

Telegram Bot Framework Custom

Control Part II

Junindar, ST, MCPD, MOS, MCT, MVP

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@gmail.com

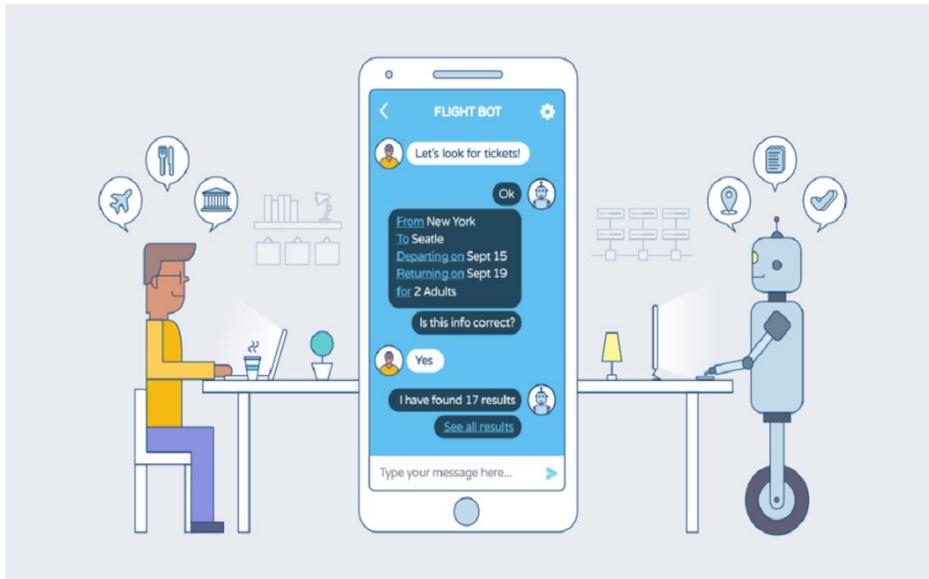
<http://junindar.blogspot.com>

Abstrak

Chatbot atau chatterbot adalah sebuah layanan obrolan robot/tokoh virtual dengan kecerdasan buatan atau AI (Artificial Intelligent) yang menirukan percakapan manusia melalui pesan suara, obrolan teks ataupun keduanya. Chatbot menggunakan berbagai teknologi yang mutakhir di dalamnya, seperti Artificial Intelligence (AI), Machine Learning, Deep Learning, dan Natural Language Processing (NLP). Machine Learning dapat di aplikasikan sebagai mesin yang mempelajari, menganalisa dan mengenal bermacam-macam Bahasa pada chatbot, sementara Natural Language Processing (NLP) memiliki kemampuan untuk mengerti dan memahami Bahasa manusia lalu memberikan respon yang sesuai dengan Bahasa yang digunakan pengguna chatbot.

Pendahuluan

Chatbot atau chatterbot adalah sebuah layanan obrolan robot/tokoh virtual dengan kecerdasan buatan atau AI (Artificial Intelligent) yang menirukan percakapan manusia melalui pesan suara, obrolan teks ataupun keduanya.



Pada dasarnya bots bekerja dengan cara melihat kata kunci dalam data yang masuk dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari basis data tekstual. Artinya, jika pengguna mengirim suatu permintaan maka bots akan membalasnya dengan respon yang spesifik sesuai dengan kata kunci yang dikirim.

Telegram Bot Framework

Telegram Bot Framework adalah sebuah framework yang memungkinkan para programmer untuk membuat bot pada Telegram dengan menggunakan platform .NET. Framework ini menyediakan tool dan fitur yang mempermudah pembuatan dan implementasi bot Telegram menggunakan bahasa pemrograman seperti C#.

Dengan menggunakan Telegram Bot Framework, kita dapat mengakses API Telegram, memproses pesan, menjawab permintaan dari pengguna, dan mengelola berbagai fungsi bot, seperti mengirim pesan teks, media, menangani perintah, dan banyak lagi.

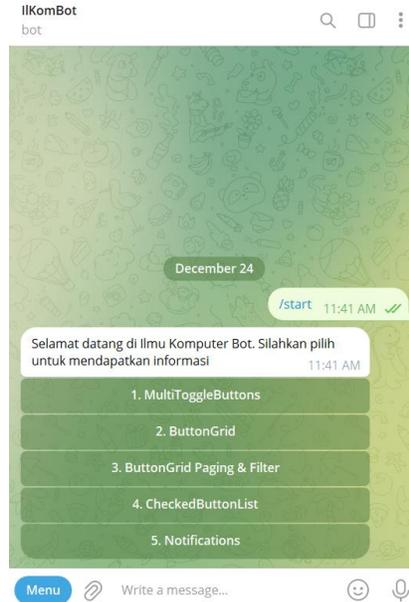
Terdapat banyak fasilitas pada framework ini, sehingga kita dapat mengembangkan bot yang powerfull dan fleksibel dengan menyediakan berbagai fungsi yang dapat digunakan untuk membuat bot yang sesuai dengan kebutuhan penggunaannya.

Telegram Bot Framework menggunakan beberapa library atau pustaka tertentu untuk memudahkan interaksi dengan API Telegram dan menyediakan berbagai fitur bagi para programmer. salah satunya adalah Telegram.Bot. Telegram.Bot adalah library utama yang digunakan dalam framework ini. Telegram.Bot merupakan library resmi dari Telegram untuk C#/.NET yang menyediakan akses ke API Telegram. library ini memungkinkan pengembang untuk membuat, mengelola, dan berinteraksi dengan bot Telegram menggunakan berbagai fitur yang disediakan oleh Telegram.

Pada latihan sebelumnya kita telah membuat chatbot sederhana dengan menggunakan beberapa control seperti Button, Progressbar, CalendarPicker, MonthPicker dan ToggleButton dengan menggunakan Telegram Bot Framework. Untuk latihan ini kita lanjutkan dengan bekerja menggunakan beberapa control lagi seperti MultiToggleButton, ButtonGrid/Paging, CheckedButtonL dan Notification. Mari ikuti langkah-langkah dibawah ini.

Note : Pada artikel ini, penulis menggunakan Visual Studio 2022 dan .Net 7.0. Dan pastikan telah mengikuti latihan pada artikel sebelumnya.

Artikel ini merupakan lanjutan dari artikel sebelumnya yang membahas control-control pada Telegram Bot Framework. Sehingga pada artikel ini tidak akan dijelaskan bagaimana cara untuk membuat menu dan action seperti pada gambar dibawah.

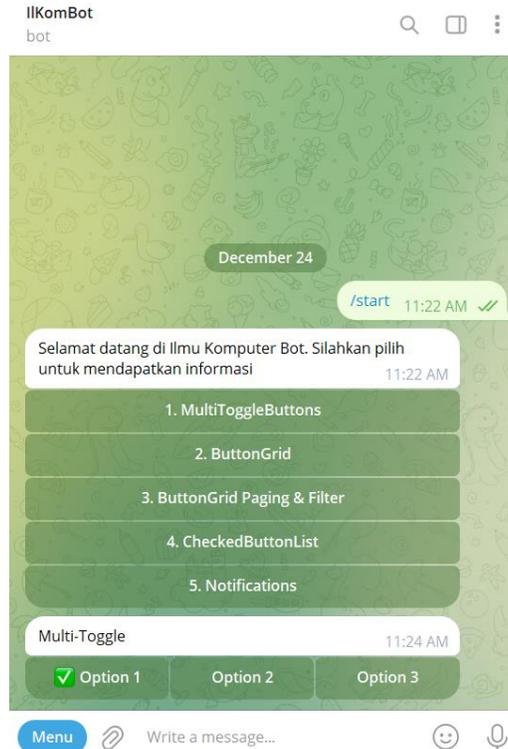


MultiToggleButton

MultiToggleButton adalah elemen UI yang memungkinkan pengguna untuk memilih beberapa opsi dari beberapa tombol yang tersedia. Dalam Telegram bot, MultiToggleButton dapat digunakan untuk membuat antarmuka yang memungkinkan pengguna untuk memilih beberapa opsi sekaligus dengan mengklik tombol yang sesuai dengan setiap opsi.

Ini dapat menjadi fitur yang berguna dalam membuat bot yang interaktif, di mana pengguna dapat memilih beberapa pilihan dari beberapa opsi yang disediakan, dan bot dapat merespons berdasarkan pilihan-pilihan ini.

Dengan menggunakan MultiToggleButton, pengguna dapat dengan mudah memilih beberapa opsi sekaligus tanpa harus mengetikkan perintah atau menjalani proses yang rumit, meningkatkan kemudahan penggunaan dalam berinteraksi dengan bot.



Tambahkan sebuah class dengan nama “MultiToggleButtons” dengan menggunakan “AutoCleanForm” sebagai base class. Dan buat dua buah method “ToggleButtons_Init” dan “Tb_Toggled”, seperti pada sintaks dibawah.

```
private void Tb_Toggled(object sender, EventArgs e)
{
    var tb = sender as MultiToggleButton;
    if (tb.SelectedOption != null)
    {
        Console.WriteLine("Toggle " + tb.Id.ToString() +
            " telah di-klik, dan option " +
            tb.SelectedOption.Value + " telah dipilih");
        return;
    }
}
```

Pada method diatas digunakan untuk mendapatkan Id dan Value dari ToggleButton yang dipilih. Lalu untuk membuat MultiToggleButton kita ketikkan sintaks dibawah ini.

```
private async Task ToggleButtons_Init(object sender, InitEventArgs e)
{
    var mtb = new MultiToggleButton();
    mtb.Options = new List<ButtonBase>() { new ButtonBase("Option 1", "1"),
        new ButtonBase("Option 2", "2"), new ButtonBase("Option 3", "3") };
    mtb.SelectedOption = mtb.Options.FirstOrDefault();
    mtb.Toggled += Tb_Toggled;
    this.AddControl(mtb);
}
```

Pada contoh ini, kita membuat 3 buah options pada MultiToggleButton. Dimana setiap option menggunakan class “ButtonBase“. Dan selanjutnya kita masukkan nilai untuk

parameter text dan value pada setiap ButtonBase. Dan terakhir pada constructor class MultiToggleButtons ketikkan sintaks berikut. Lalu jalankan program untuk mendapatkan hasil seperti pada gambar dibawah.

```
public MultiToggleButtons()  
{  
    this.DeleteMode = TelegramBotBase.Enums.EDeleteMode.OnLeavingForm;  
    this.Init += ToggleButtons_Init;  
}
```



ButtonGrid

Button Grid adalah susunan tombol yang terstruktur secara grid atau matriks. Ini memungkinkan pembuatan antarmuka pengguna yang lebih terstruktur dengan menampilkan sejumlah besar opsi dalam bentuk kotak-kotak atau grid yang teratur.

Pada Telegram bot, Button Grid dapat digunakan untuk menampilkan banyak tombol dalam tata letak grid yang terorganisir dengan baik di dalam pesan. Ini memungkinkan pengguna untuk memilih opsi yang tepat dengan lebih mudah daripada hanya menggunakan tombol-tombol yang ditampilkan secara berurutan.

Button Grid sering digunakan untuk membuat menu yang lebih terstruktur, memungkinkan pengguna untuk melihat banyak opsi sekaligus dalam satu layar dan dengan cepat memilih opsi yang diinginkan tanpa harus memasukkan perintah teks.

Penggunaan Button Grid dalam bot Telegram dapat mempermudah pengalaman pengguna dengan menyediakan antarmuka yang lebih visual dan mudah dipahami untuk berinteraksi dengan bot.



Tambahkan sebuah class “ButtonGridForm“ dengan menggunakan “AutoCleanForm“ sebagai base class dan buat variable class seperti dibawah.

```
ButtonGrid m_Buttons = null;
```

Disini kita gunakan class ButtonGrid class yang terdapat dalam library ini.

Lalu buat dua buah method seperti dibawah.

```
private async Task ButtonGridForm_Init(object sender, InitEventArgs e)
{
    m_Buttons = new ButtonGrid();
    m_Buttons.KeyboardType = TelegramBotBase.Enums.EKeyboardType.InlineKeyboard;
    ButtonForm bf = new ButtonForm();
    bf.AddButtonRow(new ButtonBase("Back", "back"), new ButtonBase("Switch Keyboard", "switch"));
    bf.AddButtonRow(new ButtonBase("Button1", "b1"), new ButtonBase("Button2", "b2"));
    bf.AddButtonRow(new ButtonBase("Button3", "b3"), new ButtonBase("Button4", "b4"));
    m_Buttons.DataSource = bf;
    m_Buttons.ButtonClicked += Bg_ButtonClicked;
    this.AddControl(m_Buttons);
}
```

Pada method diatas digunakan untuk membuat ButtonGrid. Sebelumnya kita set terlebih dahulu KeyboardType menjadi InlineKeyboard. Selanjutnya untuk membuat button kita gunakan “ButtonForm“ dengan menggunakan method “AddButtonRow“. Yang nantinya akan digunakan sebagai DataSource pada ButtonGrid (m_Buttons). Pada latihan ini kita akan menampilkan 6 buah button. Dimana terdapat 2 buah button pada setiap barisnya. Dan pada setiap button terdapat Text dan Value yang nantinya akan digunakan pada saat button dipilih atau diklik.

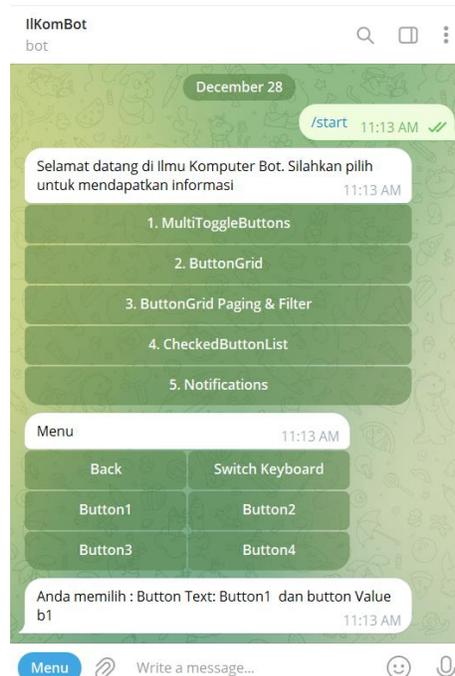
Lalu kita buat event untuk ButtonClicked, seperti pada sintaks dibawah ini.

```
private async Task Bg_ButtonClicked(object sender, ButtonClickedEventArgs e)
{
    if (e.Button == null)
        return;
    if (e.Button.Value == "back")
    {
        var start = new Menu();
        await this.NavigateTo(start);
    }
    else if (e.Button.Value == "switch")
    {
        switch (m_Buttons.KeyboardType)
        {
            case TelegramBotBase.Enums.EKeyboardType.ReplyKeyboard:
                m_Buttons.KeyboardType = TelegramBotBase.Enums.EKeyboardType.InlineKeyboard;
                break;
            case TelegramBotBase.Enums.EKeyboardType.InlineKeyboard:
                m_Buttons.KeyboardType = TelegramBotBase.Enums.EKeyboardType.ReplyKeyboard;
                break;
        }
    }
    else
    {
        await this.Device.Send($"Anda memilih : Button Text: {e.Button.Text}
        dan button Value {e.Button.Value}");
    }
}
}
```

Disini kita menggunakan Value untuk mengetahui button yang diklik oleh pengguna. Lalu pada constructor ketikkan sintaks dibawah.

```
public ButtonGridForm()
{
    this.DeleteMode = TelegramBotBase.Enums.EDeleteMode.OnLeavingForm;
    this.Init += ButtonGridForm_Init;
}
}
```

Jalankan program dan pastikan mendapatkan hasil seperti dibawah.



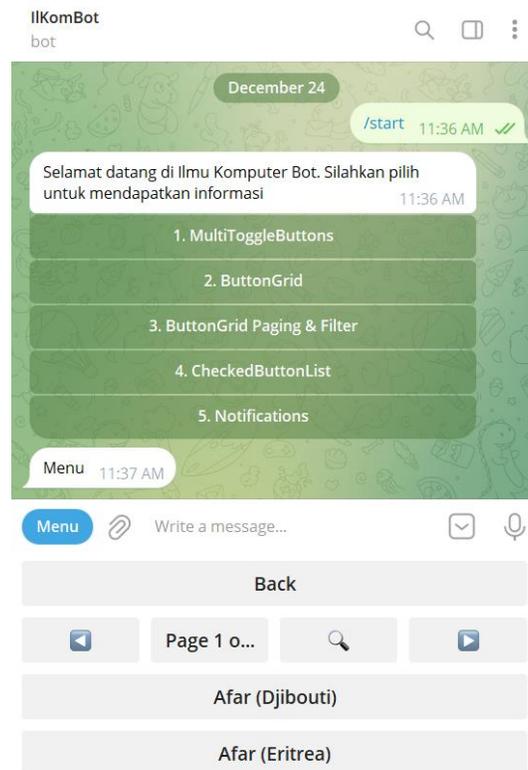
ButtonGrid Paging

Button Grid Paging adalah fitur yang memungkinkan penataan tombol dalam sebuah grid (kotak-kotak) pada antarmuka bot Telegram, dengan kemampuan untuk menavigasi melalui beberapa halaman tombol.

Dalam konteks ini, jika terdapat banyak opsi tombol yang ingin ditampilkan kepada pengguna dalam sebuah antarmuka grid, Button Grid Paging memungkinkan penggunaan beberapa halaman tombol untuk menampilkan pilihan-pilihan tersebut.

Ketika jumlah tombol melebihi kapasitas tampilan satu layar, grid tersebut akan dibagi menjadi beberapa halaman. Pengguna kemudian dapat berpindah antara halaman-halaman ini dengan menggunakan tombol navigasi seperti "next" (selanjutnya) atau "previous" (sebelumnya) untuk melihat tombol-tombol tambahan.

Dengan menggunakan fitur ini, bot dapat menampilkan banyak opsi kepada pengguna secara terstruktur dan terorganisir, memungkinkan navigasi antara halaman-halaman tombol untuk memilih opsi yang diinginkan. Ini membantu agar antarmuka bot tetap terorganisir tanpa membuatnya terlalu padat atau membingungkan bagi pengguna.



Tambahkan sebuah class dengan nama “ButtonGridPagingForm“ dengan menggunakan “AutoCleanForm“ sebagai base class. dan buat variable class seperti dibawah.

```
ButtonGrid m_Buttons = null;
```

Lalu kita lanjutkan dengan membuat sebuah method dengan nama “ButtonGridForm_Init“. Lalu ketikkan sintaks seperti dibawah ini.

```
private async Task ButtonGridForm_Init(object sender, EventArgs e)
{
    m_Buttons = new ButtonGrid();
    m_Buttons.KeyboardType = TelegramBotBase.Enums.EKeyboardType.ReplyKeyboard;
    m_Buttons.EnablePaging = true;
    m_Buttons.EnableSearch = true;
    m_Buttons.HeadLayoutButtonRow = new List<ButtonBase>()
    { new ButtonBase("Back", "back") };
    var countries = CultureInfo.GetCultures(CultureTypes.SpecificCultures);
    ButtonForm bf = new ButtonForm();
    foreach (var c in countries)
    {
        bf.AddButtonRow(new ButtonBase(c.EnglishName, c.EnglishName));
    }
    m_Buttons.DataSource = bf;
    m_Buttons.ButtonClicked += Bg_ButtonClicked;
    this.AddControl(m_Buttons);
}
```

Pada dasarnya proses yang dilakukan hampir sama seperti pada class “ButtonGridForm“. Tetapi pada latihan ini, kita akan menggunakan fitur seperti Paging, Search dan HeadLayout.

```
m_Buttons.EnablePaging = true;
m_Buttons.EnableSearch = true;
m_Buttons.HeadLayoutButtonRow = new List<ButtonBase>()
{ new ButtonBase("Back", "back") };
```

Selanjutnya sebagai latihan kita akan menampilkan nama-nama negara dengan menggunakan method “GetCultures“ pada class “CultureInfo“. Dengan menggunakan ButtonForm, akan kita masukkan setiap nilai (negara) yang didapat. Dan kita tampilkan pada setiap button (1 button pada setiap baris).

```
var countries = CultureInfo.GetCultures(CultureTypes.SpecificCultures);
ButtonForm bf = new ButtonForm();
foreach (var c in countries)
{
    bf.AddButtonRow(new ButtonBase(c.EnglishName, c.EnglishName));
}
m_Buttons.DataSource = bf;
```

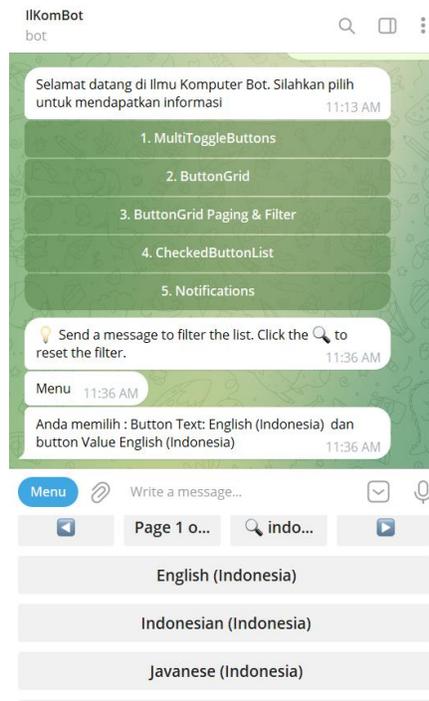
Lalu kita buat event untuk ButtonClicked, seperti pada sintaks dibawah ini.

```
private async Task Bg_ButtonClicked(object sender, ButtonClickedEventArgs e)
{
    if (e.Button == null)
        return;
    if (e.Button.Value == "back")
    {
        var start = new Menu();
        await this.NavigateTo(start);
    }
    else
    {
        await this.Device.Send($"Anda memilih : Button Text:
        {e.Button.Text} dan button Value {e.Button.Value}");
    }
}
```

Disini kita menggunakan Value untuk mengetahui button yang diklik oleh pengguna. Lalu pada constructor ketikkan sintaks dibawah.

```
public ButtonGridForm()
{
    this.DeleteMode = TelegramBotBase.Enums.EDeleteMode.OnLeavingForm;
    this.Init += ButtonGridForm_Init;
}
```

Jalankan program dan pastikan mendapatkan hasil seperti dibawah.



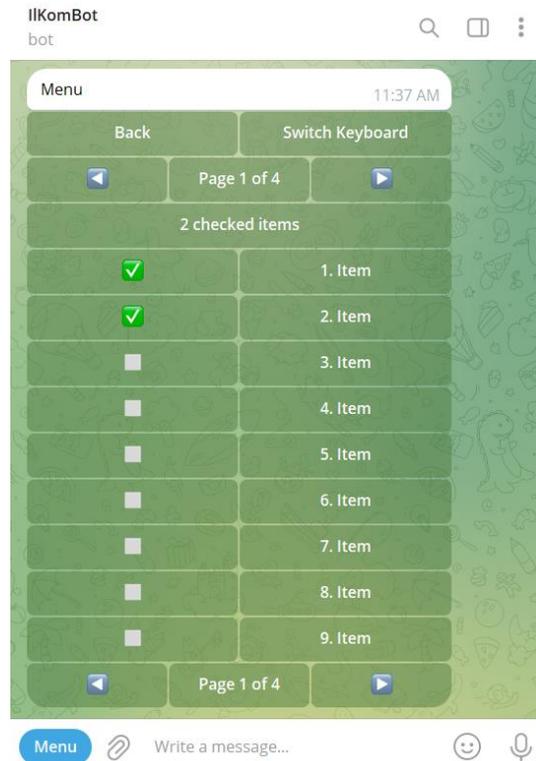
CheckedButtonList

CheckedButtonList pada kerangka kerja bot Telegram adalah fitur yang memungkinkan pembuatan daftar tombol yang dapat dicek (checked) oleh pengguna. Ini memungkinkan pengguna untuk memilih beberapa opsi dari daftar yang disediakan dengan menandai (mencentang) tombol yang sesuai dengan pilihannya.

Dalam konteks pengembangan bot Telegram, `CheckedButtonList` biasanya digunakan untuk membuat antarmuka pengguna yang memungkinkan pemilihan beberapa opsi sekaligus. Pengguna dapat menekan tombol-tombol di dalam daftar ini untuk memilih item atau opsi yang mereka inginkan.

Sebagai contoh, jika ada daftar opsi seperti "Makanan", "Minuman", dan "Pencuci Mulut", `CheckedButtonList` akan memungkinkan pengguna untuk memilih beberapa atau semua kategori ini dengan mencentang tombol yang sesuai.

Ini mempermudah interaksi pengguna dengan bot, terutama ketika ada banyak opsi yang tersedia, karena pengguna tidak perlu mengetikkan perintah atau melakukan input teks manual. Mereka bisa dengan mudah memilih opsi yang diinginkan dengan menekan tombol yang sesuai dalam daftar yang disediakan oleh `CheckedButtonList`.



Tambahkan sebuah class dengan nama “`CheckedButtonListForm`” dengan menggunakan “`AutoCleanForm`” sebagai base class. dan buat variable class seperti dibawah.

```
CheckedButtonList m_Buttons = null;
```

Lalu kita lanjutkan dengan membuat sebuah method dengan nama “`CheckedButtonListForm_Init`”. Lalu ketikkan sintaks seperti dibawah ini.

```
private async Task CheckedButtonListForm_Init(object sender, InitEventArgs e)
{
    m_Buttons = new CheckedButtonList();
    m_Buttons.KeyboardType = TelegramBotBase.Enums.EKeyboardType.InlineKeyboard;
    m_Buttons.EnablePaging = true;
    m_Buttons.HeadLayoutButtonRow = new List<ButtonBase>()
    { new ButtonBase("Back", "back"), new ButtonBase("Switch Keyboard", "switch") };
    m_Buttons.SubHeadLayoutButtonRow = new List<ButtonBase>()
    { new ButtonBase("No checked items", "$") };
    ButtonForm bf = new ButtonForm();
    for (int i = 0; i < 30; i++)
    {
        bf.AddButtonRow($"{i + 1}. Item", i.ToString());
    }
    m_Buttons.DataSource = new ButtonFormDataSource(bf);
    m_Buttons.ButtonClicked += Bg_ButtonClicked;
    m_Buttons.CheckedChanged += M_Buttons_CheckedChanged;
    this.AddControl(m_Buttons);
}
```

Pada method diatas digunakan untuk membuat CheckedButtonList. Sebelumnya kita set terlebih dahulu KeyboardType menjadi InlineKeyboard. Lalu kita set “EnabledPaging” menjadi true.

Disini kita akan menggunakan “HeadLayoutButtonRow” untuk menampilkan button “Back” dan “Switch“. Di ikuti dengan membuat button pada “SubHeadLayoutButtonRow“. Dimana button ini digunakan untuk menampilkan jumlah CheckBox yang dipilih. Pada latihan ini, kita akan membuat 30 checkbox dimana pada setiap page akan menampilkan 9 item. Untuk mendapatkan jumlah item yang dipilih kita perlu membuat method/event seperti dibawah.

```
private async Task M_Buttons_CheckedChanged(object sender, CheckedChangedEventArgs e)
{
    m_Buttons.SubHeadLayoutButtonRow = new List<ButtonBase>()
    { new ButtonBase($"{m_Buttons.CheckedItems.Count} checked items", "$") };
}
```

Dengan menggunakan method “m_Buttons.CheckedItems.Count” kita akan mendapatkan nilai dari jumlah checkbox yang telah dipilih.

Lalu kita buat event untuk ButtonClicked (detail sintaks dapat dilihat pada project lampiran).

Lalu pada constructor ketikkan sintaks dibawah.

```
public CheckedButtonListForm()
{
    this.DeleteMode = TelegramBotBase.Enums.EDeleteMode.OnLeavingForm;
    this.Init += CheckedButtonListForm_Init;
}
```

Jalankan program dan pastikan mendapatkan hasil seperti dibawah.



Notification

Notifikasi dalam kerangka kerja bot Telegram merujuk pada pesan atau pemberitahuan yang dikirim oleh bot kepada pengguna atau ke grup obrolan sebagai respons terhadap suatu peristiwa atau perintah yang telah terjadi.

Bot Telegram dapat mengirim notifikasi untuk berbagai keperluan, seperti:

Pemberitahuan Responsif: Bot memberi balasan kepada pengguna atas perintah yang diberikan. Ini bisa berupa konfirmasi bahwa perintah telah diterima dan sedang diproses.

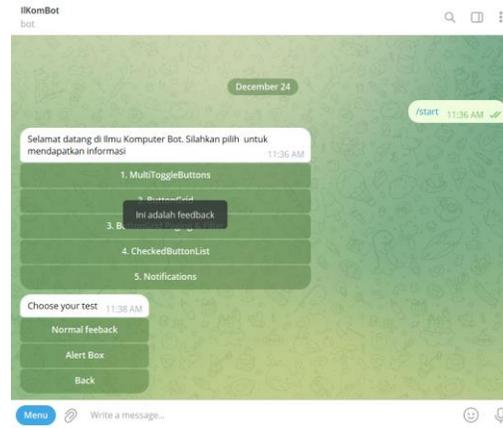
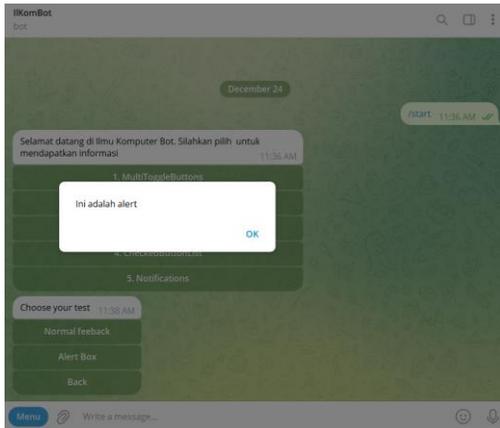
Notifikasi Kejadian Penting: Ketika ada peristiwa penting yang terjadi dalam konteks tertentu, bot dapat memberikan notifikasi. Contohnya, notifikasi untuk pengguna ketika ada pembaruan dalam layanan, informasi penting, atau perubahan status tertentu.

Pemberitahuan Rutin: Bot juga bisa mengirimkan pemberitahuan rutin seperti pengingat jadwal, informasi terkini, atau pembaruan berkala sesuai dengan fungsi bot tersebut.

Notifikasi ini bisa dalam bentuk pesan teks, media, tautan, atau informasi lainnya yang relevan dengan fungsi bot dan interaksi yang telah dilakukan pengguna. Dengan notifikasi, bot dapat memberikan informasi atau tanggapan yang lebih interaktif dan

langsung kepada pengguna, memperkaya pengalaman pengguna dalam berinteraksi dengan bot tersebut.

Sebagai latihan ikuti sintaks yang ada pada project lampiran. Dan pastikan mendapatkan hasil seperti pada gambar dibawah ini.

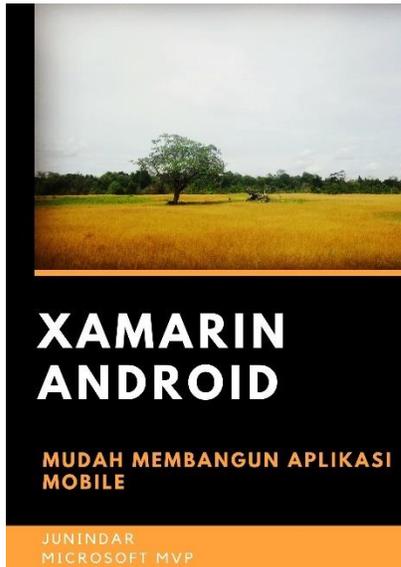


Penutup

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/2024/01/telegram-bot-framework-custom-control.html>

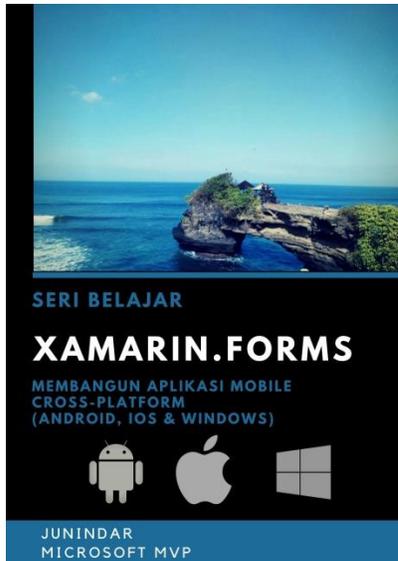
Referensi



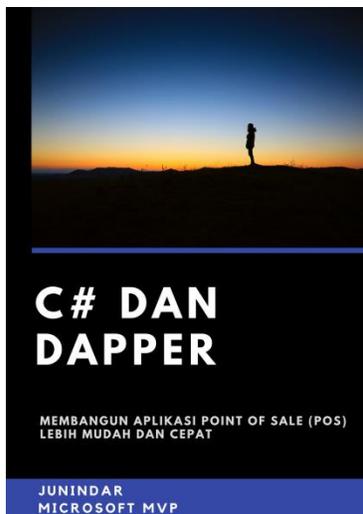
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



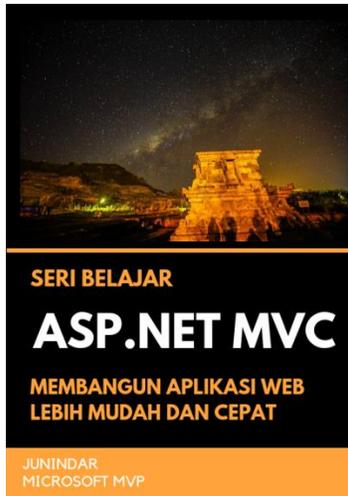
<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>



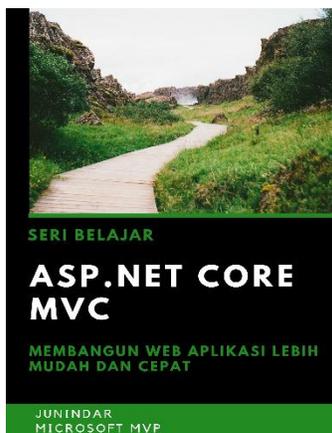
https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ



[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_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)

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”**.