

Telegram Bot Framework dan Azure Open AI

Junindar, ST, MCPD, MOS, MCT, MVP

junindar@gmail.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.

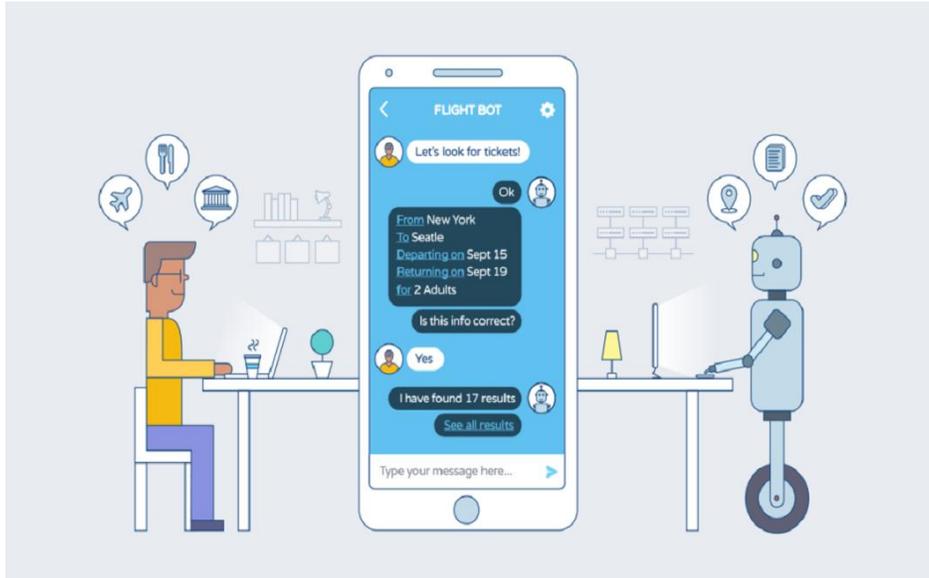
<http://junindar.blogspot.com>

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.

Azure OpenAI Services adalah sejumlah layanan yang disediakan oleh Microsoft Azure yang memungkinkan pengembang untuk mengintegrasikan teknologi kecerdasan buatan yang dikembangkan oleh OpenAI ke dalam aplikasi mereka. Ini mencakup layanan seperti Azure Cognitive Services, yang menyediakan berbagai fitur AI dan Machine Learning, serta integrasi khusus dengan model-model canggih yang dikembangkan oleh OpenAI untuk tugas-tugas seperti pemrosesan bahasa alami, pengenalan gambar, dan lainnya.

Dengan menggunakan Azure OpenAI Services, pengembang dapat dengan mudah memanfaatkan kemampuan AI dan Machine Learning tanpa perlu mengembangkan model mereka sendiri dari awal. Ini mempercepat proses pengembangan aplikasi yang cerdas dan membantu dalam menciptakan solusi yang lebih canggih untuk berbagai macam masalah bisnis dan teknis.

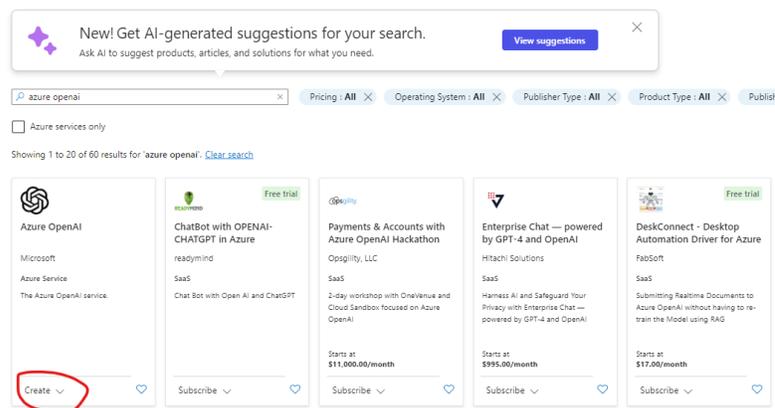
Azure OpenAI Service menyediakan akses REST API ke model OpenAI yang *powerfull* termasuk seri model GPT-4, GPT-4 Turbo dengan Vision, GPT-3.5-Turbo. Selain itu, seri model GPT-4 dan GPT-3.5-Turbo baru kini telah tersedia secara umum. Model ini dapat dengan mudah disesuaikan dengan tugas spesifik namun tidak terbatas pada pembuatan konten, ringkasan, pemahaman gambar, pencarian semantik, dan terjemahan natural language ke kode. Pengguna dapat mengakses layanan melalui REST API, Python SDK, atau antarmuka berbasis web kami di Azure OpenAI Studio.

Lalu bagaimana untuk mendapatkan akses ke Azure Open AI? Untuk mendapatkan akses ke Azure Open AI saat ini terbatas. Kita harus melakukan request terlebih dahulu melalui tautan berikut : <https://aka.ms/oaiapply>

The image shows a multi-step application form on the left and a preview of the Azure portal on the right. The form has three steps: 1. 'Your First Name *' with an input field. 2. 'Your Last Name *' with an input field. 3. 'How many Azure Subscription ID's would you like to have access to this service? *' with radio button options for 1, 2, and 3. The preview on the right shows a message: 'Please provide your 1st Azure Subscription ID (not your Tenant ID) Subscription must belong to your organization. If you do not enter a valid Subscription ID, your application will be DENIED. Don't know your Azure Subscription ID? See article for help: https://aka.ms/get-subscription-id *'. Below this is a 'Create a resource' button and a 'Subscriptions' button. The preview also shows a screenshot of the Azure portal 'Subscriptions' page with a table of subscriptions and a 'Create a resource' button at the bottom.

Dan pastikan kita telah memiliki Azure Account. Jika belum memiliki Azure Account, kita dapat membuatnya terlebih dahulu disini <https://azure.microsoft.com/en-us/free/> . Setelah selesai dengan langkah-langkah diatas, kita lanjutkan dengan membuat Azure OpenAI Service Resource.

- Login pada Azure Portal : <https://portal.azure.com/>
- Pilih “Create a resource“ dan cari “Azure OpenAI“.



- Lalu klik “Create“ pada Azure OpenAI yang ditampilkan pada hasil pencarian.

The screenshot shows the 'Create Azure OpenAI' page in the Microsoft Azure portal. The 'Basics' tab is selected. The page contains the following sections and fields:

- Basics**: Includes a description of OpenAI's capabilities and a 'Review + create' button.
- Project details**: Includes a 'View automation template' link and instructions to select a subscription and resource group.
- Instance details**: Includes dropdown menus for 'Region' (East US), 'Name', and 'Pricing tier' (Standard S0).

Pada halaman “Create Azure OpenAI“ tab Basic, masukkan informasi yang diperlukan.

- Lalu klik “Review+create“ button.

The screenshot shows the 'Review + create' tab of the 'Create Azure OpenAI' page. It includes the following sections:

- TERMS**: A section with legal terms and conditions.
- Basics**: A summary table of the configuration details.

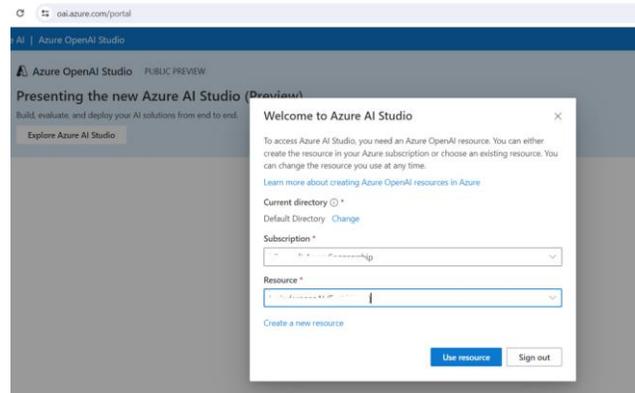
Basics	
Subscription	[Subscription ID]
Resource group	[Resource Group Name]
Region	East US
Name	[Name]
Pricing tier	Standard S0

- Jika semua informasi sudah benar klik button “Create“. Lalu tunggu hingga ada notifikasi jika resource yang dibuat telah selesai.

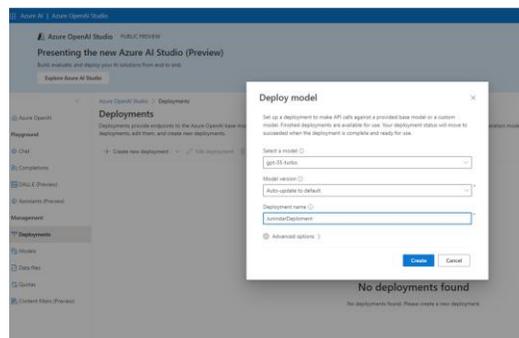
Setelah selesai membuat Resource seperti dengan langkah-langkah diatas, kita lanjutkan dengan melakukan Deploy Model pada Azure OpenAI Studio. Azure OpenAI Studio merupakan sebuah platform yang dikembangkan oleh Microsoft Azure yang memungkinkan pengguna untuk mengakses layanan dan teknologi kecerdasan buatan dari OpenAI. Ini memungkinkan pengembang dan peneliti untuk membuat, melatih, dan menerapkan model kecerdasan buatan menggunakan infrastruktur dan layanan cloud

yang disediakan oleh Microsoft Azure, sambil memanfaatkan kemampuan AI yang dikembangkan oleh OpenAI.

- Buka halaman Azure OpenAI Studio (<https://oai.azure.com/>).



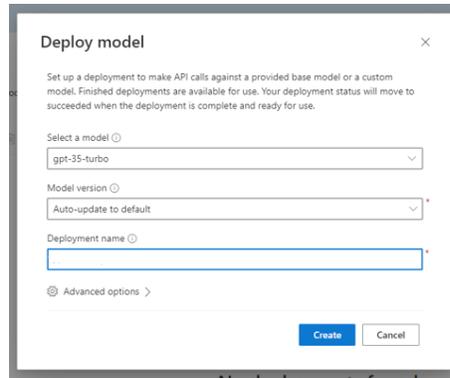
- Lalu pilih Subscription dan Azure OpenAI resource yang telah kita buat sebelumnya. Dan klik button “Use resource“.



- Lalu pilih Deployment pada section Management



- Selanjutnya pilih “Create new deployment“. Dan masukkan data-data seperti pada gambar dibawah.

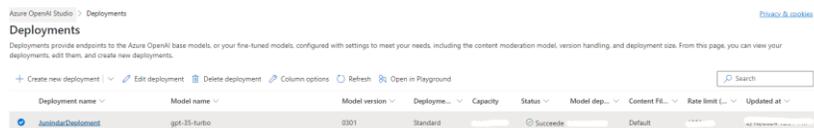


Model : Ketersediaan model bervariasi berdasarkan Region.

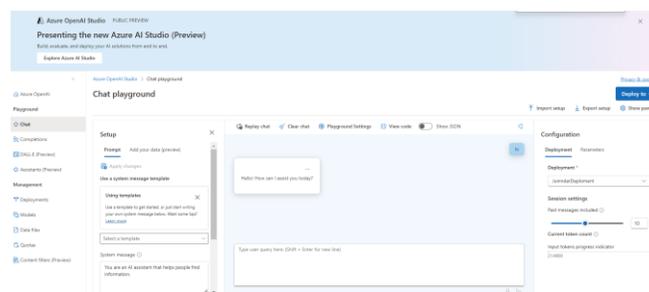
Deployment Name : Nama deployment yang akan digunakan pada code untuk memanggil Model dengan menggunakan Client Library atau REST API.

Pada latihan ini, untuk “Advanced options“ pilih default.

- Klik “Create“ button untuk Deploy Model yang dibuat. Jika berhasil maka pada halaman Deployment akan muncul model yang baru saja kita deploy.



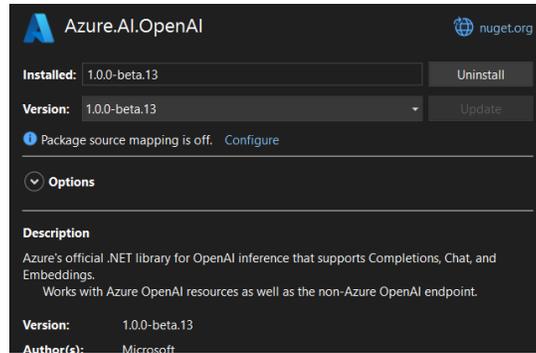
Untuk melakukan percobaan, klik Chat pada section Playground. Disini kita bisa melakukan percobaan percakapan dengan ChatGPT, seperti pada gambar dibawah ini.



Setelah selesai dengan langkah-langkah diatas, kita lanjutkan dengan membuat sebuah aplikasi console untuk menggunakan Azure OpenAI.

Pada Visual Studio buat sebuah “Console App“ project dengan menggunakan .Net 8.0.

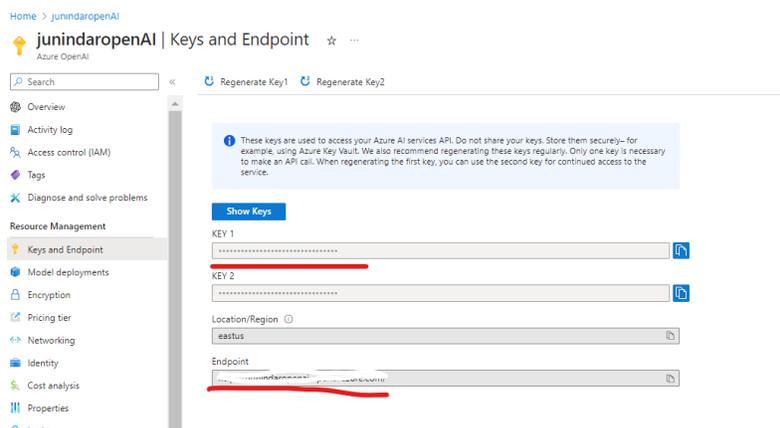
Setelah project berhasil dibuat, tambahkan nuget “ Azure.AI.OpenAI“ pada project.



Selanjutnya ketikkan sintaks seperti dibawah pada Program.cs.

```
using Azure;  
using Azure.AI.OpenAI;  
  
string endpoint = "endpoint"  
string key = "apiKey";  
var client = new OpenAIClient(new Uri(endpoint), new AzureKeyCredential(key));  
  
ChatCompletionsOptions options = new()  
{  
    DeploymentName = "NamaDeployment",  
};
```

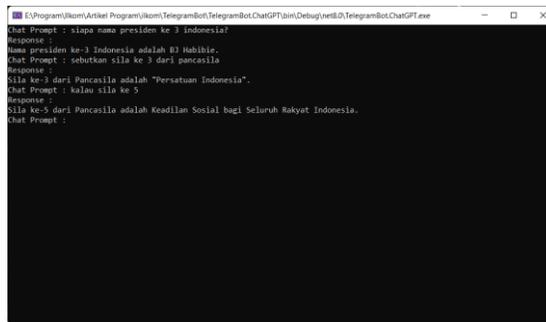
Buka halaman Resource yang telah kita buat diatas, lalu pada Section Resource Management, pilih Key and Endpoint untuk mendapatkan nilai dari EndPoint dan Key yang diperlukan pada code diatas.



Sedangkan untuk DeploymentName pada "ChatCompletionsOptions", kita gunakan nama deployment pada "Azure OpenAI Studio". Dan kita tambahkan sintaks seperti dibawah.

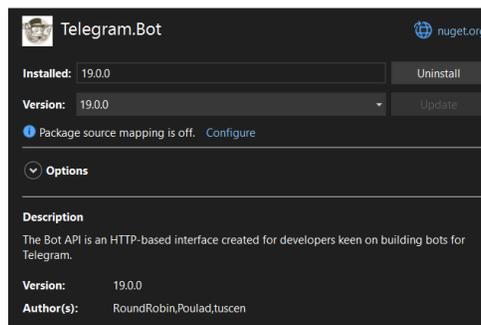
```
while (true)
{
    Console.WriteLine("Chat Prompt : ");
    string line = Console.ReadLine();
    if (line.Equals("exit", StringComparison.OrdinalIgnoreCase))
    {
        break;
    }
    options.Messages.Add(new ChatRequestUserMessage(line));
    Console.WriteLine("Response : ");
    Response<ChatCompletions> response =
    await client.GetChatCompletionsAsync( options);
    ChatCompletions completions = response.Value;
    string fullResponse = completions.Choices[0].Message.Content;
    Console.WriteLine(fullResponse);
}
```

Jika text yang dimasukkan sama dengan “exit”, maka aplikasi akan ditutup. Jika tidak, maka nilai dari text yang diketikkan akan digunakan untuk melakukan request dengan menggunakan object “ChatRequestUserMessage” dan ditambahkan kedalam IList Message. Lalu hasil dari method “GetChatCompletionsAsync”, akan ditampilkan pada layer. Seperti pada gambar dibawah ini.



Setelah kita berhasil dengan latihan diatas, maka akan kita lanjutkan dengan membuat latihan dengan menggunakan TelegramBot Library dan Azure OpenAI. Sehingga Telegram Bot yang kita buat dapat berinteraksi dengan ChatGPT.

Masih pada project yang sama tambahkan nuget Telegram.Bot pada project.



Tambahkan atau ganti sintaks pada Program.cs seperti dibawah. Untuk sintaks dari penggunaan TelegramBot, telah dijelaskan pada bagian sebelumnya.

```
string endpoint = "endPoint
string key = "apiKey";
var botClient = new TelegramBotClient("BotApiKey");
using CancellationAccessTokenSource cts = new();
var client = new OpenAIClient(new Uri(endpoint), new AzureKeyCredential(key));
ChatCompletionsOptions options = new()
{
    DeploymentName = "NamaDeployment",
};
ReceiverOptions receiverOptions = new()
{
    AllowedUpdates = Array.Empty<UpdateType>()
};
botClient.StartReceiving(
    updateHandler: HandleUpdateSendTextAsync,
    pollingErrorHandler: HandlePollingErrorAsync,
    receiverOptions: receiverOptions,
    cancellationToken: cts.Token
);
var me = await botClient.GetMeAsync();

Console.WriteLine($"Start listening for @{me.Username}");
Console.ReadLine();

cts.Cancel();
```

Untuk interaksi antara Azure OpenAI dan TelegramBot, kita gunakan method seperti dibawah ini. Dimana pesan yang didapat atau dikirimkan oleh pengguna, kita teruskan ke OpenAI.

```
async Task HandleUpdateSendTextAsync(ITelegramBotClient botClient, Update update,
CancellationAccessToken cancellationToken)
{
    if (update.Message is not { } message)
        return;
    if (message.Text is not { } messageText)
        return;
    var chatId = message.Chat.Id;
    Console.WriteLine($"Received a '{messageText}' message in chat {chatId}.");
    options.Messages.Add(new ChatRequestUserMessage(message.Text));
    Response<ChatCompletions> response =
    await client.GetChatCompletionsAsync(options);
    ChatCompletions completions = response.Value;
    string fullResponse = completions.Choices[0].Message.Content;
    await botClient.SendTextMessageAsync(
        chatId: chatId,
        text: fullResponse,
        disableNotification: true,
        replyToMessageId: update.Message.MessageId,
        cancellationToken: cancellationToken);
}
```

Dan terakhir adalah method untuk menangani jika terdapat error pada TelegramBot.

Task HandlePollingErrorAsync(ITelegramBotClient botClient, Exception exception, Cancellation token cancellationToken)

```
{  
    var ErrorMessage = exception switch  
    {  
        ApiRequestException apiRequestException  
        => $"Telegram API Error:\n[  
            {apiRequestException.ErrorCode}]\n{apiRequestException.Message}",  
        _ => exception.ToString()  
    };  
  
    Console.WriteLine(ErrorMessage);  
    return Task.CompletedTask;  
}
```

Jalankan program dan pastikan mendapatkan hasil seperti dibawah.

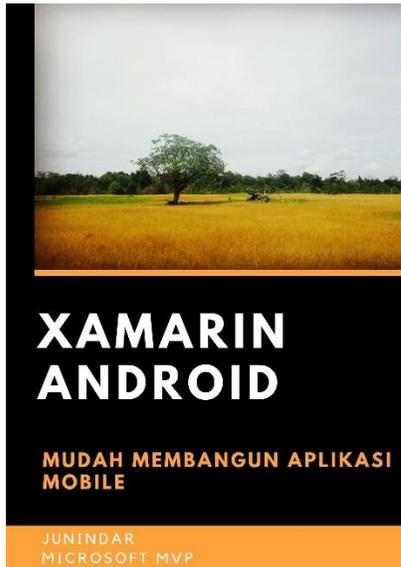


Penutup

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/2024/02/telegram-bot-framework-dan-azure-open-ai.html>

Referensi



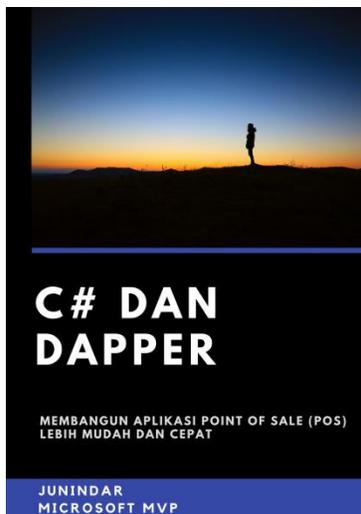
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



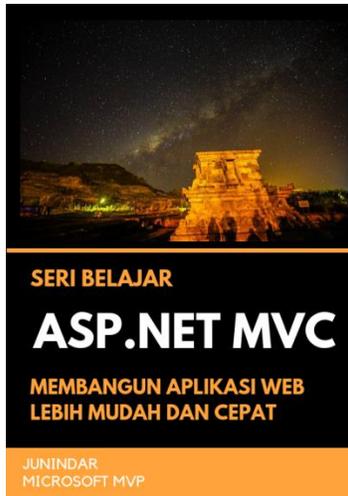
<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>



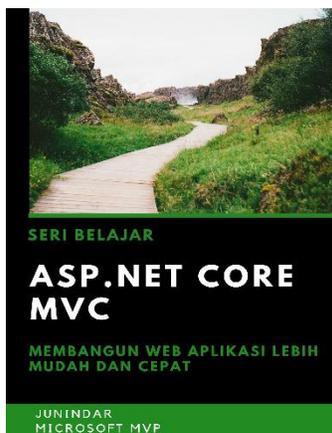
https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ



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_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”**.