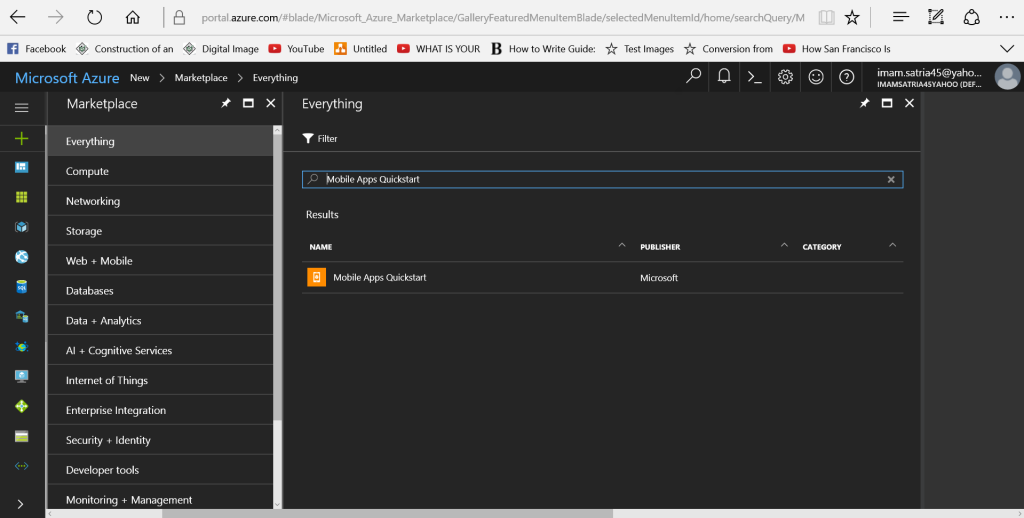
Menggunakan Azure Mobile App Serivce dan Easy Tables

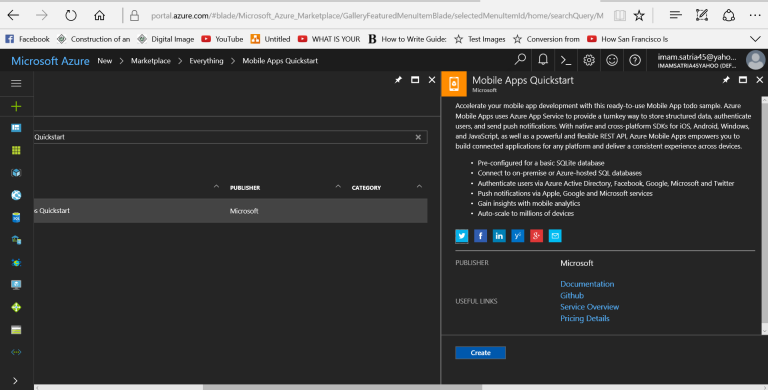
Pada Xamarin.Forms

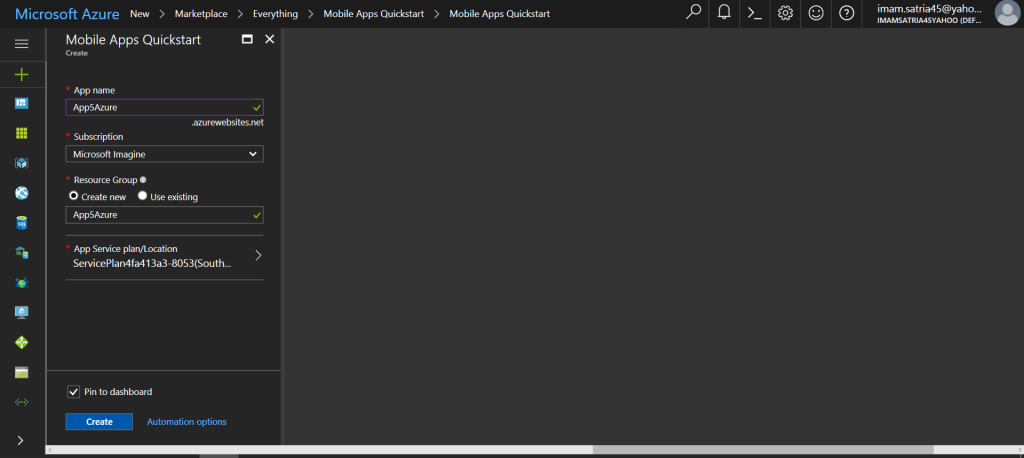
Azure Mobile App adalah salah satu layanan dari Microsoft Azure yang dikhususkan untuk mendukung aplikasi Mobile. Dengan menggunakan layanan Azure Mobile App maka pengembang aplikasi mobile tidak perlu membuat sendiri backend services secara manual. Beberapa fitur yang ada antara lain adalah: Easy Tables, Authentication, dan Easy APIs.

Pada materi kali ini kita akan mencoba menggunakan layanan Easy Tables pada Azure. Kita akan membuat aplikasi sederhana yang bisa menyimpan data sebuah buku. Ikuti langkah – langkah berikut. (Note: disarankan untuk mengetik sendri source codenya, copy-paste source code sering kali memberikan error yang tidak terdeteksi).

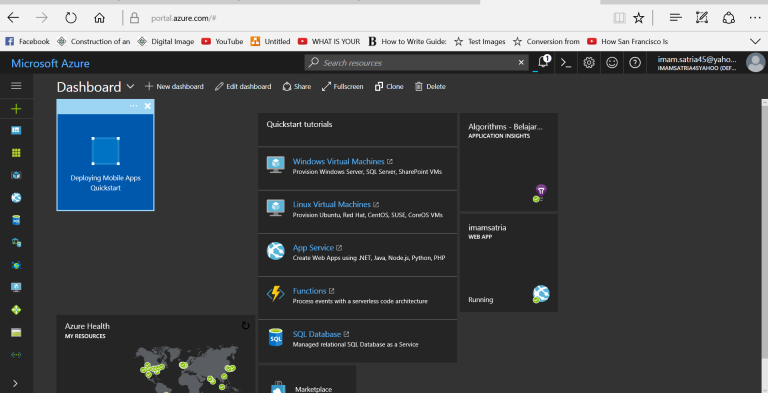
1. Pastikan Anda telah memiliki akun Microsoft Azure. Masuk kedalam portal [Microsoft Azure](http://portal.azure.com/). Kemudian pilih layanan **Mobile Apps Quickstart** untuk membuat layanan Mobile Apps dengan cepat.
2. Klik button + pada kiri layar, kemudian search Mobile Apps Quickstart.



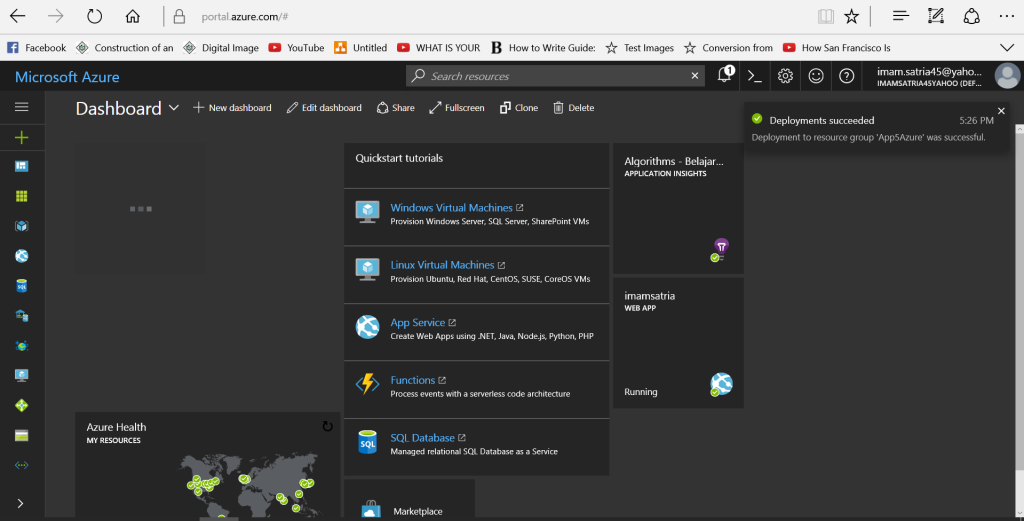
1. Klik Mobile Apps Quickstart. Kemudian klik Create.
2. Isikan App Name dan Resource Group yaitu App5Azure. Subscription Microsoft Imagine (akun ini bisa didapatkan secara gratis bila Anda seorang akademisi, tentunya dengan mendaftar terlebih dahulu). Untuk App Service plan/location biarkan saja. Klik Create.



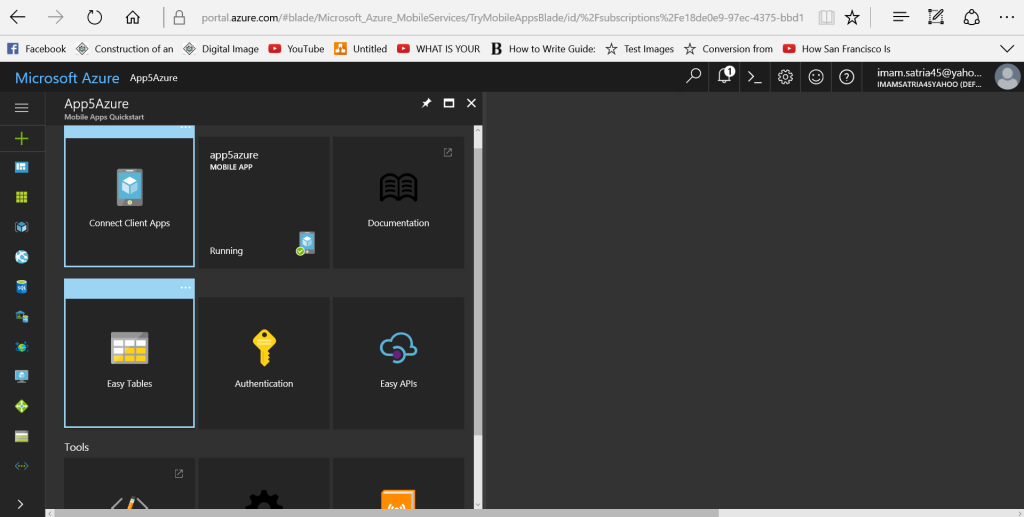
1. Tunggu deployment sampai selesai.



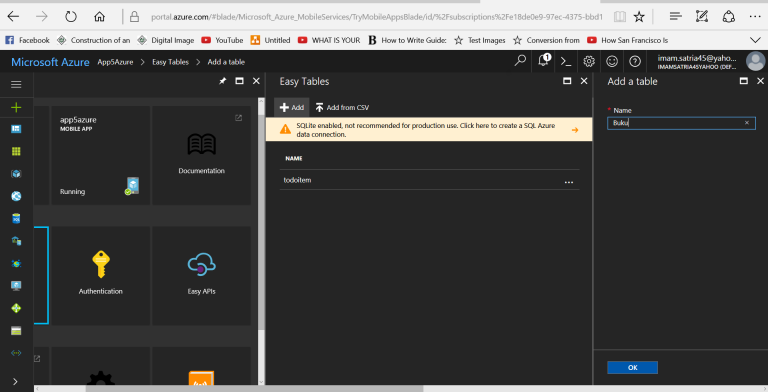
1. Akan muncul notifikasi selesai pada kanan atas layar.



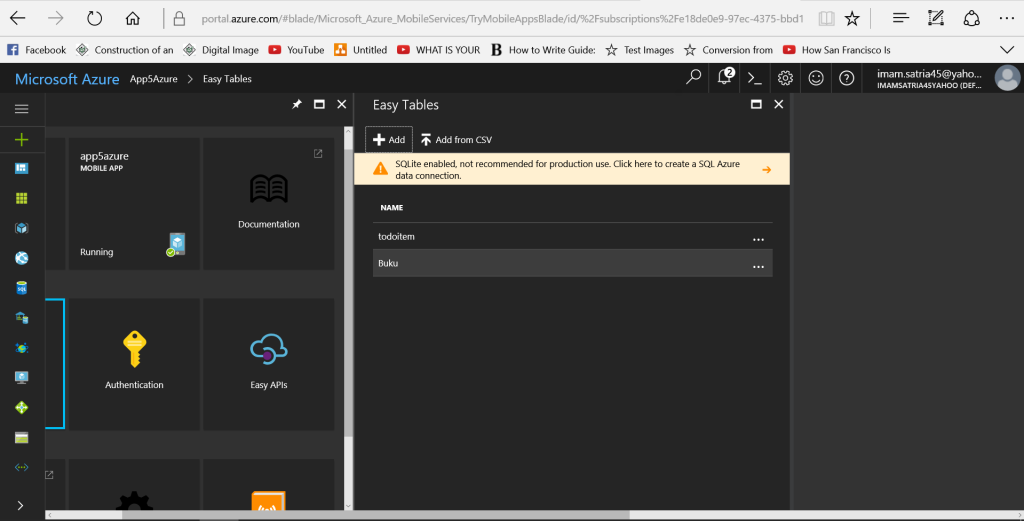
1. Buka layanan Azure yang telah kita buat tadi. Klik pada Easy Tables.



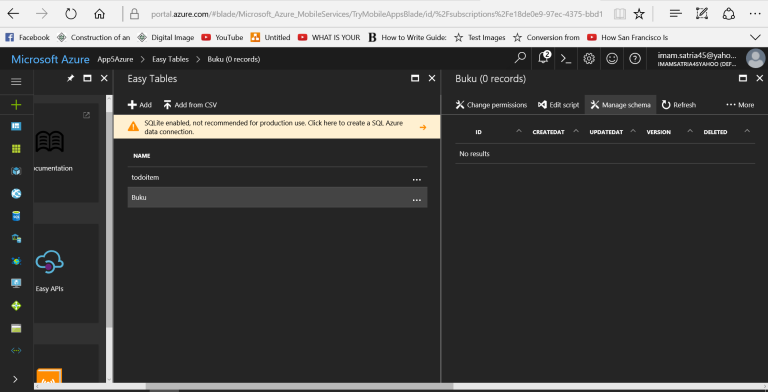
1. Klik Add untuk menambahkan tabel baru. Isikan Buku untuk Name. Kemudian klik OK.



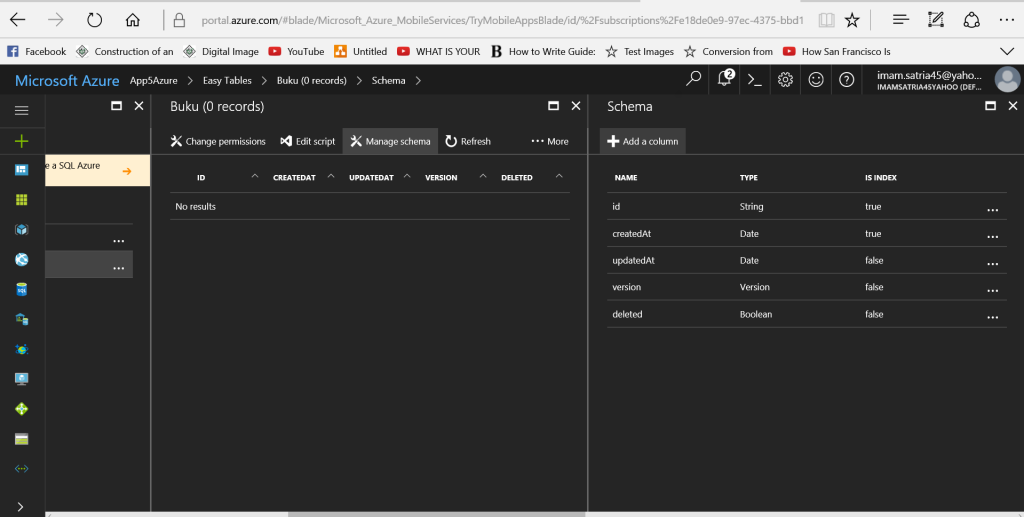
1. Klik pada tabel yang telah kita buat tadi.



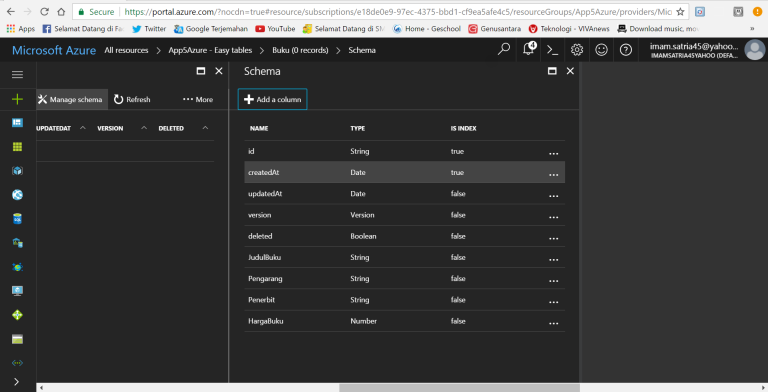
1. Klik Manage schema.



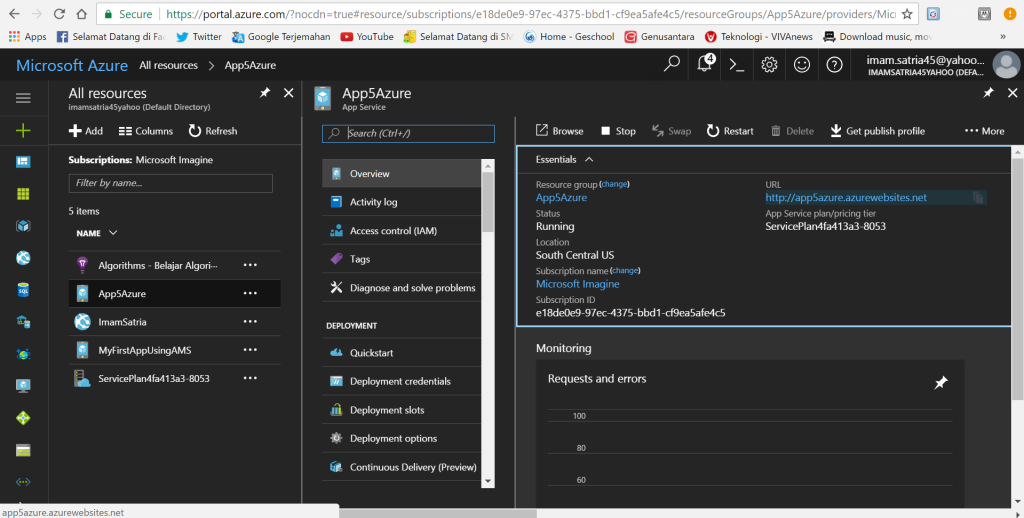
1. Klik Add a column.



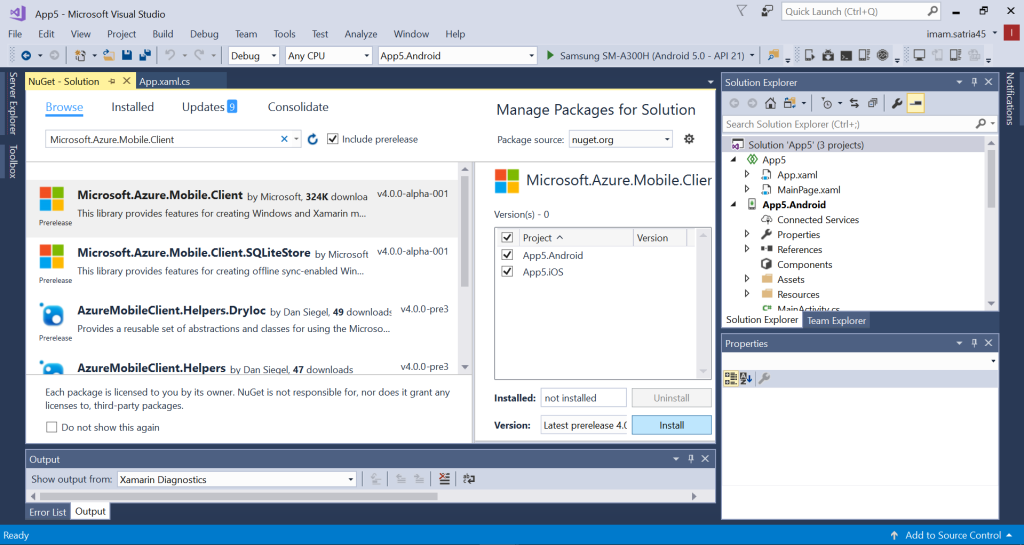
1. Tambahkan 4 buah kolom baru dengan nama JudulBuku, Pengarang, Penerbit, dan Harga. Tipe data setiap kolom adalah string, kecuali kolom Harga yang bertipe number.



1. Pada tahap ini kita telah selesai membuat komponen yang dibutuhkan untuk menyimpan informasi dari sebuah buku. Selanjutnya kita akan membuat aplikasinya. Namun, sebelum membuat aplikasi, ada baiknya kita menyiapkan alamat (URL) dari layanan Azure yang kita buat. Alamat ini dibutuhkan untuk menghubungkan aplikasi kita dengan Azure. Caranya buka kembali App5Azure, kemudian lihat pada bagian Overview. Disitu tertera alamat (URL) yang kita butuhkan.



1. Pada Visual Studio Community, buat project Cross Platform App (Xamarin). Pilih Blank App, kemudian klik OK. Tambahkan package Microsoft.Azure.Mobile.Client.



1. Tambahkan file Class baru dengan nama Buku.cs. Kelas ini berfungsi menjadi data model. Ubah kode menjadi seperti berikut.

using Microsoft.WindowsAzure.MobileServices;

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Text;

namespace App5

{

public class Buku

{

private string \_id;

[JsonProperty(PropertyName = "id")]

public string Id

{

get { return \_id; }

set { \_id = value; }

}

private string \_judulBuku;

[JsonProperty(PropertyName = "JudulBuku")]

public string JudulBuku

{

get { return \_judulBuku; }

set { \_judulBuku = value; }

}

private string \_pengarang;

[JsonProperty(PropertyName = "Pengarang")]

public string Pengarang

{

get { return \_pengarang; }

set { \_pengarang = value; }

}

private string \_penerbit;

[JsonProperty(PropertyName = "Penerbit")]

public string Penerbit

{

get { return \_penerbit; }

set { \_penerbit = value; }

}

private decimal \_harga;

[JsonProperty(PropertyName = "Harga")]

public decimal Harga

{

get { return \_harga; }

set { \_harga = value; }

}

[Version]

public string Version { get; set; }

}

}

1. Tambahkan file Class dengan nama Constant.cs. File ini berfungsi menyimpan URL dari layanan yang Azure yang kita buat. ApplicationURL didapatkan dari langkah ke – 13 (isi sesuai alamat URL Anda). Buat kodenya seperti berikut.

using System;

using System.Collections.Generic;

using System.Text;

namespace App5

{

public static class Constants

{

public static string ApplicationURL = @"http://app5azure.azurewebsites.net";

}

}

1. Tambahkan file Class dengan nama BukuManager.cs. File ini berfungsi untuk mengambil, menambah, dan mengupdate data. Ubah kode menjadi seperti berikut.

using Microsoft.WindowsAzure.MobileServices;

using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Diagnostics;

using System.Text;

using System.Threading.Tasks;

namespace App5

{

public class BukuManager

{

private IMobileServiceTable \_bukuTable;

public BukuManager()

{

var client = new MobileServiceClient(Constants.ApplicationURL);

\_bukuTable = client.GetTable();

}

public async Task GetBarangAsync()

{

try

{

IEnumerable books = await \_bukuTable.ToEnumerableAsync();

return new ObservableCollection(books);

}

catch (MobileServiceInvalidOperationException msioe)

{

Debug.WriteLine("@Invalid sync operation: {0}", msioe.Message);

}

catch (Exception e)

{

Debug.WriteLine(@"Sync error: {0}", e.Message);

}

return null;

}

public async Task SaveTaskAsync(Buku buku)

{

if (buku.Id == null)

{

await \_bukuTable.InsertAsync(buku);

}

else

{

await \_bukuTable.UpdateAsync(buku);

}

}

}

}

1. Tambahkan file Class dengan nama ActivityIndicatorScope.cs. File ini berfungsi untuk menampilkan activity indicator pada saat mengambil data pada layanan Azure. Ubah kode menjadi seperti berikut.

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

namespace App5

{

public class ActivityIndicatorScope : IDisposable

{

private bool \_showIndicator;

private ActivityIndicator \_indicator;

private Task \_indicatorDelay;

public ActivityIndicatorScope(ActivityIndicator indicator, bool showIndicator)

{

\_indicator = indicator;

\_showIndicator = showIndicator;

if (showIndicator)

{

\_indicatorDelay = Task.Delay(2000);

SetIndicatorActivity(true);

}

else

{

\_indicatorDelay = Task.FromResult(0);

}

}

private void SetIndicatorActivity(bool isActive)

{

\_indicator.IsVisible = isActive;

\_indicator.IsRunning = isActive;

}

public void Dispose()

{

if (\_showIndicator)

{

\_indicatorDelay.ContinueWith(t => SetIndicatorActivity(false),

TaskScheduler.FromCurrentSynchronizationContext());

}

}

}

}

1. Tambahkan file Content Page XAML dengan nama TambahBukuPage.xaml. File ini berfungsi untuk menambahkan dan mengupdate data buku. Ubah kode menjadi seperti berikut.

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="App5.TambahBukuPage"

Title="Tambah Buku Page">

<ContentPage.Content>

<Grid x:Name="gvBarang">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="150"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

</Grid.RowDefinitions>

<Label Text="Judu Buku: " HorizontalTextAlignment="End"

VerticalTextAlignment="Center" Grid.Row="0" Grid.Column="0"></Label>

<Entry x:Name="txtJudulBuku" Text="{Binding JudulBuku}"

Placeholder="Masukkan judul buku" Grid.Row="0" Grid.Column="1"></Entry>

<Label Text="Pengarang: " HorizontalTextAlignment="End"

VerticalTextAlignment="Center" Grid.Row="1" Grid.Column="0"></Label>

<Entry x:Name="txtPengarang" Text="{Binding Pengarang}"

Placeholder="Masukkan pengarang" Grid.Row="1" Grid.Column="1"></Entry>

<Label Text="Penerbit: " HorizontalTextAlignment="End"

VerticalTextAlignment="Center" Grid.Row="2" Grid.Column="0"></Label>

<Entry x:Name="txtPenerbit" Text="{Binding Penerbit}"

Placeholder="Masukkan penerbit" Grid.Row="2" Grid.Column="1"></Entry>

<Label Text="Harga: " HorizontalTextAlignment="End"

VerticalTextAlignment="Center" Grid.Row="3" Grid.Column="0"></Label>

<Entry x:Name="txtHarga" Text="{Binding Harga, StringFormat='{0:N}'}"

Keyboard="Numeric" Placeholder="Masukkan harga" Grid.Row="3"

Grid.Column="1"></Entry>

<StackLayout Orientation="Horizontal" Grid.Row="4" Grid.ColumnSpan="2">

<Button x:Name="btnSave" HorizontalOptions="EndAndExpand" Text="Save"

WidthRequest="150" Clicked="btnSave\_Clicked"></Button>

</StackLayout>

</Grid>

</ContentPage.Content>

</ContentPage>

1. Pada TambahBukuPage.xaml.cs, ubah kode menjadi seperti berikut.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace App5

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class TambahBukuPage : ContentPage

{

private BukuManager \_bukuManager = new BukuManager();

private bool \_isNew = false;

public TambahBukuPage ()

{

InitializeComponent ();

}

private void ClearAll()

{

foreach(var ctr in gvBarang.Children)

{

if(ctr is Entry)

{

var item = ctr as Entry;

item.Text = string.Empty;

}

}

}

public TambahBukuPage(bool isNew)

{

InitializeComponent();

\_isNew = isNew;

if (\_isNew)

txtJudulBuku.Focus();

}

private async void btnSave\_Clicked(object sender, EventArgs e)

{

if(\_isNew)

{

var buku = new Buku()

{

JudulBuku = txtJudulBuku.Text,

Pengarang = txtPengarang.Text,

Penerbit = txtPenerbit.Text,

Harga = Convert.ToDecimal(txtHarga.Text)

};

await \_bukuManager.SaveTaskAsync(buku);

ClearAll();

await DisplayAlert("Keterangan", "Data buku berhasil ditambahkan!",

"OK");

}

else

{

var buku = (Buku)this.BindingContext;

await \_bukuManager.SaveTaskAsync(buku);

await DisplayAlert("Keterangan", "Data buku berhasil diupdate!",

"OK");

}

}

}

}

1. Tambahkan file Content Page XAML dengan nama BukuPage.xaml. File ini berfungsi untuk menampilkan data buku. Ubah kode menjadi seperti berikut.

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="App5.BukuPage">

<ContentPage.ToolbarItems>

<ToolbarItem Text="+" Clicked="ToolbarItem\_Clicked"></ToolbarItem>

</ContentPage.ToolbarItems>

<ContentPage.Content>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<ActivityIndicator HorizontalOptions="Center" VerticalOptions="Center"

IsVisible="False" IsEnabled="True"

x:Name="syncIndicator"></ActivityIndicator>

<ListView HasUnevenRows="False" x:Name="listViewBuku"

ItemTapped="listViewBuku\_ItemTapped" IsPullToRefreshEnabled="True"

Refreshing="listViewBuku\_Refreshing" RowHeight="100">

<ListView.ItemTemplate>

<DataTemplate>

<ViewCell>

<StackLayout HorizontalOptions="StartAndExpand"

Orientation="Horizontal">

<StackLayout Orientation="Vertical">

<Label Text="{Binding JudulBuku}"

FontSize="16"></Label>

<StackLayout Orientation="Horizontal">

<Label Text="{Binding Pengarang}"></Label>

<Label Text="{Binding Penerbit}"></Label>

</StackLayout>

</StackLayout>

<Label Text="{Binding Harga}"

HorizontalTextAlignment="End"

HorizontalOptions="CenterAndExpand"

VerticalOptions="Center"></Label>

</StackLayout>

</ViewCell>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</ContentPage.Content>

</ContentPage>

1. Pada BukuPage.xaml.cs, ubah kode menjadi seperti berikut.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace App5

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class BukuPage : ContentPage

{

private BukuManager \_bukuManager = new BukuManager();

public BukuPage ()

{

InitializeComponent ();

}

protected override async void OnAppearing()

{

base.OnAppearing();

await RefreshItems(true);

}

private async Task RefreshItems(bool showActivityIndicator)

{

using (var scope = new ActivityIndicatorScope(syncIndicator,

showActivityIndicator))

{

listViewBuku.ItemsSource = await \_bukuManager.GetBarangAsync();

}

}

private async void ToolbarItem\_Clicked(object sender, EventArgs e)

{

TambahBukuPage tambahPage = new TambahBukuPage(true);

await Navigation.PushAsync(tambahPage);

}

private async void listViewBuku\_ItemTapped(object sender, ItemTappedEventArgs e)

{

TambahBukuPage tambahPage = new TambahBukuPage();

Buku item = (Buku)e.Item;

tambahPage.BindingContext = item;

((ListView)sender).SelectedItem = null;

await Navigation.PushAsync(tambahPage);

}

private async void listViewBuku\_Refreshing(object sender, EventArgs e)

{

var list = (ListView)sender;

Exception error = null;

try

{

await RefreshItems(false);

}

catch(Exception ex)

{

error = ex;

}

finally

{

list.EndRefresh();

}

if(error != null)

{

await DisplayAlert("Refresh Error!", "Couldn't refresh data (" +

error.Message + ")", "OK");

}

}

}

}

1. Pada App.xaml.cs, ubah kode menjadi seperti berikut.

public App ()

{

InitializeComponent();

MainPage = new NavigationPage(new BukuPage());

}

1. Jalankan aplikasi tersebut. Hasilnya seperti berikut.

