengguna yang menutupnya. Jika kita mengganti option pada "CloseAfterNavigation" menjadi true. Maka snackbar akan ditutup setelah pengguna mengganti atau keluar dari halaman saat ini.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary"
@onClick="@AddCloseAfterNavSnackbar">
    Closes After Navigation
</MudButton>

<MudButton Variant="Variant.Filled" Color="Color.Primary" @onClick="@AddSnackbar">
    Default Snackbar
</MudButton>
public void AddSnackbar()
{
    _snackBar.Add("Remains open after navigation.", Severity.Normal);
}

public void AddCloseAfterNavSnackbar()
{
    _snackBar.Add("Will close after navigation.", Severity.Normal, (options) =>
    {
        options.CloseAfterNavigation = true;
    });
}
```

#### - Custom Action Handling

Selain sebagai alert untuk menyampaikan informasi dari suatu aplikasi, kita juga dapat menambahkan fungsi pada snackbar sehingga menjadi dapat diklik, handler ditentukan dalam properti `OnClick` dari objek `SnackbarOptions`.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary" OnClick="@Show">
    Open Clickable Snackbar
</MudButton>

void Show()
{
    _snackBar.Add("I got a message for you", Severity.Normal, config =>
    {
        config.Onclick = snackbar =>
        {
            SayHello();
            return Task.CompletedTask;
        };
    });
}

public void SayHello()
{
    _snackBar.Add("Hello World!");
}
```

Pada sintaks diatas dapat kita lihat terdapat properti "OnClick" dengan menggunakan method "Show". Pada method tersebut akan menampilkan pesan, sekaligus kita set properti "OnClick" kembali dengan memanggil method "SayHello"

Terdapat Alternatif lain yaitu handler `OnClick` memanggil method yang menggunakan Action yang kita set dari konfigurasi. Seperti pada sintaks dibawah ini.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary" OnClick="@ShowWithButton">  
    Open Action Snackbar  
</MudButton>
```

```
void ShowWithButton()  
{  
    _snackBar.Add("Oops. Something really bad happened!", Severity.Normal, config =>  
    {  
        config.Action = "Help";  
        config.ActionColor = Color.Primary;  
        config.OnClick = snackbar =>  
        {  
            Help();  
            return Task.CompletedTask;  
        };  
    });  
}  
  
public void Help()  
{  
    _snackBar.Add("Please check the reactor temperature and try again");  
}
```

- Icon

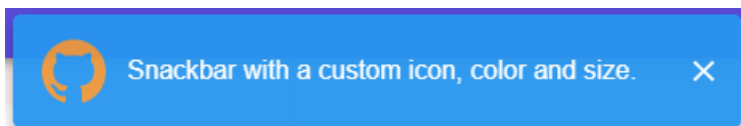
Pada Snackbar kita dapat mengganti ataupun menyembunyikan default icon.

Untuk menyembunyikan icon pada snackbar, pada konfigurasi kita set "HideIcon" menjadi true, seperti pada contoh dibawah.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary" @onclick="@((C) =>  
_snackBar.Add("Snackbar with no icon.", Severity.Success,  
config => { config.HideIcon = true; })))">  
    Open Snackbar (No Icon)  
</MudButton>
```

Sedangkan untuk mengganti icon, kita set beberapa atribut pada konfigurasi seperti sintaks dibawah ini.

```
<MudButton Variant="Variant.Filled" Color="Color.Primary" @onclick="@((C) =>  
_snackBar.Add("Snackbar with a custom icon, color and size.", Severity.Info, config =>  
{  
    config.Icon = Icons.Custom.Brands.GitHub;  
    config.IconColor = Color.Warning;  
    config.IconSize = Size.Large;  
}))">  
    Open Snackbar (Custom Icon)  
</MudButton>
```

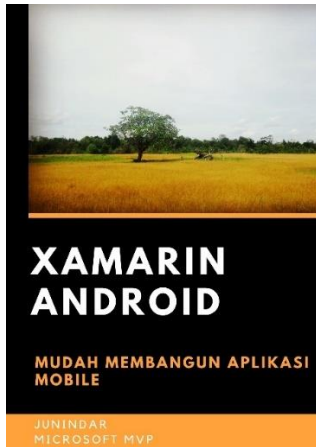


## Penutup

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

<https://junindar.blogspot.com/2023/07/mudblazor-dialog-dan-snackbar-pada.html>

## Referensi



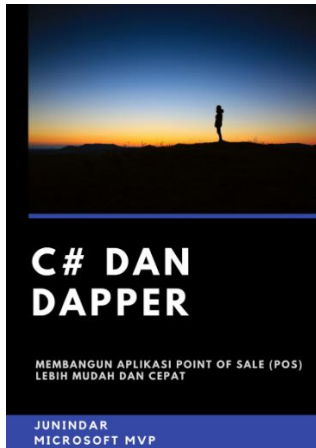
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



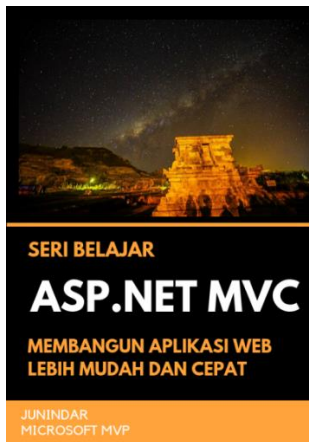
<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>



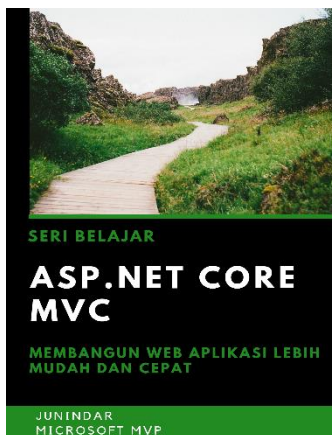
[https://play.google.com/store/books/details/Junindar\\_Xamarin\\_Forms?id=6Wg-DwAAQBAJ](https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ)



[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)

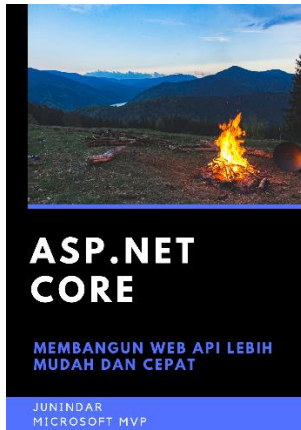


[https://play.google.com/store/books/details/Junindar\\_ASP\\_NET\\_MVC\\_Membangun\\_Aplikasi\\_Web\\_Lebih?id=XLlyDwAAQBAJ](https://play.google.com/store/books/details/Junindar_ASP_NET_MVC_Membangun_Aplikasi_Web_Lebih?id=XLlyDwAAQBAJ)



[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)

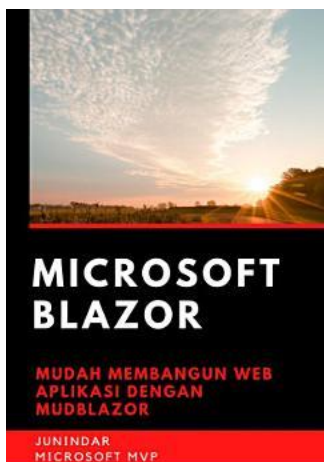




[https://play.google.com/store/books/details/Junindar\\_ASP\\_NET\\_CORE?id=COUWEAAQBAJ](https://play.google.com/store/books/details/Junindar_ASP_NET_CORE?id=COUWEAAQBAJ)



[https://play.google.com/store/books/details/Junindar\\_Microsoft\\_Blazor\\_Membangun\\_Aplikasi\\_Web\\_D?id=HKZhEAAAQBAJ](https://play.google.com/store/books/details/Junindar_Microsoft_Blazor_Membangun_Aplikasi_Web_D?id=HKZhEAAAQBAJ)



<https://play.google.com/store/books/details?id=q9usEAAAQBAJ&pli=1>

## 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”**.