IlmuKomputer.Com
Ikatlah ilmu dengan menuliskannya

MudBlazor Chart Component

Pada Blazor

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 mapun 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.

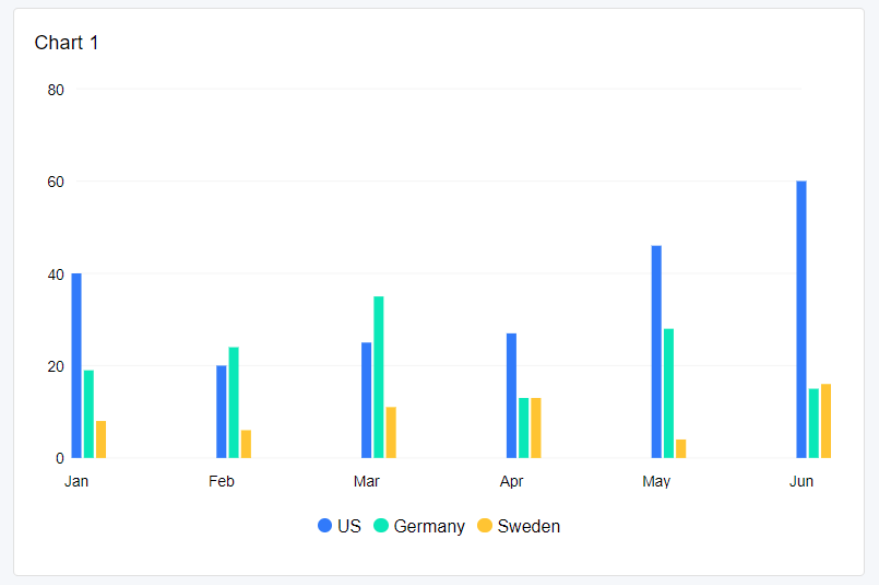
Chart atau grafik adalah representasi grafis untuk visualisasi data, di mana data diwakili oleh simbol, seperti batang dalam bagan batang, garis dalam bagan garis, atau irisan dalam bagan pai. Chart sering digunakan untuk memudahkan pemahaman tentang sejumlah besar data dan hubungan antara bagian-bagian data.

Grafik biasanya dapat dibaca lebih cepat daripada data mentah. Mereka digunakan dalam berbagai bidang, dan dapat dibuat dengan tangan (seringkali di atas kertas grafik) atau dengan komputer menggunakan aplikasi charting.

Jenis chart tertentu lebih berguna untuk menyajikan kumpulan data tertentu daripada yang lain. Misalnya, data yang menyajikan persentase dalam kelompok yang berbeda (seperti "puas, tidak puas, tidak yakin") sering ditampilkan dalam diagram lingkaran, tetapi mungkin lebih mudah dipahami jika disajikan dalam diagram batang horizontal.

Pada MudBlazor komponen chart disebut dengan nama “MudChart“. Terdapat 4 jenis chart yang tersedia pada MudBlazor, yaitu Bar, Donut, Line dan Pie chart.

Sebelum memulai latihan-latihan pada artikel ini, pastikan telah menyelesaikan latihan-latihan pada artikel sebelumnya yang dapat dilihat disini (http://junindar.blogspot.com/2022/10/mudblazor-input-component-pada-blazor.html), (http://junindar.blogspot.com/2022/12/mudblazor-input-component-pada-blazor.html) dan (http://junindar.blogspot.com/2022/12/mudblazor-input-component-picker-pada.html).



Gambar diatas adalah contoh Bar chart pada MudBlazor. Dimana terdapat 3 buag Series (US, Germany dan Sweden). Sedangkan untuk XAxis nya adalah nama-nama bulan sampai dengan bulan Juni. Untuk mendapatkan hasil seperti pada Bar Chart diatas kita gunakan sintaks dibawah. Ini.

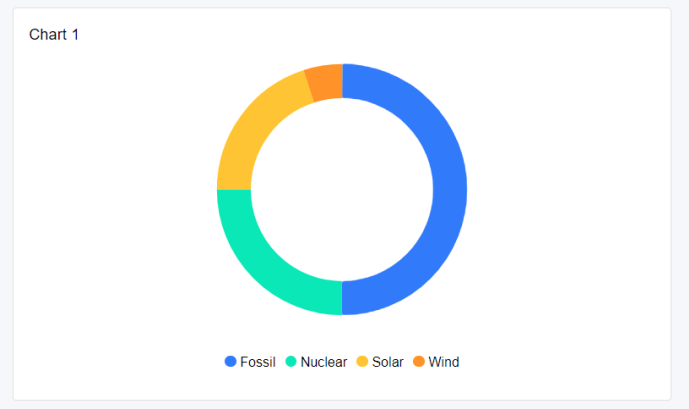
```
<MudChart ChartType="MudBlazor.ChartType.Bar" ChartSeries="@Series"
XAxisLabels="@XAxisLabels" Width="100%" Height="350px"
LegendPosition="Position.Bottom"></MudChart>

@code {
public List<ChartSeries> Series = new List<ChartSeries>()
{
  new() { Name = "US", Data = new double[] { 40, 20, 25, 27, 46, 60 } },
  new() { Name = "Germany", Data = new double[] { 19, 24, 35, 13, 28, 15 } },
  new() { Name = "Sweden", Data = new double[] { 8, 6, 11, 13, 4, 16 } },
};
public string[] XAxisLabels = { "Jan", "Feb", "Mar", "Apr", "May", "Jun"};
}
```

Pada sintaks diatas data yang digunakan (Series maupun XAxis) masih bersifat statis atau hardcode. Karena untuk mendapatkan data dari Database maupun WebApi telah dijelaskan pada artikel-artikel sebelumnya. Beberapa property yang harus diketahui pada Bar Chart MudBlazor. Yang pertama adalah “ChartType”, property ini digunakan untuk memilih jenis dari chart yang tersedia pada MudBlazor. Selanjutnya adalah ChartSeries, series adalah kunci untuk menampilkan Chart. Semua data diplot dalam chart sebagai rangkaian dan setidaknya memiliki satu series. Pada MudBlazor terdapat class “ChartSeries” yang memiliki property Name dan Data. Kita dapat membuat Series pada C# dan nantinya akan digunakan oleh ChartSeries pada MudChart. Lalu property XAxisLabel, X Axis biasa juga disebut category axis yang biasa digunakan untuk

horizontal axis. Sedangkan untuk menentukan posisi dari Legend, kita gunakan property "LegendPosition". Untuk contoh diatas posisi legend berada dibawah chart (Bottom).

Selanjutnya kita akan membuat Donut Chart dengan MudBlazor, seperti dibawah.



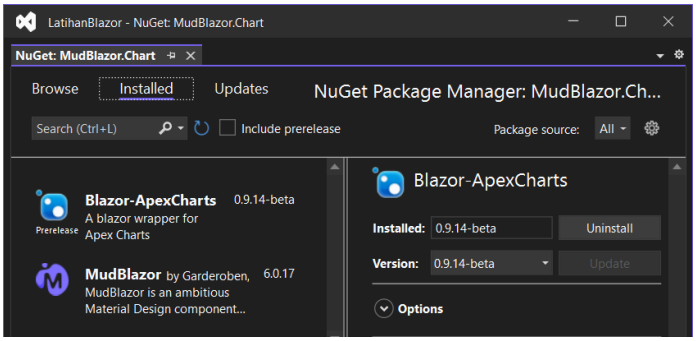
Untuk Donut Chart terdapat dua property yang penting, yaitu "InputData" dan "InputLabels". Seperti pada contoh sintaks dibawah ini.

```
<MudChart ChartType="MudBlazor.ChartType.Donut" Width="300px" Height="300px"
InputData="@data" InputLabels="@labels"></MudChart>
@code {
    public double[] data = { 50, 25, 20, 5 };
    public string[] labels = { "Fossil", "Nuclear", "Solar", "Wind" };
}
```

Fitur-fitur chart yang disediakan pada pada MudBlazor tidak banyak atau masih standar. Oleh karena itu pada artikel ini kita akan menggunakan Component yang khusus untuk Chart yaitu "ApexChart".

ApexCharts adalah Library untuk membuat modern chart, yang membantu developer membuat visualisasi yang indah dan interaktif pada halaman web. ApexCharts pada blazor adalah pengembangan dari ApexCharts pada javascript.

Pertama-tama kita perlu menambahkan library ini kedalam project. Pada Nuget Package Manager cari dan install "Blazor-ApexCharts" library.



Selanjutnya tambahkan script dibawah pada "_Layout.cshtml", setelah "_framework/blazor.server.js".

```
<script src="_content/Blazor-ApexCharts/js/apex-charts.min.js"></script>
<script src="_content/Blazor-ApexCharts/js/blazor-apex-charts.js"></script>
```

Lalu tambahkan reference "`@using ApexCharts`" pada _Imports.razor. Pada ApexCharts terdapat banyak jenis chart yang dapat digunakan. Tetapi artikel ini hanya akan membahas 4 jenis chart seperti dibawah.

- Bar Chart

  Bar Chart adalah salah satu jenis chart yang paling umum digunakan untuk menampilkan perbandingan antara beberapa kategori data dan variasi nilai yang berbeda. Bar chart diorientasikan secara horizontal atau vertikal menggunakan bar persegi panjang dengan panjang berbeda yang sebanding dengan nilai yang divisualisasikan.

1. Basic

   Dibawah ini adalah contoh sintaks untuk membuat basic Bar Chart menggunakan ApexChart.

```
<ApexCharts.ApexChart TItem="Order" Title="Order Net Value">

   <ApexPointSeries TItem="Order" Items="Orders"
   Name="Gross Value" XValue="@(e => e.Country)"
   YValue="@(e=> e.GrossValue)"
   SeriesType="SeriesType.Bar"/>

</ApexCharts.ApexChart>
```

   Dapat dilihat pada sintaks diatas, bar chart hanya memiliki sebuah series dengan nama “Gross Value”. Dengan nilai X di ambil dari property Country pada Class Order dan untuk nilai Y menggunakan property GrossValue. Class Order digunakan untuk property TItem. Untuk sintaks C# dapat dilihat dibawah ini.

```
public class Order
{
     public string Country { get; set; } = "";
     public decimal GrossValue { get; set; }
     public decimal NetValue { get; set; }
}
```

   Kita buat terlebih dahulu class Order seperti diatas. Class ini nantinya akan digunakan untuk latihan-latihan berikutnya. Pada latihan ini, kita hanya menggunakan property Country dan GrossValue saja.

```
static List<Order> GetOrders()
{
  List<Order> list = new List<Order>()
  {
    new Order { Country = "France",GrossValue = 5500,NetValue = 4500},
    new Order { Country = "Brazil",GrossValue = 6500,NetValue = 7800},
    new Order { Country = "Sweden",GrossValue = 7800,NetValue = 3670},
    new Order { Country = "Spain",GrossValue = 1400,NetValue = 4500}
  };

  return list;
}
```

   Lalu kita buat sebuah function dengan nama “GetOrders” seperti diatas.

```
private List<Order> Orders { get; set; } = GetOrders();
```

   Dan hasil dari function “GetOrders” ditampung oleh property Orders. Selanjutnya nilai dari “Orders” yang digunakan oleh property “Items” dari “`ApexPointSeries`”.

2. Multiple Series

   Selanjuntya kita buat Bar Chart yang memiliki 2 series (Gross dan Net Value). Maka kita perlu tambahkan “`ApexPointSeries`” pada Chart seperti sintaks dibawah.

```
<ApexPointSeries TItem="Order" Items="Orders"
Name="Net Value" XValue="@(e => e.Country)"
YValue="@(e => e.NetValue)" SeriesType="SeriesType.Bar"/>
```

Series diatas digunakan untuk menampilkan data dari property NetValue pada class Order. Untuk hasilnya dapat dilihat pada chart kedua pada gambar diatas.

Untuk mengganti warna dari masing-masing series kita dapat gunakan property "Color" yang terdapat pada "`ApexPointSeries`", seperti contoh dibawah ini.

```
<ApexPointSeries TItem="Order"
                 Items="Orders"
                 Name="Gross Value"
                 XValue="@(e => e.Country)"
                 YAggregate="@(e => e.Sum(e => e.GrossValue))"
                 SeriesType="SeriesType.Bar"
                 Color = "#3633FF"/>

<ApexPointSeries TItem="Order"
                 Items="Orders"
                 Name="Net Value"
                 XValue="@(e => e.Country)"
                 YAggregate="@(e => e.Sum(e => e.NetValue))"
                 SeriesType="SeriesType.Bar"
                 Color= "#E51C15"/>
```

3. Horizontal Bar Chart

   Ikuti langkah-langkah dibawah ini, untuk mengubah Bar Chart dari Vertical menjadi Horizontal. Pertama-tama buat sebuah field dengan menggunakan class "`ApexChartOptions`" seperti dibawah

```
private ApexChartOptions<Order> options;
```

   Lalu kita buat override method "`OnInitialized`", dan kita atur nilai Horizontal pada PlotOptionBar menjadi true.

```
protected override void OnInitialized()
{
  options = new ApexChartOptions<Order>
  {
     PlotOptions = new PlotOptions
     {
        Bar = new PlotOptionsBar
        {
           Horizontal = true
        }
     }
  };
}
```

   Dan pada ApexChart tambahkan property "Option" lalu masukkan nilainya dengan menggunakan field "options" yang telah kita buat diatas.

4. Stacked dan Stacked 100%

   Stacked Bar Chart adalah jenis bar chart yang digunakan untuk memisahkan kategori yang lebih besar menjadi subkategori atau subskor dan membandingkannya untuk melihat subkategori atau subskor mana yang memiliki porsi lebih besar dari total.

Ada 2 varian Stacked Bar Charts.

Normal Stacked Bar Charts

Bar berada di atas satu sama lain berdasarkan nilainya. Untuk membuat normal stacked chart dapat kita gunakan sintaks C#, dengan menggunakan class "ApexChartOptions". Pada "ApexChartOptions" terdapat property dengan menggunakan class Chart. Dari class tersebut kita dapat mengaktifkan Stacked menjadi "true". Dari "ApexChartOptions" juga kita dapat mengganti bentuk bar chart menjadi Horizontal dan warna dari series.

```
private ApexChartOptions<Order> optionsStack;
protected override void OnInitialized()
{
        optionsStack = new ApexChartOptions<Order>
        {
            Chart = new Chart
            {
                Stacked = true,
            },
            PlotOptions = new PlotOptions
            {
                Bar = new PlotOptionsBar
                {
                    Horizontal = true
                }
            },
            Colors = new List<string> { "#5cb85c", "#d9534f" }
        };
}
```

100% Stacked Bar Charts

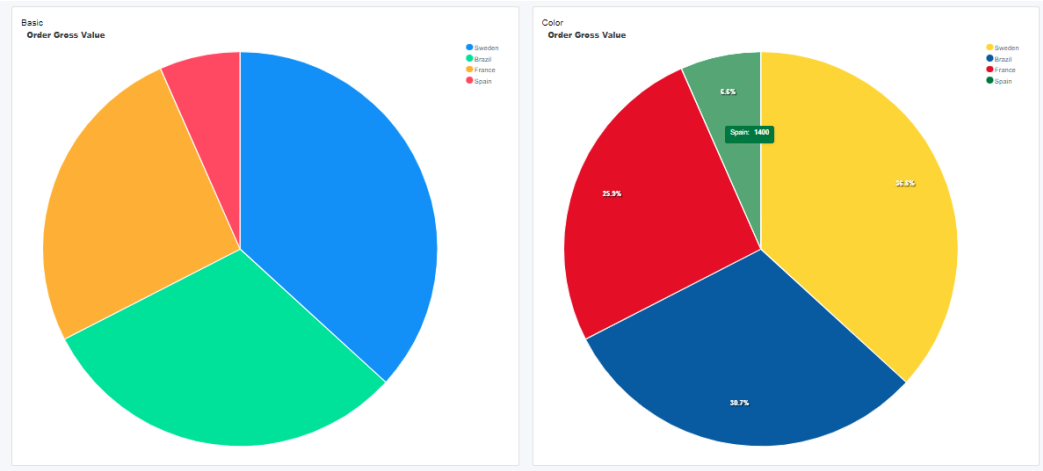
Bar berada di atas satu sama lain berdasarkan persentase atau nilainya di antara kategori.

```
private ApexChartOptions<Order> optionsStack100;
protected override void OnInitialized()
{
        optionsStack100 = new ApexChartOptions<Order>
        {
            Chart = new Chart
            {
                Stacked = true,
                StackType = StackType.Percent100,
                Toolbar = new Toolbar
                {
                    Show = false
                }
            },
        };
}
```

Untuk membuat Stacked 100%, kita gunakan StackType.Percent100. Selanjutnya kita juga dapat menambahkan Title pada Axis dan mengatur Font Style-nya. Untuk detail sintaksnya dapat dilihat pada project lampiran.

- Pie Chart

Pie Chart adalah alat visualisasi yang berguna dalam menampilkan data dan informasi dalam persentase atau rasio. Pie Chart paling efektif saat menangani sekumpulan kecil data. Semua itu dibangun dengan mengelompokkan seluruh data yang diperlukan ke dalam bingkai berbentuk lingkaran dimana data digambarkan dalam irisan.



1. Basic

```
<ApexCharts.ApexChart TItem="Order" Title="Order Gross Value">

 <ApexPointSeries TItem="Order" Items="Orders"
 SeriesType="SeriesType.Pie" Name="Gross Value"
 XValue="@(e => e.Country)" YAggregate="@(e => e.Sum(e => e.GrossValue))"
 OrderByDescending="e=>e.Y" />

</ApexCharts.ApexChart>
```

Yang perlu diperhatikan pada sintaks diatas adalah SeriesType dan YAggregate. Dimana pada SeriesType pilih "`SeriesType.Pie`" untuk tipenya, dan untuk YAggregate kita gunakan fungsi sum untuk GrosValue. Untuk menampilkan berdasarkan besaran dari nilai Y, dapat kita gunakan OrderBy atau OrderByDescending.

2. Color Point dan Label

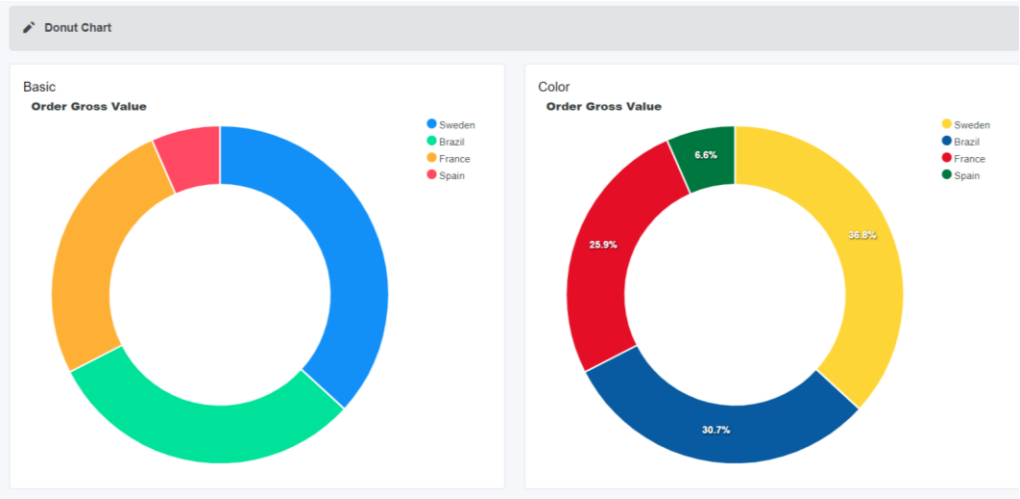
Pada latihan diatas warna yang digunakan adalah default dari sistem. Lalu untuk mengganti warna sesuai yang kita inginkan kita dapat menggunakan property "`PointColor`". Buat terlebih dahulu function "`GetPointColor`" seperti dibawah.

```
private string GetPointColor(Order order)
{
   switch (order.Country)
   {
     case "France":
          return "#e3001b";
     case "Brazil":
          return "#005ba3";
     case "Sweden":
          return "#ffd500";
     case "Spain":
          return "#00783c";
     default:
          return "#87775d";
   }
}
```

Disini fungsi ini kita tentukan warna untuk masing-masing Country. Dan terakhir pada ApexPointSeries tambahkan sebuah property "`PointColor`" dan panggi fungsi diatas seperti berikut "`PointColor="e=> GetPointColor(e)"`". Sedangkan untuk menampilkan label, gunakan sintaks berikut "`ShowDataLabels="true"`" pada ApexPointSeries.

- Donut Chart

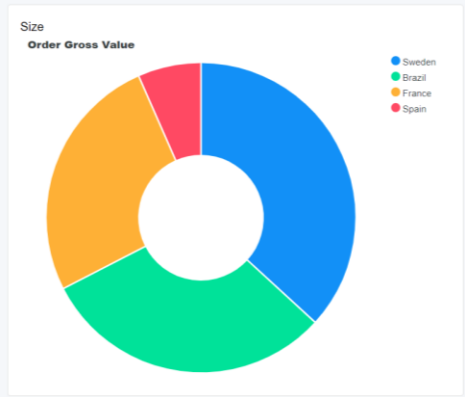
  Donut Chart mirip dengan Pie Chart di mana bagian tengah dari Chart kosong. Pie Chart dapat diubah menjadi Donut Chart dengan memodifikasi satu properti. Yaitu dengan mengganti SeriesType nya saja.



  Sintaks Pie chart diatas dapat kita implementasikan pada Donut Chart. Jadi untuk Donut Chart kita akan membuat bagaimana mengganti Size dan menampilkan Total Data ditengah-tengah Chart.

```
private ApexChartOptions<Order> options;
protected override void OnInitialized()
{
   options = new ApexChartOptions<Order>
   {
      PlotOptions = new PlotOptions
      {
         Pie = new PlotOptionsPie()
         {
             Donut = new PlotOptionsDonut()
             {
               Size = "40%"
             }
         }
      }
   };
}
```

  Untuk mengganti Size kita gunakan ApexChartOption seperti diatas. Untuk diketahui PlotOptionDonut berada pada PlotOptionPie, jadi sebelumnya kita buat terlebih dahulu PlotOptionPie baru diikuti dengan PlotOptionDonut. Dari situ baru bisa ganti Size dari donut seperti diatas. Sehingga mendapatkan hasil seperti dibawah.

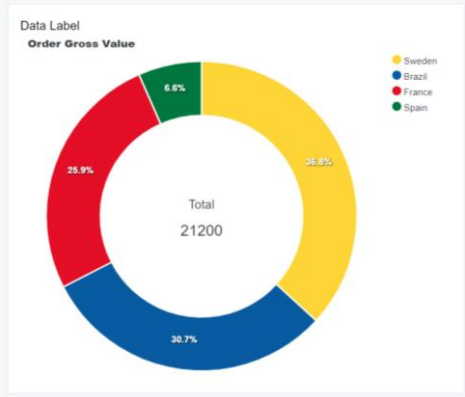


Seperti untuk mengganti Size, untuk menampilkan total nilai kita gunakan "PlotOptionDonut". Perhatikan detail sintaks dibawah.

```
Donut = new PlotOptionsDonut()
{
   Labels = new DonutLabels()
   {
        Total = new DonutLabelTotal()
        {
            Show = true
        }

   }
}
```

Sehingga mendapatkan hasil seperti dibawah.



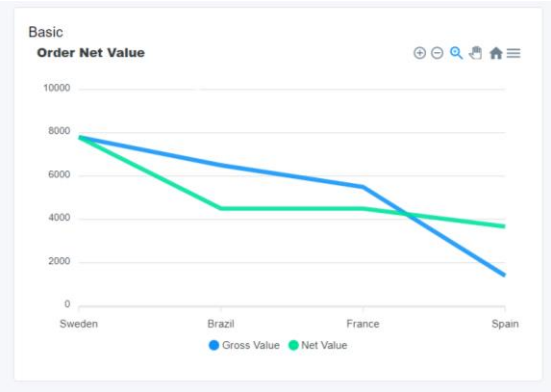
- Line Chart

  Line Chart adalah basic chart yang menggambarkan tren dan perilaku dari waktu ke waktu. Chart ini menampilkan informasi sebagai rangkaian titik data yang juga dikenal sebagai "penanda" yang terhubung dengan garis. Salahs satu kelebihan dari Line Chart adalah kemungkinan untuk memasukkan data yang besar ke dalam satu chart tanpa kesulitan dalam melihat atau memahami informasi yang disajikan.

Kapan Menggunakan Line Chart?

Ada beberapa skenario umum di mana Line Chart adalah opsi terbaik:

- Jika ingin membandingkan tren yang berbeda pada garis waktu.
- Jika ingin menunjukkan perbedaan antara 2 series data atau lebih.
- Jika ingin merepresentasikan data dengan deret waktu secara grafis dan mendetail.
- Jika ingin memvisualisasikan kumpulan data besar dan bervolume tinggi dan mengintegrasikan interaksi seperti Panning, Zooming, dan Drill-Down..



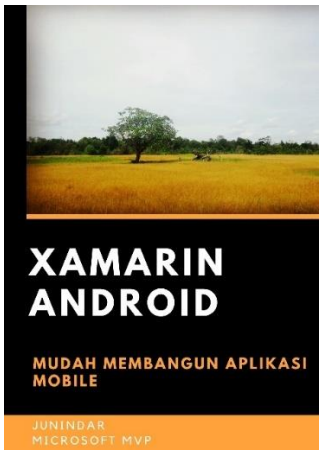
Untuk mendapatkan hasil seperti diatas, kita dapat menggunakan sintaks seperti pada Bar Chart. Hanya perlu mengganti tipe dari SeriesType menjadi Line.

Sedangkan untuk menampilkan Data Label, kita hanya perlu menggunakan property "`ShowDataLabels`" pada "`ApexPointSeries`". Lalu untuk mengubah style dari Label maupun background, kita gunakan "`ApexChartOptions`" yang detail sintaksnya dapat dilihat pada project lampiran.

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/2023/01/mudblazor-chart-component-pada-blazor.html

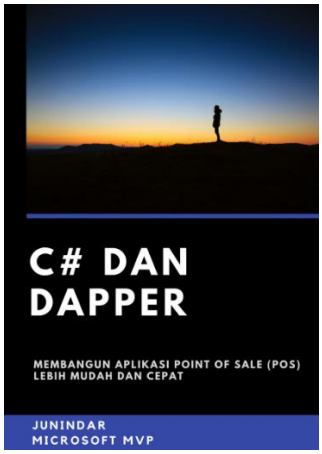
Referensi



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

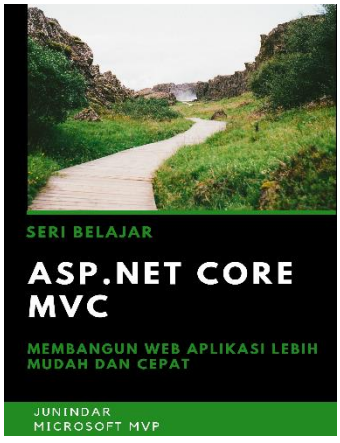
https://play.google.com/store/books/details?id=VSLiDQAAQBAJ

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



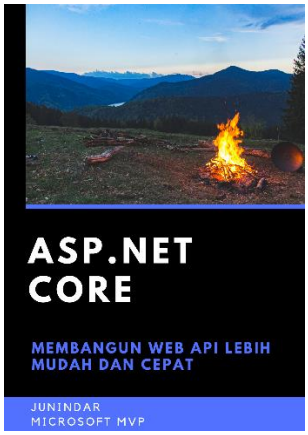
https://play.google.com/store/books/details/Junindar_C_dan_Dapper_Membangun_Aplikasi_POS_Point?id=6TErDwAAQBAJ

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

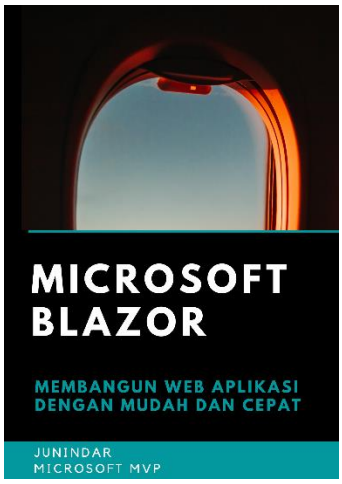


https://play.google.com/store/books/details/Junindar_ASP_NET_CORE_MVC?id=xEe5DwAAQBAJ

MudBlazor Chart Component Pada Blazor
Junindar, ST, MCPD, MOS, MCT, MVP .NET



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



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".**