

Telegram.Bot Pada C# : Send Message

Part I

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

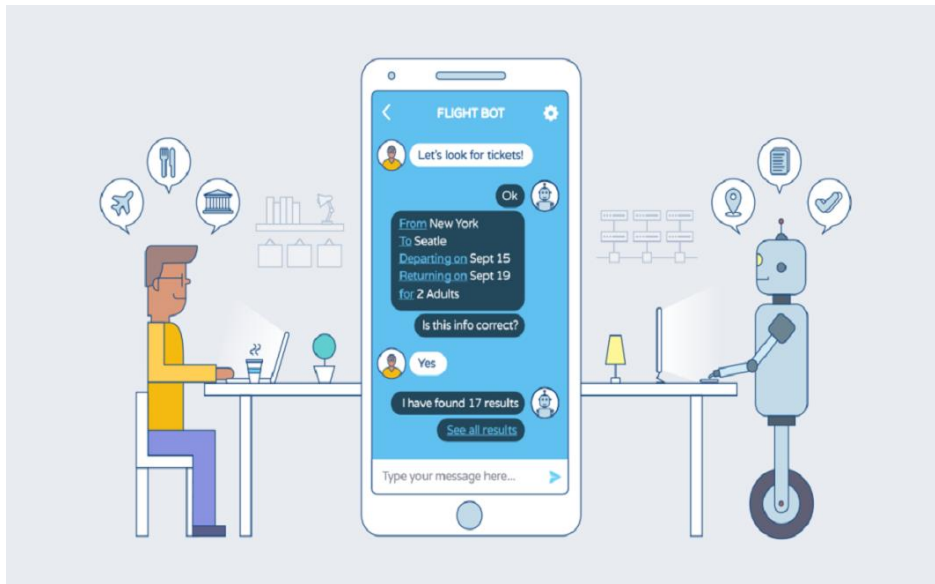
Chatbot atau chatterbot adalah sebuah layanan obrolan robot/tokoh virtual dengan kecerdasan buatan atau AI (Artificial Intelligent) yang menirukan percakapan manusia melalui pesan suara, obrolan teks ataupun keduanya. Chatbot menggunakan berbagai teknologi yang mutakhir di dalamnya, seperti Artificial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

Telegram.Bot

Junindar, ST, MCPD, MOS, MCT, MVP .NET

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.

Pada latihan sebelumnya kita telah membuat aplikasi chatbot sederhana menggunakan Telegram.Bot library. Dimana kita sudah bisa melakukan percakapan dengan bot tersebut. Pada artikel ini akan dibahas bagaimana mengirimkan pesan teks, photo, sticker dan audio pada chatbot.

Untuk memudahkan dalam memahami artikel ini, disarankan untuk membaca artikel sebelumnya.

Text

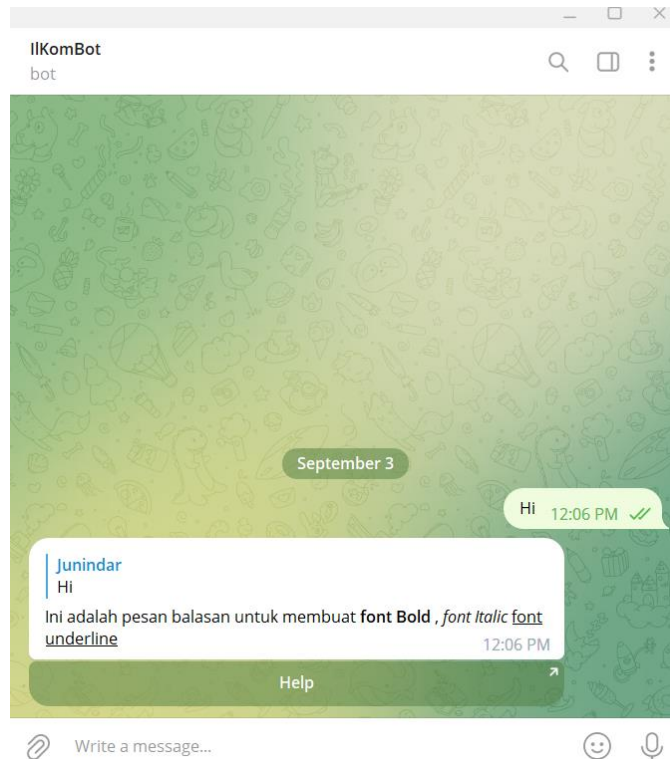
Untuk mengirim pesan teks kita gunakan method “SendMessage“ dari BotClient. Dan mungkin merupakan method dari API Bot Telegram yang paling banyak digunakan.

Pesan teks mudah dikirim dan cepat ditampilkan pada perangkat dengan jaringan yang lebih lambat. Pada telegram juga memungkinkan kita untuk menggunakan format Markdown atau HTML.

Berikut adalah sintaks untuk mengirimkan pesan dengan menggunakan Markdown,

```
Message sendMessage = await botClient.SendTextMessageAsync(  
    chatId: chatId,  
    text: "Ini adalah pesan balasan untuk membuat font Bold , font Italic font underline ",  
    parseMode: ParseMode.MarkdownV2,  
    disableNotification: true,  
    replyToMessageId: update.Message.MessageId,  
    replyMarkup: new InlineKeyboardMarkup(  
        InlineKeyboardButton.WithUrl(  
            text: "Help",  
            url: "https://core.telegram.org/bots/api#sendmessage")),  
    cancellationTokens: cancellationToken);
```

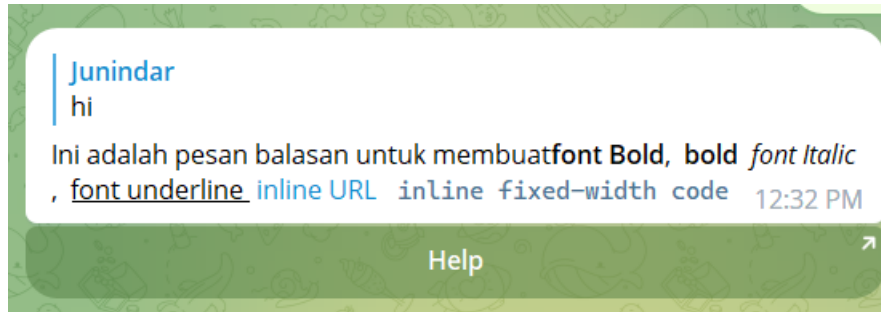
Pada sintaks diatas, kita gunakan beberapa parameter seperti disableNotification, replyToMessageId dan replyMarkup. Untuk replyMarkup kita gunakan InlineKeyboardMarkup, sehingga nantinya pada pesan yang dikirimkan terdapat sebuah button yang jika ditekan akan membuka link yang telah kita set pada code.



Untuk detail penggunaan Markdown dan HTML dapat dilihat pada link berikut <https://core.telegram.org/bots/api#markdownv2-style> .

Untuk menggunakan HTML, kita ganti “MarkdownV2” pada ParseMode menjadi HTML. Berikut adalah contoh pesan yang menggunakan HTML

```
string texto = @"Ini adalah pesan balasan untuk membuat <b>font Bold</b>, <strong>  
bold </strong>," +  
@"<i> font Italic </i>, <u> font underline </u>" +  
@"<a href = 'http://www.example.com/'> inline URL </a>";
```

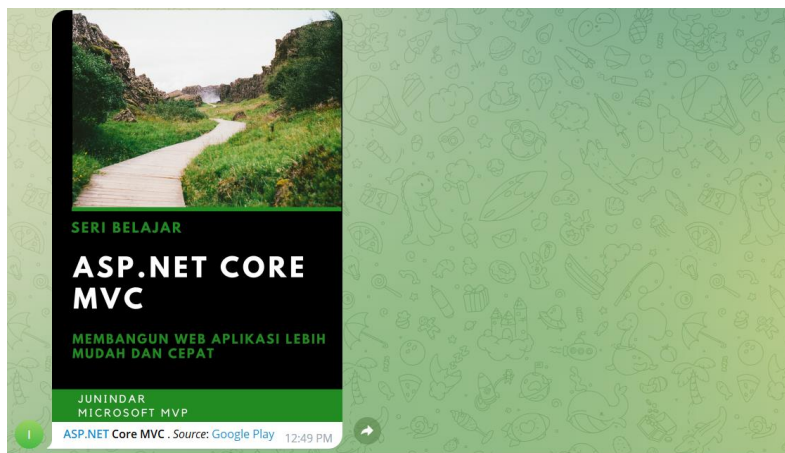


Photo

Untuk mengirim pesan teks kita gunakan method “SendPhoto” dari BotClient. Terdapat beberapa cara untuk mengirim photo, yaitu dengan menggunakan Uri, Stream dan FileId.

Berikut contoh sintaks untuk mengirimkan photo menggunakan Uri.

```
Message sentMessage = await botClient.SendPhotoAsync(  
chatId: chatId,  
photo: InputFile.FromUri("https://raw.githubusercontent.com/junindar/ilkom/master/  
TelegramBot/Images/AspNetCore.png"),  
caption: "<b>ASP .NET Core MVC </b>. <i>Source</i>: <a  
href=\"https://play.google.com/store/books/details/Junindar_ASP_NET_Core_MVC?id=CIG9  
DwAAQBAJ&hl=en&gl=US\">Google Play</a>",  
parseMode: ParseMode.Html,  
cancellationTokens: cancellationTokens);
```



Selain mengirimkan photo, kita dapat juga menambahkan Caption pada photo tersebut, dengan menggunakan parameter Caption. Untuk Caption kita juga dapat menggunakan HTML pada ParseMode nya.

Latihan berikutnya adalah bagaimana mengirimkan photo dari local drive. Untuk mengirimkan photo dari local drive, sebelumnya kita convert terlebih dahulu file tersebut menjadi stream. Berikut sintaks detailnya.

```
FileStream fStream = new FileStream(@"E:\Program\Ilkom\Artikel  
Program\ilkom\TelegramBot\Images\Blazor.png", FileMode.Open);  
  
Message sentMessage = await botClient.SendPhotoAsync(  
chatId: chatId,  
photo: InputFile.FromStream(fStream),  
caption: "<b>ASP .NET Core MVC </b>. <i>Source</i>: <a  
href=\"https://play.google.com/store/books/details/Junindar_Microsoft_Blazor_Membang  
un_Aplikasi_Web_D?id=HKZhEAAAQBAJ&hl=en&gl=US\">Google Play</a>\"",  
parseMode: ParseMode.Html,  
cancellationTokens: cancellationTokens);  
fStream.Close(); fStream.Dispose();
```



Write a message...

Pesan yang dikembalikan dari method “SendPhoto“ terdapat object array (PhotoSize). Nilainya adalah PhotoSize dengan setiap elemen mewakili photo yang sama dalam dimensi atau size berbeda. Sehingga jika kita perlu mengirim photo ini lagi, disarankan untuk menyimpan array ini sehingga Anda dapat menggunakan kembali nilai FileId. Berikut contohnya.

Buat dua buah class Photo dan BookCover seperti dibawah.

```
public class Photo  
{  
    public string FileId { get; set; }  
    public string FileName { get; set; }  
    public string FileUniqueId { get; set; }  
    public string Caption { get; set; }  
    public string Url { get; set; }  
}
```

```
public class BookCover
{
    public string FileName { get; set; }
    public string Path { get; set; }
    public string Caption { get; set; }
    public string Url { get; set; }
}
```

Lalu tambahkan dua buah variable yang menggunakan class-class diatas.

```
List<Photo> lstPhotos = new List<Photo>();
List<BookCover> lstCovers=new List<BookCover>();
```

Selanjutnya tambahkan sebuah function seperti dibawah. Untuk latihan ini kita gunakan static data. Dan nilai yang dikembalikan oleh function ini akan ditampung oleh variable “lstCovers”.

```
List<BookCover> getBookCovers()
{
    List<BookCover> bookCovers = new List<BookCover>
    {
        new()
        {
            Caption = "<b>ASP .NET Core MVC </b>. <i>Source</i>:",
            Path = @"E:\Program\Ilkom\Artikel
            Program\ilkom\TelegramBot\Images\AspNetCore.png",
            Url=
            "<a
            href=\"https://play.google.com/store/books/details/Junindar_ASP_NET_Core_MVC?id=
            CIG9DwAAQBAJ&hl=en&gl=US\">Google Play</a>"
            },
            new() {},
            new() {},
            new() {}
        };
    return bookCovers;
}
lstCovers = getBookCovers();
```

Dan pada method “HandleUpdateAsync” kita tambahkan sintaks seperti dibawah ini. Pada sintaks dibawah, kita buat sebuah logic untuk mengecek jumlah data pada “lstPhoto”. Jika nilainya adalah “0”, maka kita gunakan data pada “lstCovers” (static data) untuk mengirimkan photo. Dengan menggunakan “lstCovers” berarti gambar-gambar yang akan dikirimkan berasal dari local drive, sehingga kita menggunakan “FromStream”. Lalu setelah bot mengirimkan pesan photo, selanjutnya kita menggunakan nilai yang dikembalikan oleh method “SendPhotoAsync”. Lalu kita simpan data-data photo kedalam “lstPhoto”. Sehingga jika ada request lagi kita dapat menggunakan object ini dengan memanfaatkan “FileId”.

```
if (lstPhotos.Count ==0)
{
    foreach (var itm in lstCovers)
    {
        itm.FileName = Path.GetFileName(itm.Path);
        FileStream fStream = new FileStream(itm.Path, FileMode.Open);
        Message sendMessage = await botClient.SendPhotoAsync(
            chatId: chatId,

            photo: InputFile.FromStream(fStream),
            caption: $"{itm.Caption} {itm.Url}",
            parseMode: ParseMode.Html,
            cancellationTokens: cancellationTokens);
        fStream.Close(); fStream.Dispose();
        if (sendMessage.Photo != null)
        {
            var photo = sendMessage.Photo.First();
            lstPhotos.Add(new Photo
            {
                Caption = itm.Caption,
                FileName = itm.FileName,
                FileId = photo.FileId,
                Url = itm.Url,
                FileUniqueId = photo.FileUniqueId
            });
        }
    }
}
else
{
    foreach (var itm in lstPhotos)
    {
        await botClient.SendPhotoAsync(
            chatId: chatId,

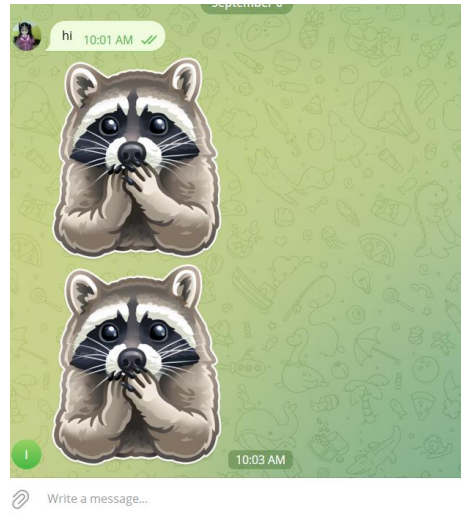
            photo: InputFile.FromFileId(itm.FileId),
            caption: $"{itm.Caption} {itm.Url}",
            parseMode: ParseMode.Html,
            cancellationTokens: cancellationTokens);
    }
}
```

Sticker

Untuk mengirimkan sticker pada telegram, kita harus menggunakan file dengan format WebP. Pada dasarnya untuk mengirimkan sticker hampir sama seperti photo. Kita dapat mengirimkan sticker menggunakan Uri, Stream maupun FileId. Tetapi pada sticker kita tidak dapat menambah caption seperti pada photo. Pada yang perlu diketahui pada photo kita gunakan parameter photo, sedangkan untuk sticker diganti dengan parameter sticker.

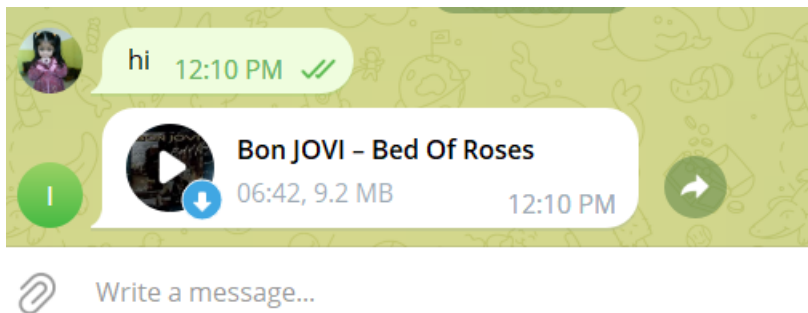
```
Message message1 = await botClient.SendStickerAsync(
    chatId: chatId,
    sticker: InputFile.FromUri("https://github.com/TelegramBots/book/raw/master/src/docs/sticker-
fred.webp"),
    cancellationTokens: cancellationTokens);

Message message2 = await botClient.SendStickerAsync(
    chatId: chatId,
    sticker: InputFile.FromFileId(message1.Sticker!.FileId),
    cancellationTokens: cancellationTokens);
```

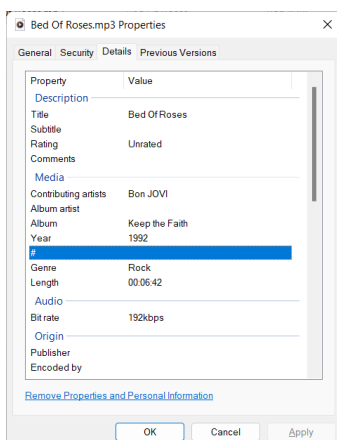


Audio & Voice Message

Kedua jenis pesan ini sangat mirip. Audio adalah file bertipe MP3 yang dapat diputar di pemutar musik. Sedangkan Voice memiliki format OGG dan tidak dapat dimainkan di pemutar musik.



Pada gambar diatas, audio yang dikirim bot disertai dengan nama artis maupun judul/title dari file tersebut.



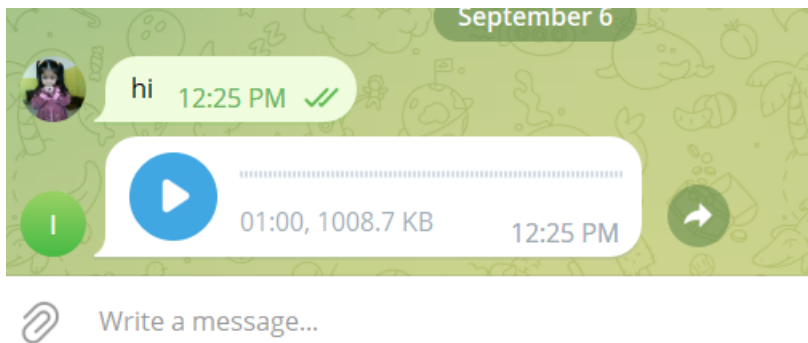
Jika pada file audio sudah memiliki property seperti berikut, maka secara otomatis nama artis maupun title akan ikut ditampilkan pada file tersebut. Tanpa harus kita menembahkannya pada coding.

```
FileStream fStream = new FileStream(@"..\audio\Bed Of Roses.mp3", FileMode.Open);  
FileStream fStreamThumb = new FileStream(@"..\Images\BonJovi.JPG", FileMode.Open);  
Message sentMessage = await botClient.SendAudioAsync(  
    chatId: chatId,  
    audio: InputFile.FromStream(fStream),  
    thumbnail: InputFile.FromStream(fStreamThumb),  
    //performer: "Bon Jovi",  
    //title: "Bed Of Roses",  
    cancellationToken: cancellationToken);  
fStream.Close(); fStream.Dispose();  
fStreamThumb.Close(); fStreamThumb.Dispose();
```

Lihat pada sintaks diatas, untuk mengirimkan audi kita gunakan method “SendAudio“. Pada method tersebut juga terdapat parameter “performer“ dan “title“, sehingga jika pada file yang akan dikirimkan tidak memiliki property tersebut kita dapat menembahkannya dengan menggunakan dua parameter tersebut. Lalu kita juga dapat menggunakan parameter “thumbnail“, untuk menambahkan gambar pada file. Seperti pada contoh gambar diatas.

Untuk voice message menggunakan file audio ber-extension “OGG“. Berikut contoh sintaks untuk mengirimkan file voice message. Tidak jauh berbeda disini kita menggunakan method “SendVoice“. Pada method ini tidak terdapat paramater title, performer maupun thumbnail.

```
FileStream fStream = new FileStream(@"\Voice\sample1.ogg", FileMode.Open);  
Message sentMessage = await botClient.SendVoiceAsync(  
    chatId: chatId,  
    voice: InputFile.FromStream(fStream),  
    cancellationToken: cancellationToken);  
fStream.Close(); fStream.Dispose();
```

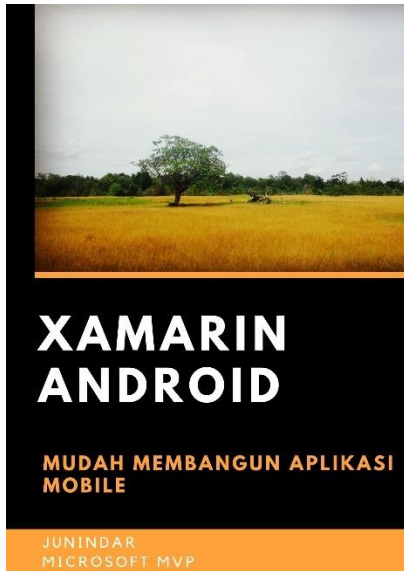


Penutup

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/2023/09/telegrambot-pada-c-send-message-part-i.html>

Referensi



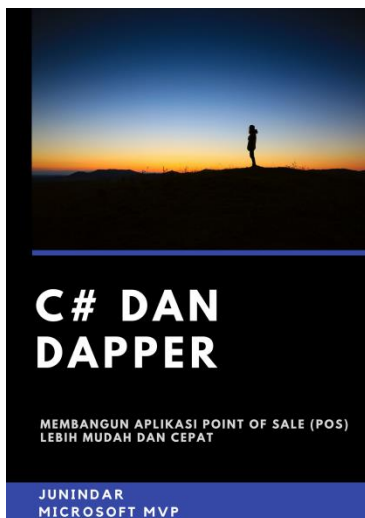
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



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



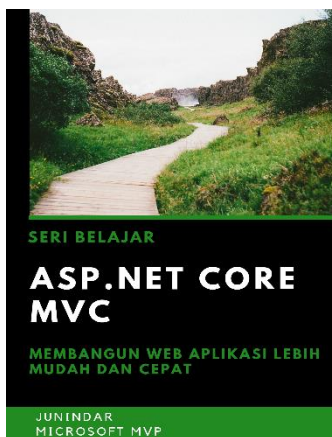
https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ



[https://play.google.com/store/books/details/Junindar_C dan Dapper Membangun Aplikasi POS P
oint?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”**.