

Kelas Belajar Membangun Aplikasi Multi-platform (<https://www.dicoding.com/academies/20>)

Topik Modul 4: Navigasi

Upgrade untuk Mengikuti Kelas Secara Penuh
(<https://www.dicoding.com/academies/20/pricingplan>)

Pendahuluan

Pada modul ini akan dibahas bagaimana cara membuat navigasi antar halaman pada aplikasi Xamarin Forms. Navigasi digunakan untuk memudahkan pengguna untuk berpindah antar halaman. Selain itu pada modul ini juga akan dibahas bagaimana cara untuk mengirimkan parameter antar halaman dan membuat berbagai macam model navigasi seperti menu, tabs, dan master pages.

Jenis-jenis Navigasi pada Xamarin Forms

Ada beberapa jenis navigasi yang dapat digunakan pada Xamarin Forms, beberapa diantaranya yaitu:

- Hierarchical
- Modal
- Tabs
- Master Pages

Practice #4.1 Menambahkan Navigasi Sederhana

Pada contoh yang pertama akan ditunjukkan bagaimana cara untuk menggunakan navigasi, cara ini digunakan untuk memanggil halaman selanjutnya dari halaman pertama.

1. Buat **Solusi Xamarin Cross Platform Portable** dengan nama **Modul4**.
2. Pada project portable tambahkan halaman xaml baru dengan nama **NavigationPage1.xaml**. Kemudian tambahkan kode xaml berikut.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage Title="Navigasi Hirarki" xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="Modul4.NavigationPage1">
    <ContentPage.Content>
    <StackLayout>
        <Label Text="Home Page" FontSize="40"></Label>
        <Button Text="Go To Second Page" x:Name="btnSecond"></Button>
    </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Pada kode xaml diatas terdiri dari dua kontrol yaitu label dan button. Jika tombol ditekan maka diharapkan halaman 2 akan ditampilkan.

3. Untuk mengaktifkan mekanisme navigasi pada halaman, tambahkan objek **NavigationPage** pada file **App.xaml.cs**.

```
public App()
{
    InitializeComponent();
    MainPage = new NavigationPage(new NavigationPage1());
}
```

Pada kode diatas dapat dilihat bahwa ketika memanggil halaman **NavigationPage1** untuk pertama kali, maka objek **NavigationPage** juga harus dibuat agar fitur navigasi dapat digunakan.

4. Pada file **NavigationPage1.xaml.cs** tambahkan kode berikut:

```

public partial class NavigationPage1 : ContentPage
{
    public NavigationPage1()
    {
        InitializeComponent();
        btnSecond.Clicked += BtnSecond_Clicked;
    }

    private async void BtnSecond_Clicked(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new NavigationPage2());
    }
}

```

Method **Navigation.PushAsync()** diatas digunakan untuk memanggil halaman selanjutnya yaitu halaman **NavigationPage2**.

5. Pada project portable, tambahkan halaman xaml dengan nama **NavigationPage2.xaml**. Kemudian tambahkan kode 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="Modul4.NavigationPage2">
    <ContentPage.Content>
    <StackLayout>
        <Label Text="Second Page" FontSize="40"></Label>
        <Button x:Name="btnBack" Text="Back"/>
    </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

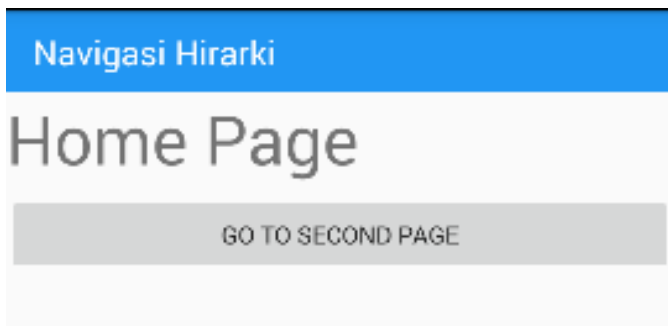
6. Kemudian tambahkan kode pada **NavigationPage2.xaml.cs**. Method **Navigation.PopAsync()** digunakan untuk kembali ke halaman sebelumnya.

```

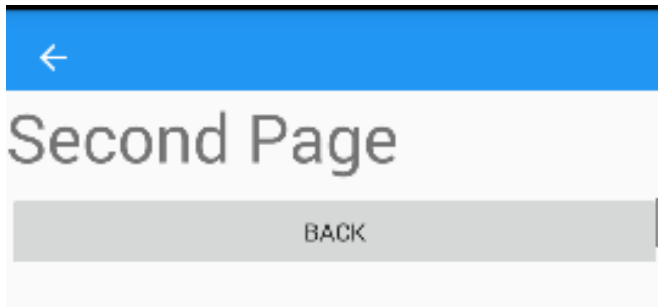
public partial class NavigationPage2 : ContentPage
{
    public NavigationPage2()
    {
        InitializeComponent();
        btnBack.Clicked += async (sender, e) =>
        {
            await Navigation.PopAsync(true);
        };
    }
}

```

7. Tekan tombol **F5** untuk menjalankan aplikasi. Ketika tombol **“Go To Second Page”** dipilih maka otomatis halaman kedua akan ditampilkan.



8. Untuk kembali ke halaman sebelumnya, anda dapat menekan tombol “**BACK**” atau menekan tombol panah dipojok kiri atas.



Practice #4.2 Membuat Dropdown Menu

Pada contoh dibawah ini akan ditunjukan bagaimana menggunakan kontrol Dropdown Menu dan cara navigasi antar halaman.

1. Pada project portable, tambahkan halaman xaml dengan nama **DropdownMenu.xaml**. kemudian tambahkan kode xaml berikut ini:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage Title="Dropdown menggunakan Toolbar" xmlns="http://xamarin.com/schemas/2014/f
orms"
            xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
            x:Class="Modul4.DropdownMenu">
    <ContentPage.ToolbarItems>
    <ToolbarItem Text="Home" Order="Secondary" Clicked="Navigate" CommandParameter="Navigation
Page1" />
    <ToolbarItem Text="Second" Order="Secondary" Clicked="Navigate" CommandParameter="Navigati
onPage2" />
    </ContentPage.ToolbarItems>
    <ContentPage.Content>
    </ContentPage.Content>
</ContentPage>
```

Pada kode xaml diatas dapat dilihat penambahan ToolbarItems untuk membuat menu dropdown.

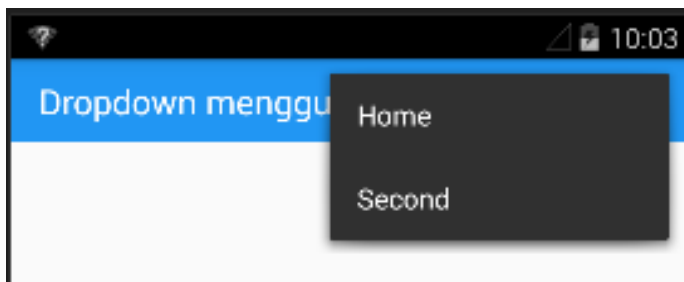
2. Kemudian tambahkan kode pada DropdownMenu.xaml.cs.

```
public partial class DropdownMenu : ContentPage
{
    public DropdownMenu()
    {
        InitializeComponent();
    }

    protected async void Navigate(object sender, EventArgs args)
    {
        string type = (string)((ToolbarItem)sender).CommandParameter;
        Type pageType = Type.GetType("Bab4." + type + ", Modul4");
        Page page = (Page)Activator.CreateInstance(pageType);
        await this.Navigation.PushAsync(page);
    }
}
```

Pada kode diatas ketika toolbar menu item dipilih maka program akan melakukan pengecekan item mana yang dipilih, kemudian membuat instan objek dari Page yang dituju, dan terakhir mengarahkan ke halaman yang dituju.

3. Untuk melihat hasilnya pada android emulator, tekan tombol F5, maka akan ditampilkan gambar sebagai berikut. Jangan lupa untuk mengganti inisialisasi page yang diload oleh aplikasi pada App.xaml.cs



#Practice 4.3 Menggunakan Modal Form

Modal Forms digunakan untuk menampilkan form bertipe modal. Ada tiga alternatif penggunaan modal form pada Xamarin Forms yaitu:

- Modal Form untuk Navigasi
- Modal Form untuk Alert
- Modal Form untuk Action Sheet pop-up menu

Dengan menggunakan modal form anda dapat menampilkan halaman full-screen. Ketika modal form ditampilkan, maka navigation bar tidak akan ditampilkan sampai form modal tersebut ditutup.

1. Pada project portable tambahkan halaman xaml baru dengan nama **ModalPage.xaml**. Kemudian tambahkan kode berikut:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage Title="Contoh Modal" xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="Modul4.ModalPage">
    <ContentPage.Content>
    <StackLayout>
        <Label Text="First Page" FontSize="40" />
        <Button Text="Go to Second Page Modally" Clicked="Navigate" />
    </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

2. Buat juga halaman xaml dengan nama **ModalSecondPage.xaml**, kemudian tambahkan kode 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="Modul4.ModalSecondPage">
    <ContentPage.Content>
    <StackLayout>
        <Label Text="Second Page" FontSize="40" />
        <Button Text="Pop back to First Page" Clicked="Navigate" />
    </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

3. Untuk menampilkan Modal Form yang sudah dibuat pada langkah sebelumnya, tambahkan kode berikut pada file **ModalPage.xaml.cs**.

```
public partial class ModalPage : ContentPage
{
    public ModalPage()
    {
        InitializeComponent();
    }

    protected async void Navigate(object sender, EventArgs args)
    {
        await Navigation.PushModalAsync(new ModalSecondPage(), false);
    }
}
```

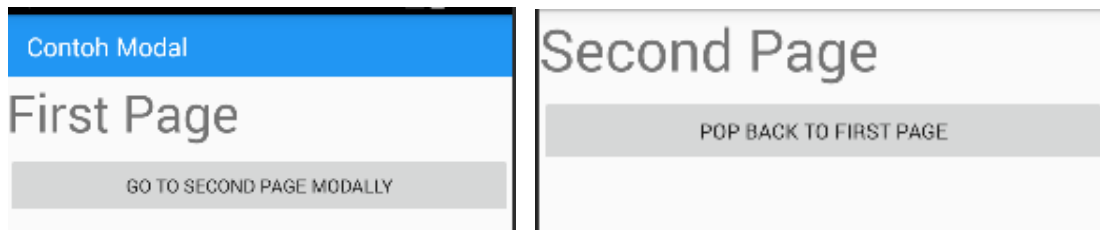
Pada kode diatas dapat dilihat bahwa method `Navigation.PushModalAsync` digunakan untuk memanggil form modal.

4. Untuk dapat kembali ke halaman sebelumnya, anda dapat menambahkan kode berikut pada file **ModalSecondPage.xaml.cs**.

```
public partial class ModalSecondPage : ContentPage
{
    public ModalSecondPage()
    {
        InitializeComponent();

        protected async void Navigate(object sender, EventArgs args)
        {
            await Navigation.PopModalAsync();
        }
    }
}
```

5. Tekan tombol F5 untuk menjalankan program pada Android Emulator. Tampilan pemanggilan modal form dapat dilihat pada gambar berikut ini. Jangan lupa untuk mengganti inisialisasi page yang di load oleh aplikasi pada `App.xaml.cs`



Practice #4.4 Membuat Popup Alert

Selain untuk menampilkan halaman baru seperti pada contoh sebelumnya, modal form juga dapat digunakan untuk membuat popup Alert.

1. Pada project portable tambahkan halaman xaml dengan nama **PopupMainPage.xaml**. kemudian tambahkan kode xaml berikut ini.

```
<?xml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:local="clr-namespace:Modul4;assembly=Modul4"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Modul4.PopupMainPage">
    <TabbedPage.Children>
    <local:AlertPage Title="Alerts" Icon="icon.png" />
    <local:ActionSheetPage Title="ActionSheets" Icon="icon.png" />
    </TabbedPage.Children>
</TabbedPage>
```

2. Kemudian tambahkan juga halaman baru dengan nama **AlertPage.xaml**.

```
<?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="Modul4.AlertPage">
    <ContentPage.Content>
    <StackLayout Padding="0,20,0,0">
        <Label Text="DisplayAlert" FontSize="Large" HorizontalOptions="Center" />
        <Button Text="Alert Simple" Clicked="OnAlertSimpleClicked" />
        <Button Text="Alert Yes/No" Clicked="OnAlertYesNoClicked" />
    </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

3. Pada file **PopupMainPage.xaml.cs** ubahlah sebagai berikut:

```
public partial class PopupMainPage : TabbedPage
{
    public PopupMainPage()
    {
        InitializeComponent();
    }
}
```

4. Pada file **AlertPage.xaml.cs** tambahkan kode berikut ini untuk menampilkan popup display alert.


```
public partial class AlertPage : ContentPage
{
    public AlertPage()
    {
        InitializeComponent();
    }
    async void OnAlertSimpleClicked(object sender, EventArgs e)
    {
        await DisplayAlert("Alert", "You have been alerted", "OK");
    }
    async void OnAlertYesNoClicked(object sender, EventArgs e)
    {
        var answer = await DisplayAlert("Question?", "Would you like to play a game",
        "Yes", "No");
    }
}
```

5. Selain alert kita juga dapat membuat action sheet popup. Action Sheet mirip seperti Alert hanya saja ada tambahan tombol konfirmasi, dan nilai kembalian konfirmasi dari pengguna dapat diambil. Untuk itu tambahkan halaman xaml dengan nama **ActionSheetPage.xaml**.

```
<?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="Modal4.ActionSheetPage">
    <ContentPage.Content>
    <StackLayout Padding="0,20,0,0">
        <Label Text="DisplayActionSheet" FontSize="Large" HorizontalOptions="Center" />
        <Button Text="ActionSheet Simple" Clicked="OnActionSheetSimpleClicked" />
        <Button Text="ActionSheet Cancel/Delete" Clicked="OnActionSheetCancelDeleteClicked" />
    </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

6. Tambahkan **code-behind** pada file **ActionSheetPage.xaml.cs**.

```

public partial class ActionSheetPage : ContentPage
{
    public ActionSheetPage()
    {
        InitializeComponent();
    }
    async void OnActionSheetSimpleClicked(object sender, EventArgs e)
    {
        var action = await DisplayActionSheet("ActionSheet: Send to?", "Cancel", null,
"Email", "Twitter", "Facebook");
    }
    async void OnActionSheetCancelDeleteClicked(object sender, EventArgs e)
    {
        var action = await DisplayActionSheet("ActionSheet: SavePhoto?", "Cancel", "De
lete", "Photo Roll", "Email");
    }
}

```

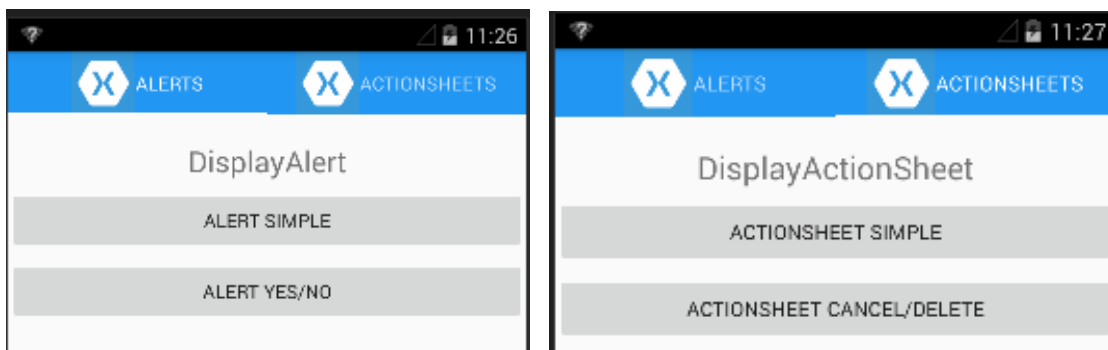
7. Untuk mengaktifkan mekanisme navigasi pada halaman, tambahkan objek **PopupMainPage** pada file **App.xaml.cs**.

```

public App()
{
    InitializeComponent();
    MainPage = new Modul4.PopupMainPage();
}

```

8. Tekan tombol **F5** untuk menjalankan aplikasi pada Android Emulator. Jangan lupa untuk mengganti inisialisasi page yang di load oleh aplikasi pada **App.xaml.cs**



Practice #4.5 Mengirimkan Data Antar Page

Pada aplikasi yang cukup kompleks dan terdiri lebih dari satu halaman dibutuhkan mekanisme untuk mengirimkan data antar halaman.

Ada beberapa cara yang dapat digunakan untuk mengirimkan data antar halaman. Cara yang pertama dan yang paling mudah adalah mengirimkan data melalui objek konstruktor.

1. Pada project portable, tambahkan **attribut x:Name="lblparam"** pada **Label** di halaman **NavigationPage2.xaml** yg sebelumnya sudah dibuat. Sehingga tampilannya berikut ini.

```
<?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="Modul4.NavigationPage2">
  <ContentPage.Content>

    <StackLayout>

      <Label x:Name ="lblParam" Text="Second Page" FontSize="40"></Label>

      <Button x:Name="btnBack" Text="Back"/>

    </StackLayout>

  </ContentPage.Content>
</ContentPage>
```

2. Kemudian tambahkan kode berikut pada halaman **NavigationPage2.xaml.cs** yang sudah dibuat pada latihan sebelumnya.

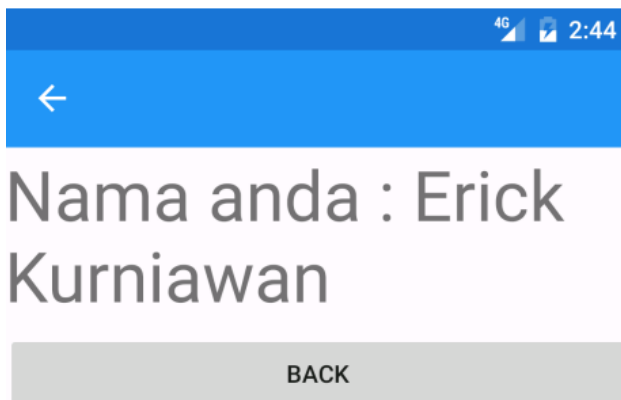
```
public NavigationPage2(string param)
{
    InitializeComponent();
    lblParam.Text = "Nama anda : " + param;
    btnBack.Clicked += async (sender, e) =>
    {
        await Navigation.PopAsync(true);
    };
}
```

Dapat dilihat pada kode diatas bahwa pada konstruktor **NavigationPage2** ditambahkan satu parameter yaitu **param** yang bertipe **string**. Kemudian nilai dari parameter tersebut akan ditampilkan pada kontrol **lblParam**.

3. Untuk mengirimkan data pada saat objek **NavigationPage2** dibuat, tambahkan kode berikut pada **NavigationPage1.xaml.cs**.

```
public partial class NavigationPage1 : ContentPage
{
    -----
    private async void BtnSecond_Clicked(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new NavigationPage2("Erick Kurniawan"));
    }
}
```

4. Tekan tombol F5 untuk menjalankan program pada Android Emulator. Tampilan pengiriman data antar page dapat dilihat pada gambar berikut ini. Jangan lupa untuk mengganti inisialisasi page yang diload oleh aplikasi pada **App.xaml.cs**



Practice #4.6 Menggunakan objek **Application.Current.Properties**

Selain menggunakan parameter pada konstruktor, anda juga dapat juga menggunakan objek **Application.Current.Properties** untuk mengirimkan data antar halaman. Objek ini mirip seperti objek session pada aplikasi web yang juga berfungsi untuk menyimpan data yang dapat diakses oleh halaman lain.

1. Pada file **app.xaml.cs**, tambahkan kode berikut untuk membuat objek **Application.Current.Properties** pada saat program dijalankan pertama kali.

```
public App()
{
    InitializeComponent();
    Application.Current.Properties["id"] = "22002321";
    MainPage = new NavigationPage(new NavigationPage1());
}
```

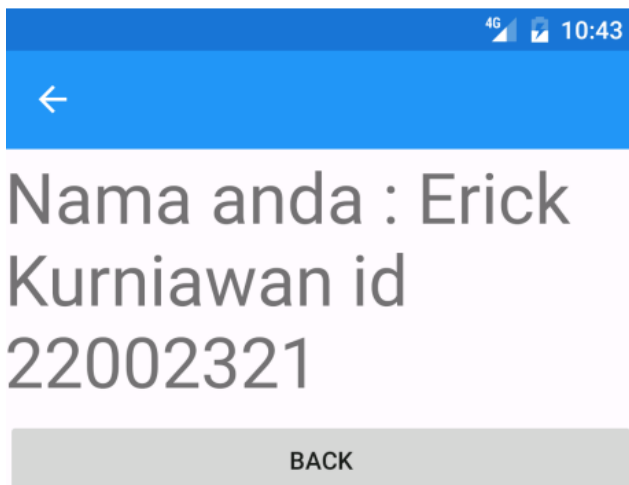
2. Kemudian untuk mengambil nilai dari objek tersebut pada halaman **NavigationPage2**, tambahkan kode berikut:

```
public partial class NavigationPage2 : ContentPage
{
    public NavigationPage2(string param)
    {
        InitializeComponent();

        lblParam.Text = "Nama anda : " + param + "id " + (string)Application.Current.Properties["id"];

        btnBack.Clicked += async (sender, e) =>
        {
            await Navigation.PopAsync(true);
        };
    }
}
```

3. Tekan tombol F5 untuk menjalankan program pada Android Emulator. Tampilan dapat dilihat pada gambar berikut ini. Jangan lupa untuk mengganti inisialisasi page yang di load oleh aplikasi pada **App.xaml.cs**



Practice #4.7 Menggunakan Master Page

Master Page adalah salah satu jenis template navigasi yang sudah disediakan pada Xamarin Forms. Dengan menggunakan master page pengguna dapat memilih menu pada area sebelah kiri halaman, dan konten dari aplikasi akan ditampilkan di area kanan halaman.

1. Pada project portable, tambahkan kode xaml berikut pada halaman MainPage.xaml.

```
<?xml version="1.0" encoding="UTF-8"?>
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:Modul4;assembly=Modul4"
    x:Class="Modul4.MainPage">
    <MasterDetailPage.Master>
        <local:MasterPage x:Name="masterPage" />
    </MasterDetailPage.Master>
    <MasterDetailPage.Detail>
        <NavigationPage>
            <x:Arguments>
                <local:ContactsPage />
            </x:Arguments>
        </NavigationPage>
    </MasterDetailPage.Detail>
</MasterDetailPage>
```

2. Langkah pertama yang harus dilakukan adalah dengan membuat container yang akan diisi dengan halaman master dan halaman navigasi.

3. Pada project portable Modul4, tambahkan class dengan nama MasterPageItem.cs. Class ini berisi data dari menu yang akan ditampilkan pada MasterPageItem.

```
public class MasterPageItem
{
    public string Title { get; set; }
    public string IconSource { get; set; }
    public Type TargetType { get; set; }
}
```

4. Kemudian pada MainPage.cs tambahkan kode berikut ini.

```

public partial class MainPage : MasterDetailPage
{
    public MainPage ()
    {
        InitializeComponent ();
        masterPage.ListView.ItemSelected += OnItemSelected;
    }

    void OnItemSelected (object sender, SelectedItemChangedEventArgs e)
    {
        var item = e.SelectedItem as MasterPageItem;
        if (item != null) {
            Detail = new NavigationPage ((Page)Activator.CreateInstance (item.TargetType));
            masterPage.ListView.SelectedItem = null;
            IsPresented = false;
        }
    }
}

```

5. Kemudian tambahkan halaman xaml dengan nama MainPage.xaml. Halaman ini akan berisi daftar menu yang akan ditampilkan.

```

<?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="Modul4.MasterPage"
    Padding="0,40,0,0"
    Icon="icon.png"
    Title="Personal Organiser">
<ContentPage.Content>
<StackLayout VerticalOptions="FillAndExpand">
<ListView x:Name="listView" VerticalOptions="FillAndExpand" SeparatorVisibility="None">
<ListView.ItemTemplate>
<DataTemplate>
<ImageCell Text="{Binding Title}" ImageSource="{Binding IconSource}" />
</DataTemplate>
</ListView.ItemTemplate>
</ListView>
</StackLayout>
</ContentPage.Content>
</ContentPage>

```

6. Pada halaman master page ditambahkan kontrol ListView untuk menampilkan menu. Kemudian pada MainPage.xaml.cs tambahkan kode berikut:

```

public partial class MasterPage : ContentPage
{
    public ListView ListView { get { return listView; } }

    public MasterPage()
    {
        InitializeComponent();

        var masterPageItems = new List<MasterPageItem> ();
        masterPageItems.Add (new MasterPageItem {
            Title = "Contacts",
            IconSource = "contacts.png",
            TargetType = typeof(ContactsPage)
        });
        masterPageItems.Add (new MasterPageItem {
            Title = "TodoList",
            IconSource = "todo.png",
            TargetType = typeof(TodoListPage)
        });
        masterPageItems.Add (new MasterPageItem {
            Title = "Reminders",
            IconSource = "reminders.png",
            TargetType = typeof(ReminderPage)
        });

        listView.ItemsSource = masterPageItems;
    }
}

```

Jika Terdapat merah pada ContactsPage, TodoListPage dan ReminderPage abaikan. Karena halaman tersebut belum dibuat. Selanjutnya akan membuat classhalaman tersebut

7. Untuk menampilkan halaman di bagian detail tambahkan halaman xaml dengan nama **ContactsPage.xaml**.

```

<?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="Modul4.ContactsPage"
    Title="Contacts Page">
    <ContentPage.Content>
    <StackLayout>
    <Label Text="Contacts data goes here" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />
    </StackLayout>
    </ContentPage.Content>
</ContentPage>

```


8. Kemudian tambahkan juga halaman xaml dengan nama **ReminderPage.xaml**.

```
<?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="Modul4.ReminderPage"
  Title="Reminder Page">
<ContentPage.Content>
<StackLayout>
<Label Text="Reminder data goes here" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />
</StackLayout>
</ContentPage.Content>
</ContentPage>
```

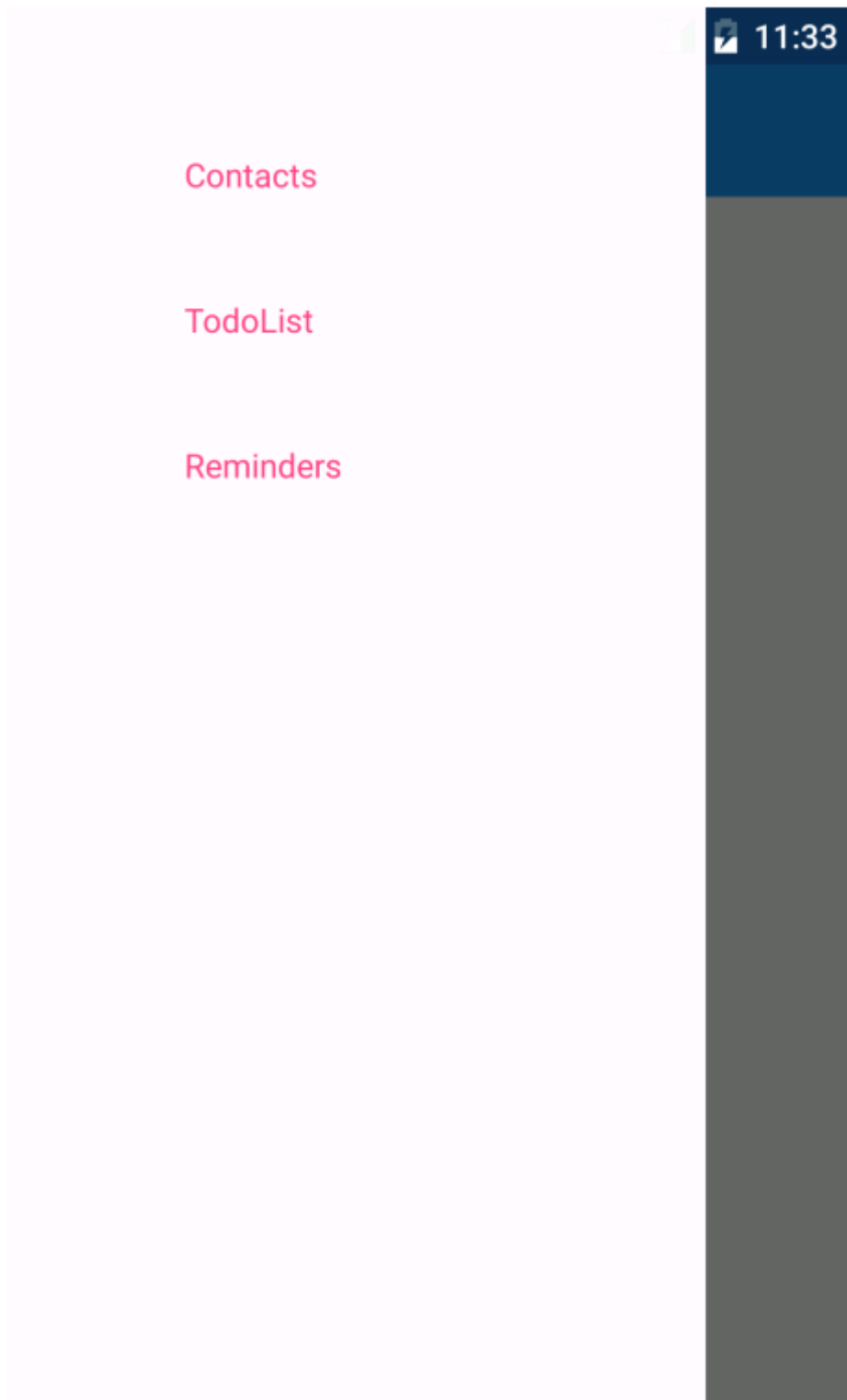
9. Terakhir tambahkan halaman xaml dengan nama **TodoListPage.xaml**.

```
<?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="Modul4.TODOListPage"
  Title="TODOList Page">
<ContentPage.Content>
<StackLayout>
<Label Text="TODO list data goes here" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />
</StackLayout>
</ContentPage.Content>
</ContentPage>
```

10. Pada file **App.xaml.cs** tambahkan kode berikut untuk menjalankan halaman **MainPage** ketika aplikasi dijalankan untuk pertama kali.

```
public App ()
{
  MainPage = new Modul4.MainPage();
}
```

11. Tekan tombol **F5** untuk menjalankan program pada Android Emulator. Tampilan pengiriman master page dapat dilihat pada gambar berikut ini.



#Practice 4.8 Menambahkan Tab Page

Selain Master Page, Xamarin Forms juga menyediakan template untuk menggunakan Tab Page. Tab Page digunakan jika ingin menampilkan banyak halaman yang akan digabungkan pada satu halaman saja.

1. Pada project portable tambahkan halaman xaml baru dengan nama **TabbedPageDemo.xaml**. Kemudian tambahkan kode xaml berikut:

```

<?xml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:local="clr-namespace:Modul4;assembly=Modul4"
             x:Class="Modul4.TabbedPageDemo">
    <TabbedPage.Resources>
    <ResourceDictionary>
        <local:NonNullToBooleanConverter x:Key="booleanConverter" />
    </ResourceDictionary>
    </TabbedPage.Resources>
    <TabbedPage.ItemTemplate>
    <DataTemplate>
        <ContentPage Title="{Binding Name}" Icon="monkeyicon.png">
            <StackLayout Padding="5, 25">
                <Label Text="{Binding Name}"
                       Font="Bold, Large"
                       HorizontalOptions="Center" />
                <Image Source="{Binding PhotoUrl}"
                       WidthRequest="200"
                       HeightRequest="200" />
                <StackLayout Padding="50, 10">
                    <StackLayout Orientation="Horizontal">
                        <Label Text="Family:"
                               HorizontalOptions="FillAndExpand" />
                        <Label Text="{Binding Family}"
                               Font="Bold, Medium" />
                    </StackLayout>
                    <StackLayout Orientation="Horizontal"
                               IsVisible="{Binding Subfamily,
                               Converter={StaticResource booleanConverter}}">
                        <Label Text="Subfamily:"
                               HorizontalOptions="FillAndExpand" />
                        <Label Text="{Binding Subfamily}"
                               Font="Bold, Medium" />
                    </StackLayout>
                    <StackLayout Orientation="Horizontal"
                               IsVisible="{Binding Tribe,
                               Converter={StaticResource booleanConverter}}">
                        <Label Text="Tribe:"
                               HorizontalOptions="FillAndExpand" />
                        <Label Text="{Binding Tribe}"
                               Font="Bold, Medium" />
                    </StackLayout>
                    <StackLayout Orientation="Horizontal">
                        <Label Text="Genus:"
                               HorizontalOptions="FillAndExpand" />
                        <Label Text="{Binding Genus}"
                               Font="Bold, Medium" />
                    </StackLayout>
                </StackLayout>
            </StackLayout>
        </ContentPage>
    </DataTemplate>

```

```
</TabbedPage.ItemTemplate>  
</TabbedPage>
```

Pada elemen **TabbedPage** dapat ditambahkan template yang berisi layout dari Tab halaman yang akan dibuat.

2. Untuk menambahkan icon berupa file .png, anda dapat menaruh icon tersebut kedalam project Droid pada folder **Resources\drawable** seperti yang sudah ditunjukan pada modul sebelumnya. Contoh disini dengan nama file **monkeyicon.png**

3. Kemudian tambahkan kode berikut pada file **App.xaml.cs** untuk menjalankan halaman **TabbedPageDemo** ketika aplikasi dijalankan untuk pertama kali.

```
public App()  
{  
    MainPage = new TabbedPageDemo();  
}
```

4. Buat class baru dengan nama **MonkeyDataModel.cs**. Class ini akan digunakan sebagai sumber data yang ditampilkan pada kontrol Tab.

```
public class MonkeyDataModel
{
    public string Name { set; get; }
    public string Family { set; get; }
    public string Subfamily { set; get; }
    public string Tribe { set; get; }
    public string Genus { set; get; }
    public string PhotoUrl { set; get; }
    public static IList<MonkeyDataModel> All { set; get; }

    static MonkeyDataModel ()
    {
        All = new ObservableCollection<MonkeyDataModel> {
            new MonkeyDataModel {
                Name = "Chimpanzee",
                Family = "Hominidae",
                Subfamily = "Homininae",
                Tribe = "Panini",
                Genus = "Pan",
                PhotoUrl = "http://upload.wikimedia.org/wikipedia/commons/thumb/6/62/Schimpanse_Zoo_Leipzig.jpg/640px-Schimpanse_Zoo_Leipzig.jpg",
            },
            new MonkeyDataModel {
                Name = "Orangutan",
                Family = "Hominidae",
                Subfamily = "Ponginae",
                Genus = "Pongo",
                PhotoUrl = "http://upload.wikimedia.org/wikipedia/commons/b/be/Orang_Utan%2C_Semenggok_Forest_Reserve%2C_Sarawak%2C_Borneo%2C_Malaysia.JPG",
            },
            new MonkeyDataModel {
                Name = "Tamarin",
                Family = "Callitrichidae",
                Genus = "Saguinus",
                PhotoUrl = "http://upload.wikimedia.org/wikipedia/commons/thumb/8/85/Tamarin_portrait_2_edit3.jpg/640px-Tamarin_portrait_2_edit3.jpg",
            }
        };
    }
}
```

5. Tambahkan kode **NotNullToBooleanConverter.cs** pada project portable. Kode ini digunakan untuk mengkonversi data yang akan ditampilkan pada tab jika datanya bernilai null.

```
class NonNullToBooleanConverter : IValueConverter
{
    public object Convert (object value, Type targetType, object parameter, CultureInfo culture)
    {
        if (value is string) {
            return !string.IsNullOrEmpty ((string)value);
        }

        return value != null;
    }

    public object ConvertBack (object value, Type targetType, object parameter, CultureInfo culture)
    {
        return null;
    }
}
```

6. Langkah terakhir tambahkan kode pada file **TabbedPageDemo.cs**.

```
public TabbedPageDemo()
{
    InitializeComponent ();
    ItemsSource = MonkeyDataModel.All;
}
```

Tekan tombol **F5** untuk menjalankan program pada Android Emulator. Tampilan pengiriman master page dapat dilihat pada gambar berikut ini.

**CHIMPANZEE****ORANGUTAN****TAMARIN**

Chimpanzee



Family:	Hominidae
Subfamily:	Homininae
Tribe:	Panini
Genus:	Pan

← Ke Halaman Kelas (<https://www.dicoding.com/academies/20>)

Selesai & Lanjutkan → (<https://www.dicoding.com/academies/20/tutorials/710?from=698>)

← Sebelumnya (<https://www.dicoding.com/academies/20/tutorials/671>)

 (<https://www.facebook.com/dicoding>) 
(<https://twitter.com/dicoding>)

Bantuan (<https://www.dicoding.com/consultation>) Blog (<https://blog.dicoding.com>) FAQ
(<https://www.dicoding.com/faq>) Aturan Pakai (<https://www.dicoding.com/termsfuse>) Privacy
(<https://www.dicoding.com/privacy>)

© 2017 Dicoding Indonesia

