

Pengenalan Microsoft Bot Framework

Junindar, ST, MCPD, MOS, MCT, MVP

junindar@gmail.com

<http://junindar.blogspot.com>

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

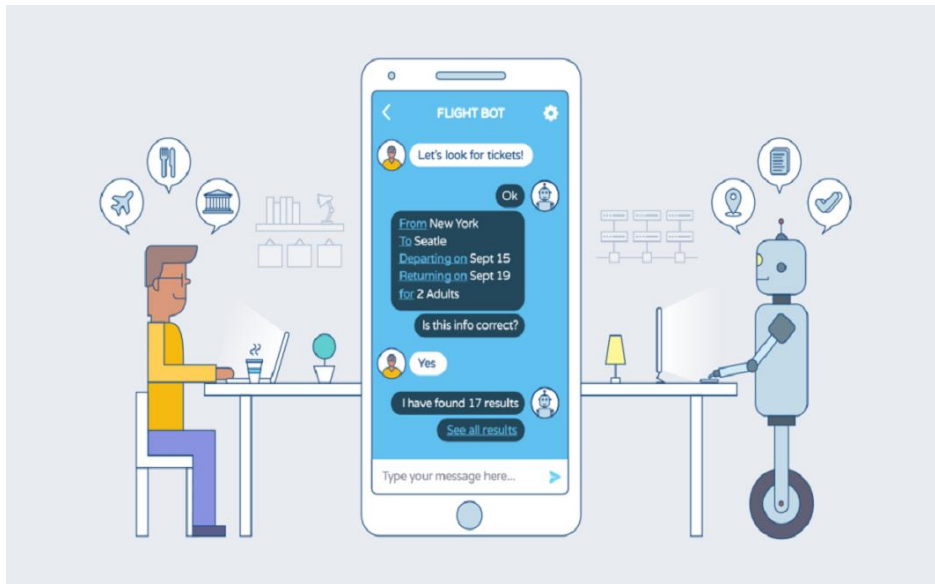
Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

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 Lenguage Processing (NLP). Machine Learning dapat di aplikasikan sebagai mesin yang mempelajari, menganalisa dan mengenal bermacam-macam Bahasa pada chatbot, sementara Natural Lenguage Processing (NLP) memiliki kemampuan untuk mengerti dan memahami Bahasa manusia lalu memberikan respon yang sesuai dengan Bahasa yang digunakan pengguna chatbot.

Pendahuluan

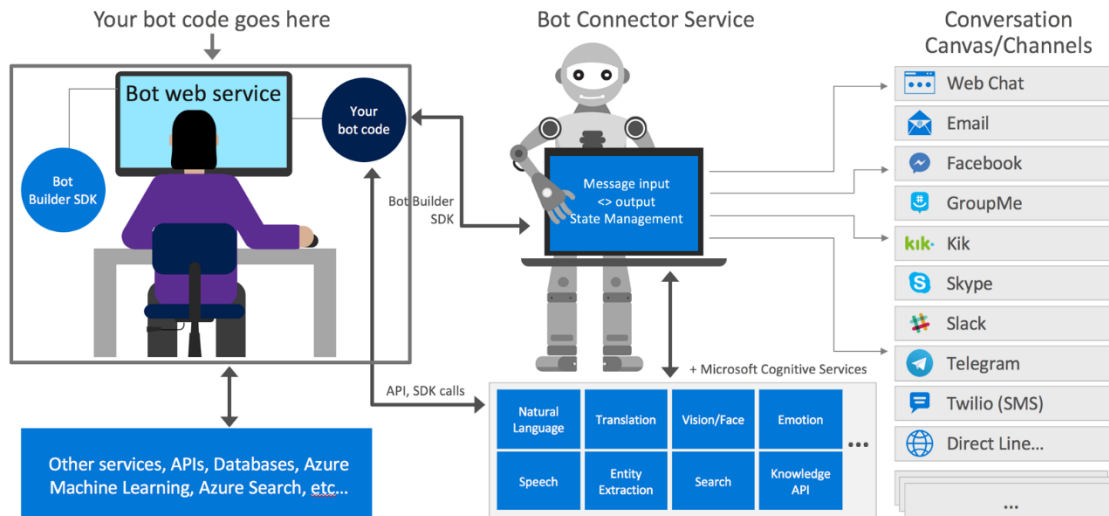
Chatbot atau chatterbot adalah sebuah layanan obrolan robot/tokoh virtual dengan kecerdasan buatan atau AI (Artificial Intelligent) yang menirukan percakapan manusia melalui pesan suara, obrolan teks ataupun keduanya.



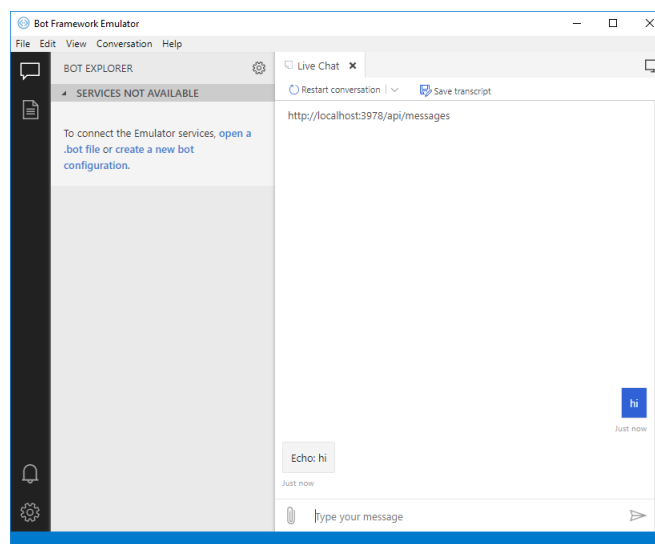
Pada dasarnya bots bekerja dengan cara melihat kata kunci dalam data yang masuk dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari basis data tekstual. Artinya, jika pengguna mengirim suatu permintaan maka bots akan membalasnya dengan respon yang spesifik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

Microsoft sebagai perusahaan Technology Information juga ikut dalam pengembangan technology chatbot ini yaitu dengan mengeluarkan beberapa produk yang mendukung technology ini, seperti Microsoft Bot Framework.

Microsoft Bot Framework adalah sebuah tools yang disediakan oleh Microsoft kepada para developer untuk membuat bot menjadi lebih mudah. tools ini support untuk C# maupun node.js dengan menggunakan REST (Representational state transfer)



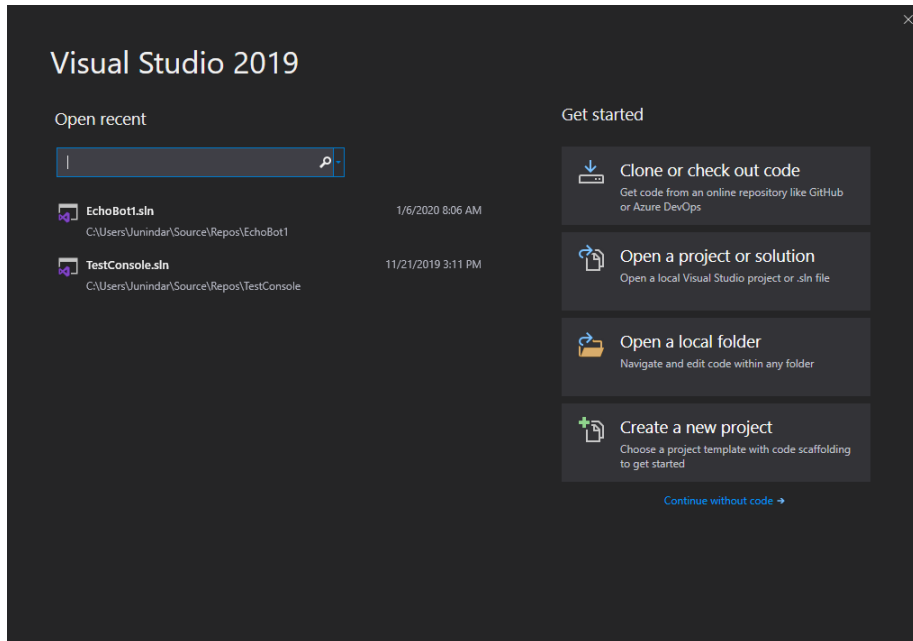
Sebelum kita memulai dengan menggunakan Microsoft Bot Framework, sebaiknya kita persiapkan terlebih dahulu tools yang diperlukan untuk membuat chatbot dengan Microsoft BotFramework. Pastikan pada PC/Laptop sudah memiliki Visual Studio 2017 atau versi di atasnya. Jika kondisi diatas sudah terpenuhi, selanjutnya kita download Bot Framework SDK Templates for VS disini <https://marketplace.visualstudio.com/items?itemName=BotBuilder.botbuilderv4> , lalu double click file vsix yang sudah didownload. Dan terakhir adalah dengan menginstall Bot Framework Emulator yang bisa didapatkan disini. Emulator digunakan pada saat proses development untuk melakukan pengujian code yang telah kita buat.



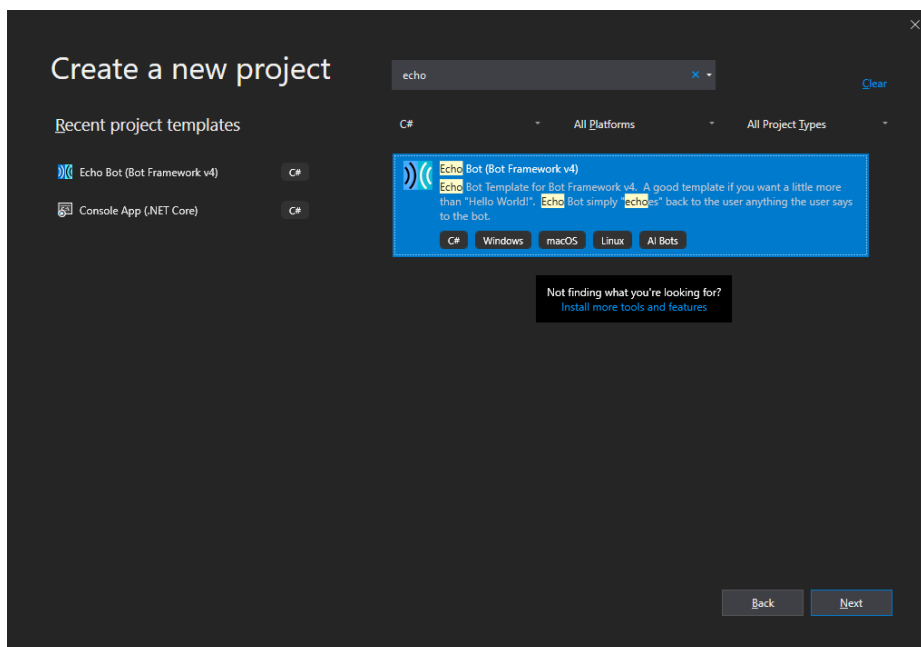
Selanjutnya untuk membuat chatbot dengan menggunakan Microsoft Bot Framework dapat mengikuti langkah-langkah dibawah ini.

Note : Pada artikel ini, penulis menggunakan Visual Studio 2019.

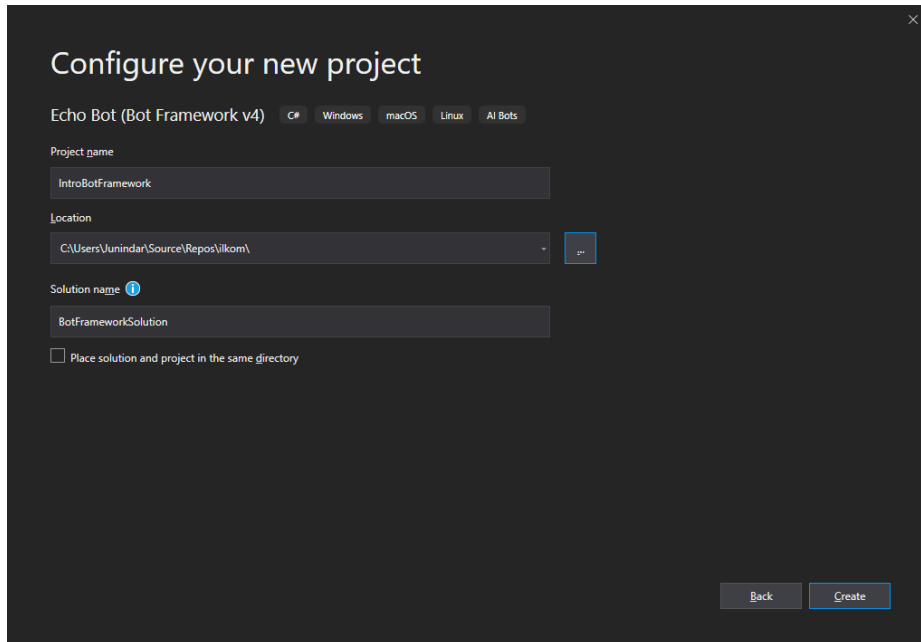
1. Buka Visual Studio 2019 dan klik “Create a new project“.



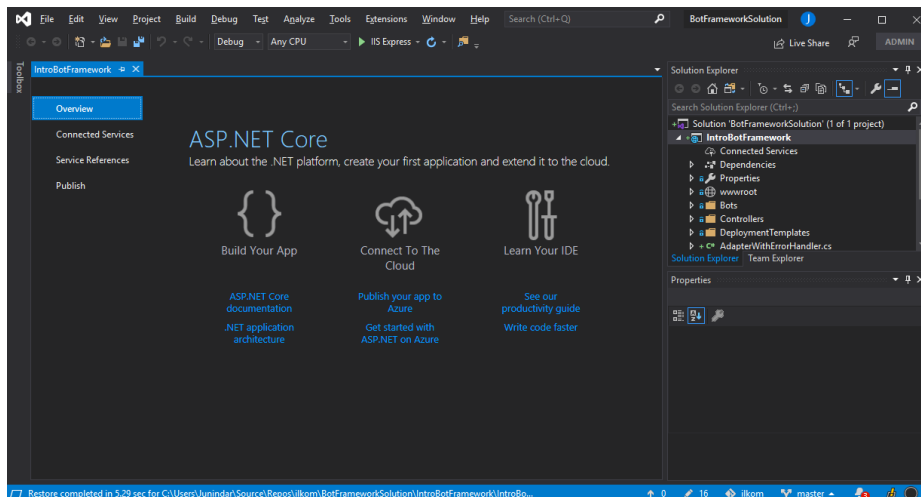
2. Selanjutnya pada dialog Create a new project, cari template „Echo Bot“ dan klik Next.



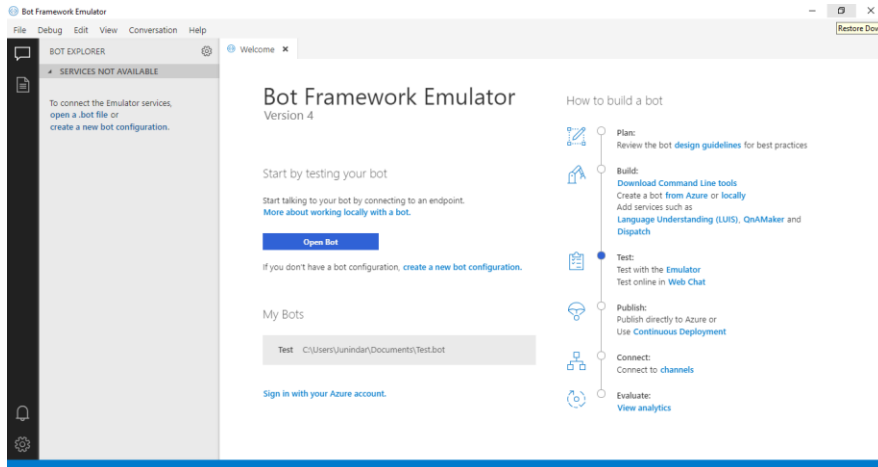
3. Pada dialog selanjutnya, isikan Project Name, Solution name dan lokasi penyimpanan project ini. Dan klik button Create



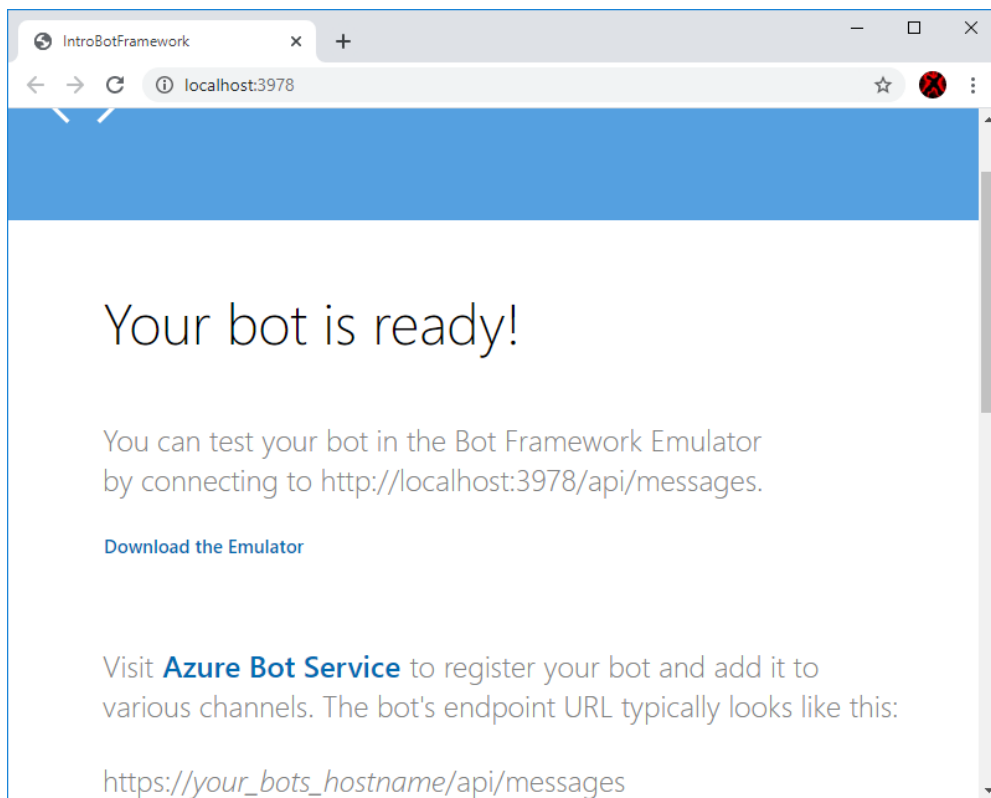
Setelah selesai dengan langkah-langkah diatas, maka project latihan kita akan dibuat seperti pada gambar dibawah.



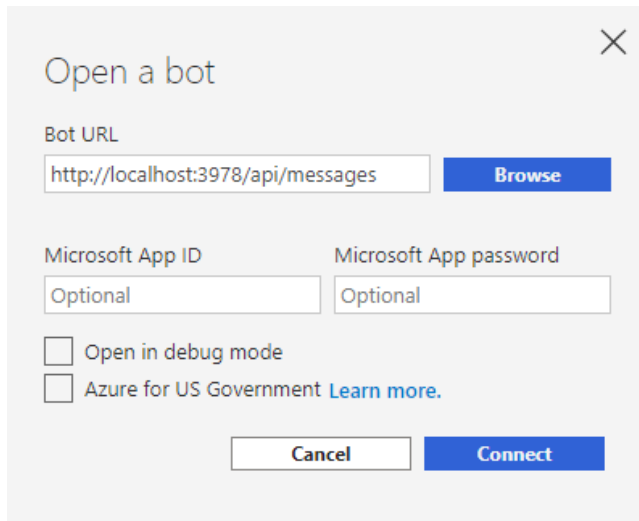
4. Untuk mencoba hasil dari project yang baru saja dibuat, buka Bot Framework Emulator.



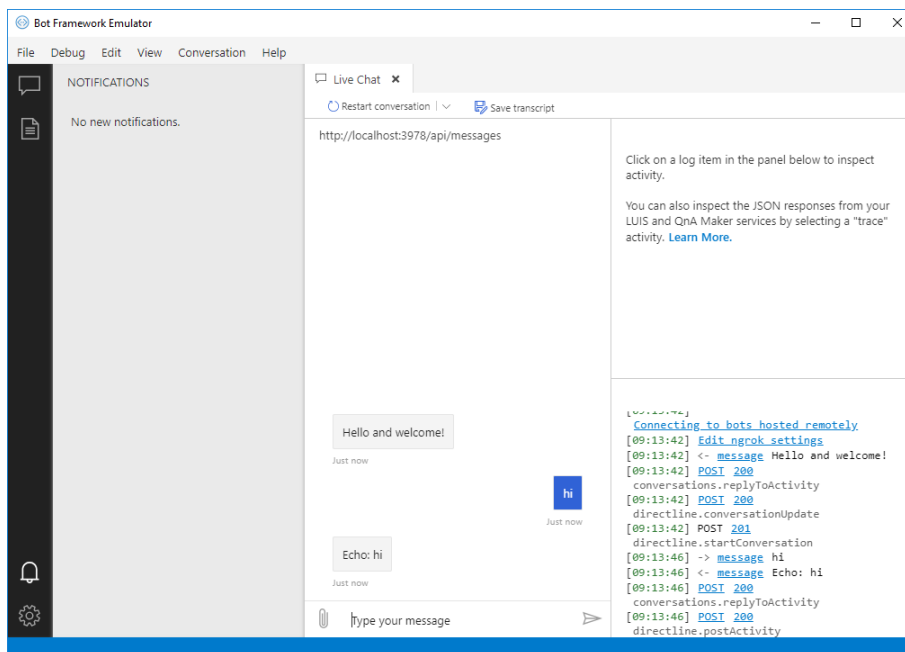
5. Lalu jalankan project (F5) pada visual studio dan pastikan mendapatkan hasil seperti gambar dibawah.



6. Kembali ke Emulator, klik button “Open a bot“. Pada jendela Open a bot ketikkan Url dan tambahkan “api/messages“ pada textbox Bot a url. Dan klik button Connect.

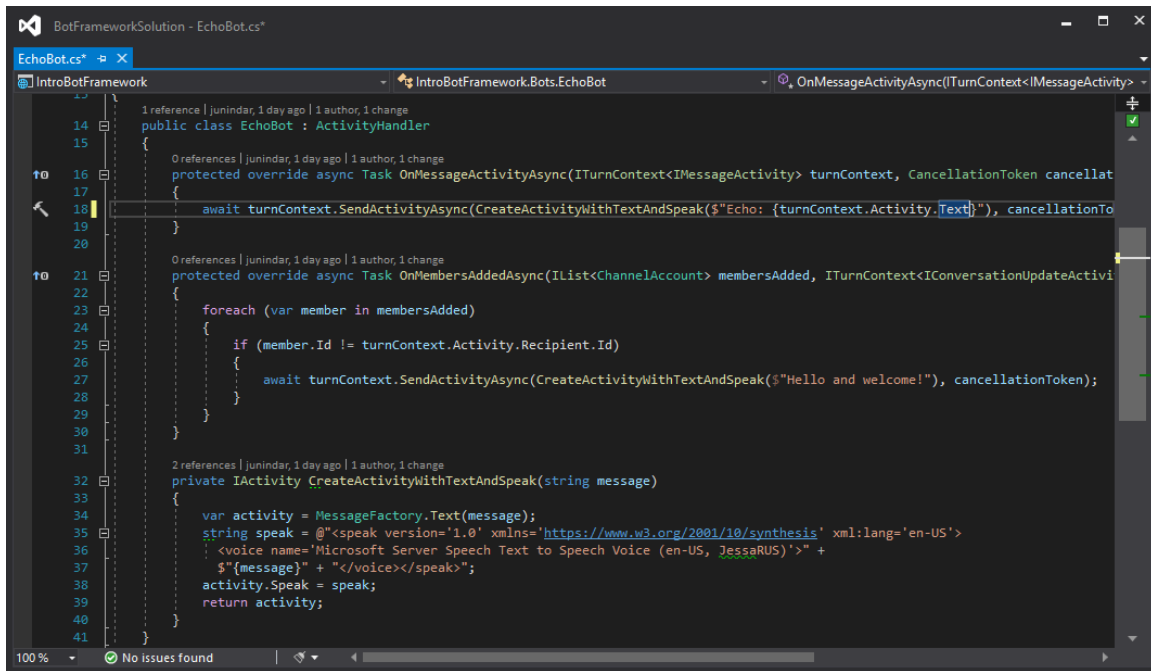


Pastikan mendapatkan hasil seperti gambar dibawah ini.



Lalu kembali ke Visual Studio dan stop aplikasi chatbot yang sebelumnya telah kita jalankan. Buka file “EchoBot.cs“ pada folder Bots. Pada file tersebut terdapat 3 method yaitu **OnMessageActivityAsync**, **OnMembersAddedAsync** dan **CreateActivityWithTextAndSpeak**. Method **OnMessageActivityAsync** berfungsi untuk interaksi pada user. **OnMembersAddedAsync** merupakan method yang digunakan ketika system mendeteksi terdapat pengguna baru pada chat. Sedangkan

CreateActivityWithTextAndSpeak merupakan method yang digunakan untuk mengkonversi text menjadi SSML (Speech Synthesis Markup Language).



```
1 reference | junindar, 1 day ago | 1 author, 1 change
public class EchoBot : ActivityHandler
{
    0 references | junindar, 1 day ago | 1 author, 1 change
    protected override async Task OnMessageActivityAsync(ITurnContext<IMessageActivity> turnContext, CancellationToken cancellat
    {
        await turnContext.SendActivityAsync(CreateActivityWithTextAndSpeak($"Echo: {turnContext.Activity.Text}"), cancellationTo
    }

    0 references | junindar, 1 day ago | 1 author, 1 change
    protected override async Task OnMembersAddedAsync(IList<ChannelAccount> membersAdded, ITurnContext<IConversationUpdateActivi
    {
        foreach (var member in membersAdded)
        {
            if (member.Id != turnContext.Activity.Recipient.Id)
            {
                await turnContext.SendActivityAsync(CreateActivityWithTextAndSpeak($"Hello and welcome!"), cancellationToken);
            }
        }
    }

    2 references | junindar, 1 day ago | 1 author, 1 change
    private IActivity CreateActivityWithTextAndSpeak(string message)
    {
        var activity = MessageFactory.Text(message);
        string speak = @"<speak version='1.0' xmlns='https://www.w3.org/2001/10/synthesis' xml:lang='en-US'>
        <voice name='Microsoft Server Speech Text to Speech Voice (en-US, JessaRUS)'> +
        $"{message}" + "</voice></speak>";
        activity.Speak = speak;
        return activity;
    }
}
```

Pada latihan ini, kita akan membuat chatbot sederhana yang akan menyimpan data dari user. Pertama-tama buat sebuah class dengan nama UserProfile dengan memiliki tiga buah property seperti dibawah ini. Property Nama dan Alamat adalah informasi yang akan kita simpan dimana informasi ini dimasukkan oleh pengguna. Sedangkan PromptedUserForName digunakan sebagai indikator.

```
public class UserProfile
{
    public string Nama { get; set; }
    public string Alamat { get; set; }
    public bool PromptedUserForName { get; set; } = false;
}
```

Lalu buka file Startup.cs dan tambahkan sintaks dibawah pada method **ConfigureServices**

```
services.AddSingleton<IStorage, MemoryStorage>();
services.AddSingleton<UserState>();
```

Sintaks diatas berfungsi untuk meregister MemoryStorage, dimana akan digunakan untuk membuat objek UserState.

Pada file EchoBot.cs, tambahkan sebuah field dan constructor seperti dibawah.


```
private BotState _userState;  
public EchoBot( UserState userState)  
{  
    _userState = userState;  
}
```

Untuk diketahui Class EchoBot mengimplemtasi dari IBot Interface, dan pada IBot interface terdapat method OnTurn(), method ini dipanggil untuk setiap aktivitas user maupun event seperti mengirim pesan, user sedang mengetik dan sebagai nya. Dan ketikkan sintaks berikut yang digunakan untuk menyimpan data kedalam storage.

```
public override async Task OnTurnAsync(ITurnContext turnContext, CancellationTokentoken = default(CancellationTokentoken))  
{  
    await base.OnTurnAsync(turnContext, cancellationToken);  
    await _userState.SaveChangesAsync(turnContext, false, cancellationToken);  
}
```

Sekarang kita akan mengganti sintaks pada method **OnMessageActivityAsync**, hapus terlebih dahulu sintaks yang ada. Selanjutnya kita akan membuat property accessor menggunakan method CreateProperty.

```
var userStateAccessors = _userState.CreateProperty<UserProfile>(nameof(UserProfile));
```

Dan untuk mendapatkan property dari storage (state cache) kita panggil method GetAsync seperti pada sintaks dibawah.

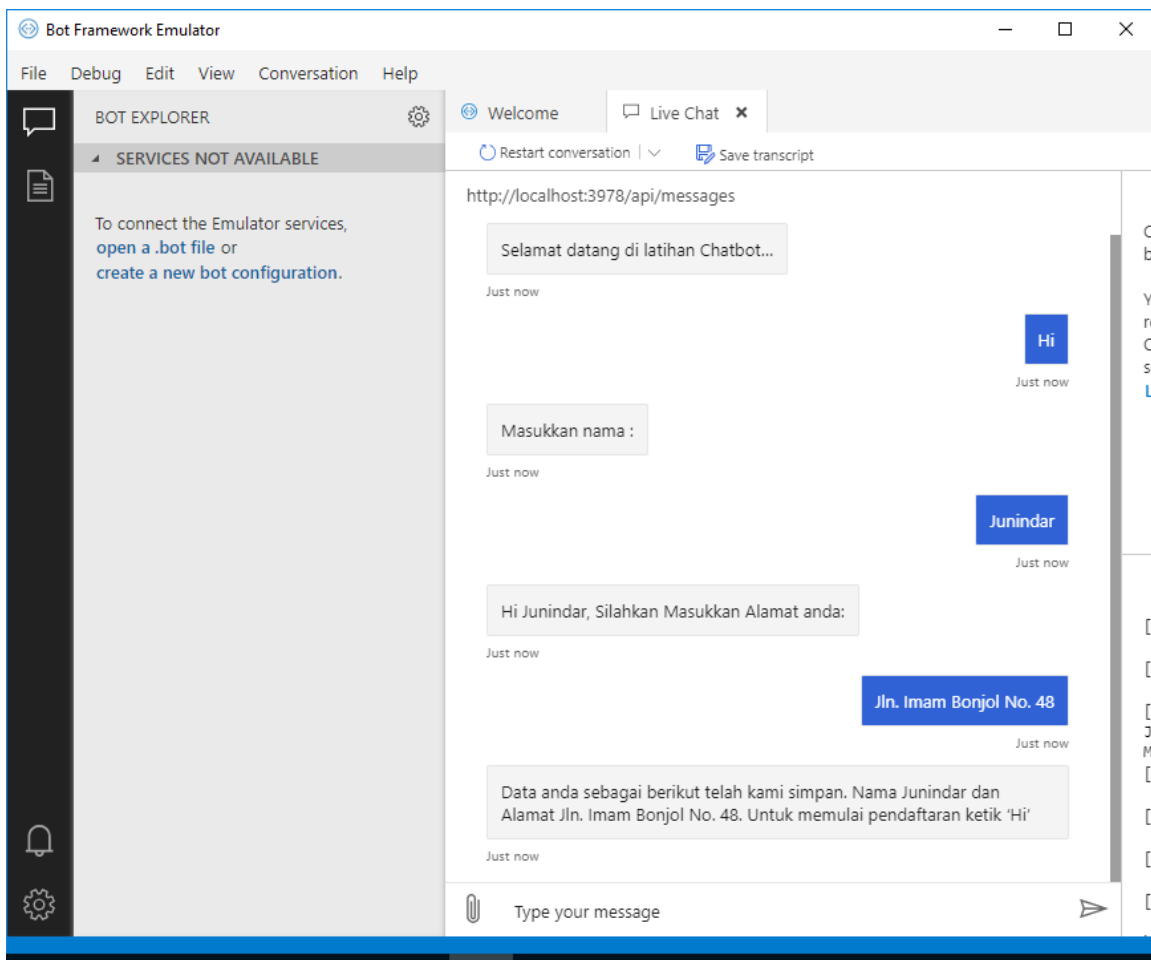
```
var userProfile = await userStateAccessors.GetAsync(turnContext, () => new UserProfile());
```

Selanjutnya kita tambahkan sintaks dibawa.

```
if (string.IsNullOrEmpty(userProfile>Nama))  
{  
    if (!userProfile.PromptedUserForName)  
    {  
        await turnContext.SendActivityAsync("Masukkan nama :");  
        userProfile.PromptedUserForName = true;  
    }  
    else  
    {  
        userProfile>Nama = turnContext.Activity.Text?.Trim();  
        await turnContext.SendActivityAsync($"Hi {userProfile>Nama}, Silahkan Masukkan Alamat anda:");  
    }  
}  
else  
{  
    userProfile.Alat = turnContext.Activity.Text?.Trim();  
    var textMsg =  
        $"Data anda sebagai berikut telah kami simpan. Nama {userProfile>Nama} dan Alamat {userProfile.Alat}. Untuk memulai pendaftaran ketik 'Hi';  
    userProfile>Nama = string.Empty;  
    userProfile.Alat = string.Empty;  
    userProfile.PromptedUserForName = false;  
    await turnContext.SendActivityAsync(textMsg);  
}
```

Jika `userProfile>Nama` sama dengan `empty`, selanjutnya akan dicek `userProfile.PromptedUserForName` bernilai `true` atau `false`. Jika bernilai `false`, maka akan keluar pertanyaan “Masukkan Nama” dan mengganti nilai `userProfile.PromptedUserForName` menjadi `true`. Setelah pengguna memasukkan Nama, maka data tersebut akan disimpan pada `userProfile>Nama`. Lalu mengirimkan pesan selanjutnya „“Hi {Nama, Silahkan masukkan nama anda :}““. Dan terakhir dengan mengirimkan pesan konfirmasi dari data-data yang pengguna masukkan.

Dan terakhir ganti text (Hello dan welcome !) pada method `OnMembersAddedAsync` menjadi “Selamat datang di latihan chatbot” dan jalankan program. Pastikan mendapatkan hasil seperti pada gambar dibawah.

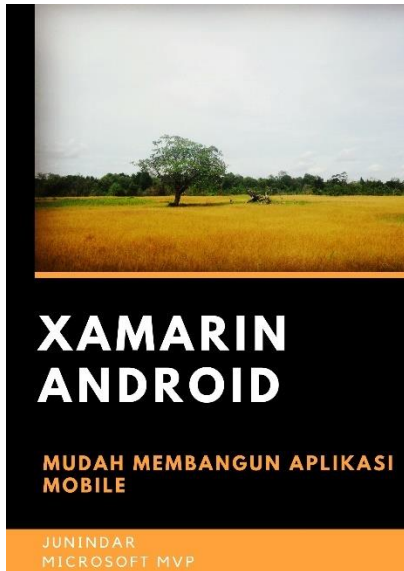


Penutup

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

<http://junindar.blogspot.com/2020/01/pengenalan-microsoft-bot-framework.html>

Referensi



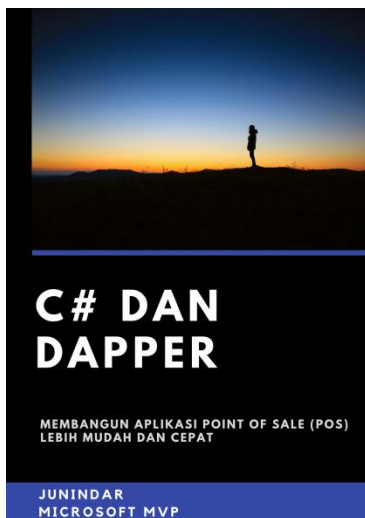
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



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



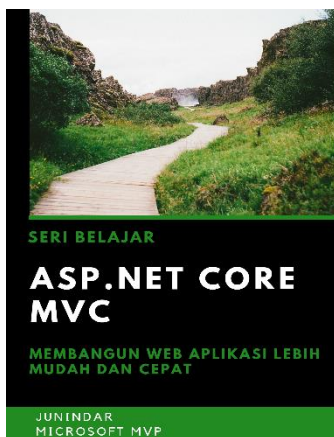
https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ



[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”**.