

Telegram Bot Framework Custom

Control 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

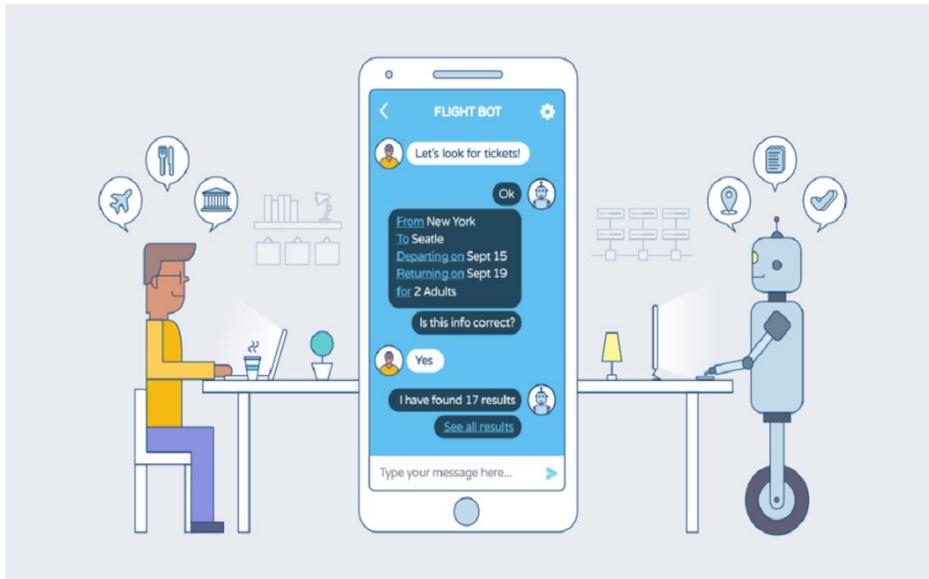
<http://junindar.blogspot.com>

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.

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.Bo 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 memiliki beberapa command pada menu dengan menggunakan Telegram Bot Framework. Untuk latihan ini kita lanjutkan dengan membuat chatbot dengan menggunakan control basic seperti button, maupun custom control. 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.

Pada Visual Studio buat sebuah “Console App“ project dengan menggunakan .Net 7.0. Setelah project berhasil dibuat, tambahkan “TelegranBotBase“ nuget pada project.

Lalu buat sebuah class dengan nama “Menu.cs“ seperti dibawah. Class ini akan digunakan sebagai StartForm (sintaks detail pada Program.cs) yang mana pada class ini akan terdapat menu dari control button.

```
public class Menu : AutoCleanForm
{
    public Menu()
    {
        this.DeleteMode = TelegramBotBase.Enums.EDeleteMode.None;
    }
}
```

Pada class ini menggunakan “AutoCleanForm“, dimana form ini akan menghapus pesan-pesan lama yang dikirim. Oleh karena itu pada constructor, nilai dari DeleteMode kita set menjadi None.

```
public override async Task Load(MessageResult message)
{
    if (message.MessageText.ToLower() != "/start") return;
    await Device.HideReplyKeyboard(".....");
}
```

Lalu pada method Load ketikkan sintaks diatas. Pada method ini akan dicek nilai dari MessageText. Jika nilainya sama dengan “/start“, maka bot akan membalas dengan menggunakan method “HideReplyKeyboard“, dimana method ini digunakan untuk membalas pesan dan nantinya akan dihapus kembali.

Selanjutnya pada method “Render“ kita tambahkan sintaks seperti dibawah.

```
public override async Task Render(MessageResult message)
{
    try
    {
        if (message.MessageText.ToLower() != "/start") return;

        ButtonForm btn = new ButtonForm();

        btn.AddButtonRow(new ButtonBase("1. Button",
            new CallbackData("a", "button").Serialize()));
        btn.AddButtonRow(new ButtonBase("2. Progress Bar",
            new CallbackData("a", "progress").Serialize()));
        btn.AddButtonRow(new ButtonBase("3. Calendar Picker",
            new CallbackData("a", "calendar").Serialize()));
        btn.AddButtonRow(new ButtonBase("4. Month Picker",
            new CallbackData("a", "month").Serialize()));
        btn.AddButtonRow(new ButtonBase("5. ToggleButtons",
            new CallbackData("a", "togglebuttons").Serialize()));
        await this.Device.Send("Selamat datang di Ilmu Komputer Bot.
            Silahkan pilih untuk mendapatkan informasi", btn);
    }
    catch (Exception ex)
    {
        await this.Device.Send("Terjadi kesalahan sistem, silahkan hubungi admin",
            null);
    }
}
```

Pada method ini kita akan membuat menu dengan menggunakan ButtonForm. Terdapat 5 buah button pada method ini. Dimana jika setiap button dipilih akan mengaktifkan form-form tersendiri. Oleh karena itu kita perlu membuat action pada setiap button. Jika kita lihat diatas terdapat class “CallbackData“ dengan dua paramater. Yang pertama adalah method sedangkan yang kedua adalah value. Dimana nilai dari parameter value inilah yang akan kita gunakan pada method Action yang akan kita buat setelah ini.

Ketikkan sintaks untuk method Action seperti dibawah ini. Pada method ini akan kita ambil nilai dari “CallbackData“ pada button yang dipilih. Lalu nilai“value“ dari “CallbackData“ akan kita gunakan pada switch case.

```
public override async Task Action(MessageResult message)
{
    var call = message.GetData<CallbackData>();
    await message.ConfirmAction();
    if (call == null) return;
    message.Handled = true;
    switch (call.Value)
    {
        case "button":
            //detail sintaks pada lampiran project
            break;
        case "progress":
            //detail sintaks pada lampiran project
            break;
        case "calendar":
            //detail sintaks pada lampiran project
            break;
        case "month":
            //detail sintaks pada lampiran project
            break;
        case "togglebuttons":
            //detail sintaks pada lampiran project
            break;
        default:
            message.Handled = false;
            break;
    }
}
```

Jalankan program dan pada menu pilih Start, pastikan mendapatkan hasil seperti dibawah ini.



Button Form

Tambahkan sebuah class dengan nama “ButtonTestForm“ dengan menggunakan “AutoCleanForm“ sebagai base class. Dan buat dua buah method “Action“ dan “Render“.

Pada dasarnya konsep yang digunakan sama dengan class Menu. Pada method “Render“ digunakan untuk membuat button dan untuk method “Action“ digunakan untuk menjalankan proses jika button dipilih. Untuk detail sintaks dapat dilihat pada project lampiran. Lalu kembali pada class Menu, dan tambahkan sintaks dibawah pada method “Action“ (`case "button":`). Dimana jika button 1 dipilih akan menampilkan atau mengaktifkan ButtonTestForm.

```
var bf = new ButtonTestForm();  
await this.NavigateTo(bf);
```



Progressbar

ProgressBar adalah sebuah control antarmuka grafis yang menampilkan visualisasi kemajuan atau progres suatu tugas atau proses. Biasanya, ProgressBar digunakan dalam aplikasi atau perangkat lunak untuk memberi informasi kepada pengguna tentang seberapa jauh suatu tugas telah selesai atau berapa persen proses telah dilakukan. Ini dapat berupa garis horizontal dengan penanda atau grafik yang bergerak untuk menunjukkan perkembangan dari awal hingga akhirnya suatu tugas. ProgressBar membantu pengguna untuk melihat dan memahami seberapa jauh suatu proses telah berjalan serta perkiraan waktu yang mungkin dibutuhkan untuk menyelesaikannya.

Untuk membuat progressbar dengan Telegram Bot Framework, kita gunakan class ProgressBar yang terdapat dalam library ini. Ikuti langkah-langkah dibawah ini :

Tambahkan sebuah class dengan nama “ProgressTestForm“ dengan menggunakan “AutoCleanForm“ sebagai base class. Lalu buat dua buah method seperti dibawah.

```
private async Task ProgressTest_Opened(object sender, EventArgs e)
{
    await this.Device.Send("Selamat datang di ProgressTest");
}
private async Task ProgressTest_Closed(object sender, EventArgs e)
{
    await this.Device.Send("Keluar dari ProgressTest");
}
```

Dua method diatas adalah akan digunakan untuk event Opened dan Closed. Lalu pada constructor ketikkan sintaks dibawah.

```
public ProgressTestForm()
{
    this.DeleteMode = TelegramBotBase.Enums.EDeleteMode.OnLeavingForm;
    this.Opened += ProgressTest_Opened;
    this.Closed += ProgressTest_Closed;
}
```

Selanjutnya buat method Render untuk membuat menu yang akan menampilkan jenis dari ProgressBar.

```
public override async Task Render(MessageResult message)
{
    ButtonForm btn = new ButtonForm();
    //detail sintaks di project lampiran
    await this.Device.Send("Pilih progress bar:", btn);
}
```

Dan terakhir tambahkan method Action seperti yang telah kita buat sebelumnya.

```
public override async Task Action(MessageResult message)
{
    var call = message.GetData<CallbackData>();
    await message.ConfirmAction();
    if (call == null) return;
    ProgressBar Bar = null;

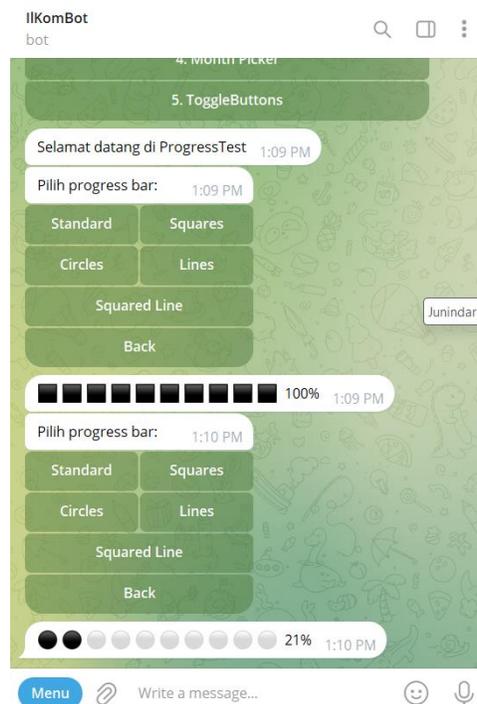
    switch (call.Value)
    {
        case "standard":
            //detail sintaks di project lampiran
            break;
        case "squares":
            //detail sintaks di project lampiran
            break;
        case "circles":
            //detail sintaks di project lampiran
            break;
        case "lines":
            //detail sintaks di project lampiran
            break;
        case "squaredlines":
            //detail sintaks di project lampiran
            break;
        case "start":
            var sf = new Menu();
            await this.NavigateTo(sf);
            return;
        default:
            return;
    }
}
```

```
//Render Progress bar
await Bar.Render(message);
this.Controls.Add(Bar);
for (int i = 0; i <= 100; i++)
{
    Bar.Value++;
    await Bar.Render(message);
}
}
```

Terdapat 5 buah jenis ProgressBar pada Telegram Bot Framework seperti pada sintaks diatas. Berikut contoh untuk menggunakan salah satu jenis dari ProgressBar.

```
Bar = new ProgressBar(0, 100, ProgressBar.EProgressStyle.standard);
Bar.Device = this.Device;
```

Jalankan program dan pastikan mendapatkan hasil seperti dibawah.



CalendarPicker

CalendarPicker adalah komponen atau elemen antarmuka pengguna (UI) yang memungkinkan pengguna memilih tanggal dari kalender. Biasanya, ini digunakan dalam pengembangan aplikasi atau situs web yang memerlukan input tanggal dari pengguna. CalendarPicker menampilkan kalender dengan navigasi yang memungkinkan pengguna untuk melihat bulan, tahun, dan hari-hari tertentu dalam kalender. Pengguna dapat memilih tanggal dengan mengklik atau mengetuk tanggal yang diinginkan pada kalender tersebut.

Untuk membuat CalendarPicker dengan Telegram Bot Framework, kita gunakan class CalendarPicker yang terdapat dalam library ini. Ikuti langkah-langkah dibawah ini :

Tambahkan sebuah class dengan nama “ CalendarPickerForm“ dengan menggunakan “AutoCleanForm“ sebagai base class. Selanjutnya buat dua buah property pada class seperti dibawah.

```
public CalendarPicker Picker { get; set; }  
int? selectedDateMessage { get; set; }
```

Lalu kita lanjutkan dengan membuat sebuah method dengan nama “CalendarPickerForm_Init“

```
private async Task CalendarPickerForm_Init(object sender, InitEventArgs e)  
{  
    this.Picker = new CalendarPicker();  
    this.Picker.Title = "Pilih Tanggal";  
    this.AddControl(Picker);  
}
```

Method diatas adalah akan digunakan untuk event Init. Dimana dengan menggunakan property Picker yang telah kita buat sebelumnya, kita set Title dari CalendarPicker. Lalu pada constructor ketikkan sintaks dibawah.

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

Selanjutnya buat method Render untuk membuat menu dan menampilkan tanggal yang dipilih oleh pengguna.

```
public override async Task Render(MessageResult message)  
{  
    String s = "";  
  
    s = "Tanggal yang dipilih : " +  
    this.Picker.SelectedDate.ToShortDateString() + "\r\n";  
    s += "Bulan yang dipilih " +  
    this.Picker.Culture.DateTimeFormat.MonthNames[this.Picker.VisibleMonth.Month - 1] +  
    "\r\n";  
    s += "Tahun yang dipilih " + this.Picker.VisibleMonth.Year.ToString();  
  
    ButtonForm bf = new ButtonForm();  
    bf.AddButtonRow(new ButtonBase("Back", "back"));  
  
    if (selectedDateMessage != null)  
    {  
        await this.Device.Edit(this.selectedDateMessage.Value, s, bf);  
    }  
    else  
    {  
        var m = await this.Device.Send(s, bf);  
        this.selectedDateMessage = m.MessageId;  
    }  
}
```

Untuk method Action dapat dilihat pada project lampiran.



MonthPicker

MonthPicker adalah elemen antarmuka pengguna (UI) yang memungkinkan pengguna untuk memilih bulan dari kalender dalam format yang terpisah. Mirip dengan CalendarPicker, MonthPicker terutama fokus pada pilihan bulan tanpa memperlihatkan hari-hari atau tanggal spesifik. Biasanya, ini digunakan dalam pengembangan aplikasi atau situs web di mana pemilihan bulan saja yang diperlukan tanpa perlu menentukan tanggal spesifik. Pengguna dapat memilih bulan dengan menavigasi melalui daftar bulan dalam tahun tertentu yang ditampilkan oleh MonthPicker.

Untuk MonthPicker konsep dan penggunaannya hampir sama dengan DatePicker. Sintak detail dapat dilihat pada project lampiran.



ToggleButton

ToggleButton adalah elemen antarmuka pengguna (UI) yang memungkinkan pengguna untuk beralih antara dua status atau mode yang berbeda dengan satu tombol. Saat tombol ditekan, statusnya berubah menjadi yang lainnya, dan ini memberikan indikasi visual tentang keadaan yang sedang aktif atau nonaktif.

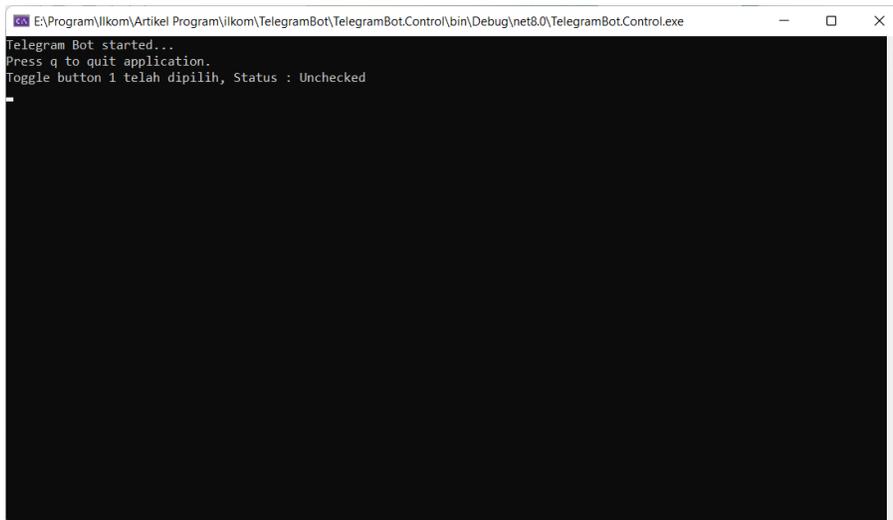
Contohnya, sebuah ToggleButton bisa mewakili mode on/off, aktif/nonaktif, atau status lainnya yang dapat diubah dengan mengklik tombol tersebut. Ketika ditekan, biasanya ada perubahan warna atau tampilan yang menunjukkan bahwa status atau mode telah berubah.

ToggleButton sering digunakan dalam aplikasi atau situs web untuk memungkinkan pengguna dengan cepat dan mudah beralih antara dua opsi atau keadaan yang berbeda.



```
private void Tb_Toggled(object sender, EventArgs e)
{
    var tb = sender as ToggleButton;
    Console.WriteLine("Toggle button " + tb.Id.ToString() +
        " telah dipilih, Status : " +
        (tb.Checked ? "Checked" : "Unchecked"));
}
```

Untuk mendapatkan nilai atau status dari ToggleButton ketikkan sintaks diatas.

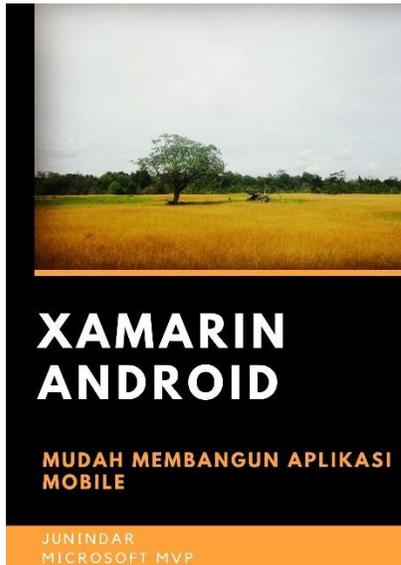


Penutup

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/2023/12/telegram-bot-framework-custom-control.html>

Referensi



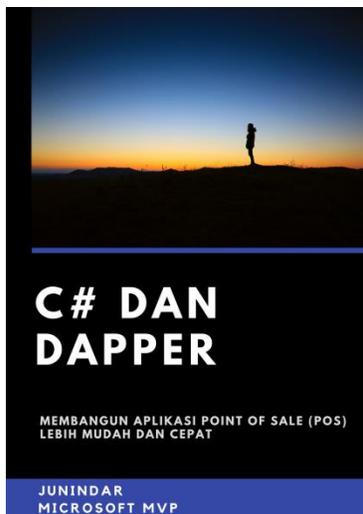
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



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



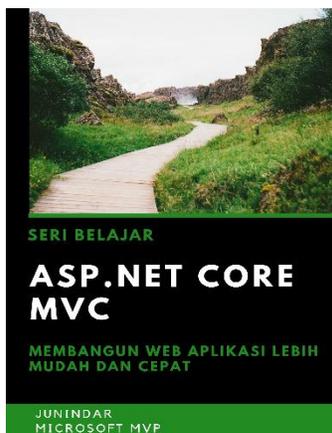
https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ



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 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=x Ee5DwAAQBAJ](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”**.