

Membuat Aplikasi Android Login Screen Menggunakan SQLITE dan XAMARIN

Junindar, ST, MCPD, MOS, MCT, MVP .NET
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

Android adalah salah satu *mobile oprating system* yang saat banyak digunakan di berbagai *smartphone*, baik yang berspesifikasi rendah maupun mumpuni. Dikarenakan android saat ini merupakan salah satu mobile OS paling digemari, maka tidak dipungkiri sangat banyak sekali aplikasi-aplikasi mobile untuk android. Untuk membuat aplikasi mobile ini ada beberapa tools yang bisa digunakan dan salah satunya adalah dengan menggunakan Xamarin.

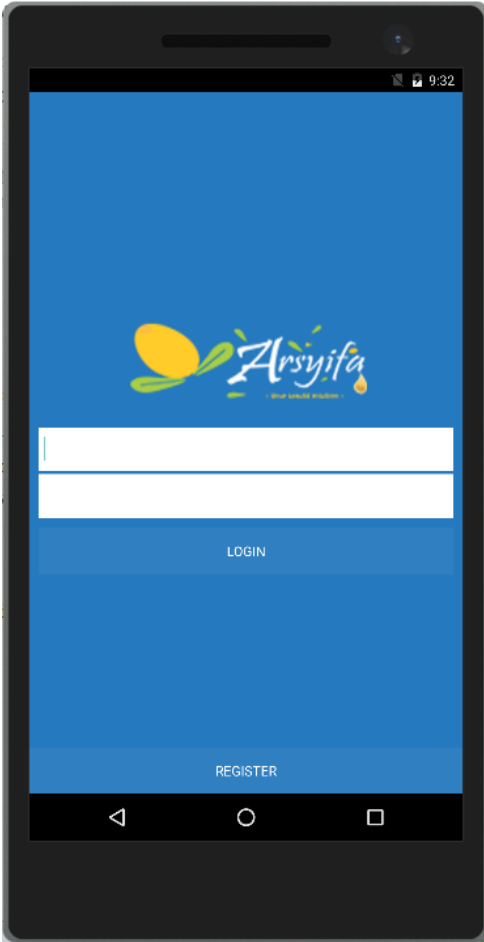
Pendahuluan

Pada artikel ini akan dijelaskan bagaimana cara untuk membuat Login Screen pada aplikasi android, dengan menggunakan SQLITE sebagai media penyimpanan datanya.

SQLITE adalah salah satu database yang bersifat open source, yang di rancang untuk menyimpan data pada perangkat elektronik yang memiliki memory terbatas. SQLite mendukung penggunaan standar relational database seperti SQL syntax dan transaction. SQLITE juga tersedia untuk Android. Sedangkan tipe data yang didukung oleh SQLITE adalah Numeric (integer,float dan double), Text (char,varchar dan text), DateTime dan BLOB.

Login screen biasanya digunakan untuk aplikasi yang memerlukan otorisasi pada penggunanya. Dan untuk menyimpan username dan password didalam device kita memerlukan SQLITE sebagai databasenya.

Tools yang kita gunakan untuk membangun aplikasi ini adalah Xamarin. Xamarin adalah salah tools untuk membuat aplikasi mobile dimana bahasa pemograman yang digunakan adalah C#. Xamarin merupakan perusahaan perangkat lunak yang berdiri pada bulan Mei tahun 2011. *Integrated development environment* (IDE) pada Xamarin di kenal dengan nama Xamarin Studio. Tetapi selain Xamarin Studio kita juga dapat menggunakan Visual Studio dan pada artikel ini kita akan menggunakan Visual Studio 2015.

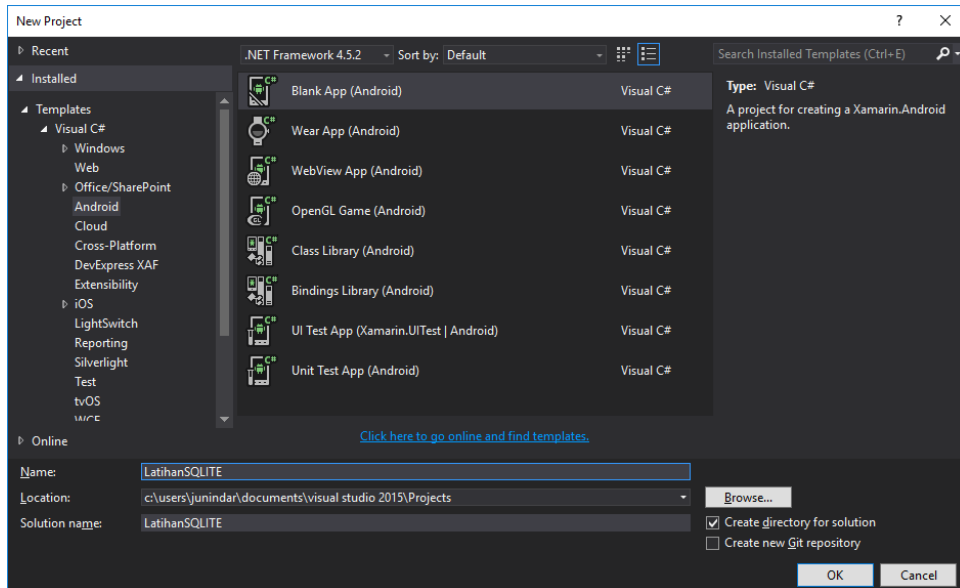


Isi

Pada latihan ini kita akan membuat sebuah aplikasi Login beserta registrasi screen untuk user mendaftar sebelum masuk kedalam Main Screen nya.

Untuk memudahkan memahami isi dari artikel ini, kita akan membuat sebuah project latihan, Ikuti langkah-langkah dibawah ini.

1. Buat sebuah project Android dengan nama “LatihanSQLITE”.



2. Setelah berhasil membuat project diatas, maka pada Solution Explorer akan terdapat beberapa file. Ganti nama file MainActivity menjadi LoginActivity. Pastikan nama class juga berubah. Pada folder Resources > Layout, tambahkan sebuah item “Android Layout” dengan nama Login.axml
3. Buka Login.axml (source) lalu ganti sintaks axml nya seperti dibawah.

```
<?xml version= 1.0 encoding= utf-8 ?>
<RelativeLayout xmlns:android= http://schemas.android.com/apk/res/android
    android:layout_width= match_parent
    android:layout_height= match_parent
    android:background= #2579BF
    android:orientation= vertical >
    <LinearLayout
        android:layout_width= fill_parent
        android:layout_height= wrap_content
        android:layout_margin= 10dp
        android:layout_centerInParent= true
        android:id= @+id/login_layout
        android:orientation= vertical >
        <ImageView
            android:layout_width= wrap_content
            android:layout_height= wrap_content
            android:layout_gravity= center
            android:layout_marginBottom= 20dp
            android:src= @drawable/splash_logo_arsyifa />
        <EditText
            android:id= @+id/txtuserName
            android:layout_width= fill_parent
            android:layout_height= 45dp
            android:padding= 5dp
            android:textColor= @android:color/black
            android:background= @android:color/white />
        <View
            android:layout_width= fill_parent
            android:layout_height= 3dp
            android:background= #2579BF />
        <EditText
            android:id= @+id/txtpassword
            android:layout_width= fill_parent
            android:layout_height= 45dp
            android:padding= 5dp
            android:background= @android:color/white
            android:textColor= @android:color/black
            android:inputType= textPassword />
        <Button
            android:id= @+id/btnlogin
            android:layout_width= fill_parent
            android:layout_height= wrap_content
            android:layout_below= @id/txtpassword
            android:layout_marginTop= 10dp
            android:background= #307FC1
            android:text= Login
            android:textColor= @android:color/white />
    </LinearLayout>
```

```
<LinearLayout
    android:layout_width= fill_parent
    android:layout_height= wrap_content
    android:layout_gravity= right
    android:background= @android:color/black
    android:layout_alignParentBottom= true
    android:gravity= right
    android:orientation= vertical >
    <Button
        android:id= @+id/btnregister
        android:layout_width= fill_parent
        android:layout_height= wrap_content
        android:background= #307FC1
        android:text= Register
        android:textColor= @android:color/white />
</LinearLayout>
</RelativeLayout>
```

Dari sintaks diatas akan mendapatkan design seperti gambar dibawah.



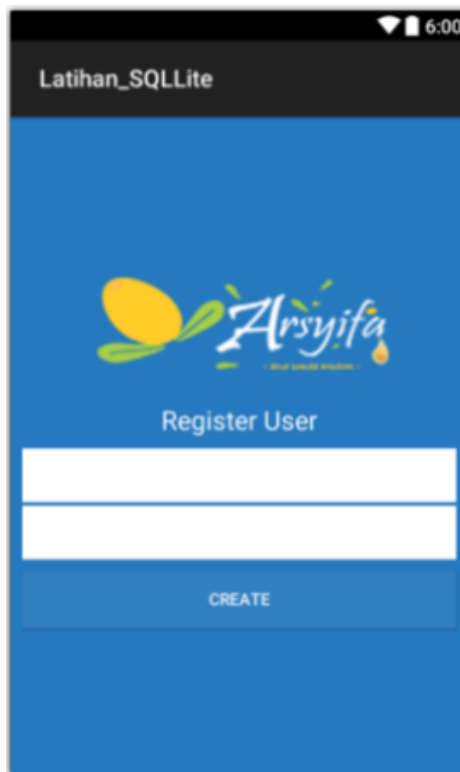
Screen diatas adalah screen login, dimana pada screen tersebut terdapat button Register. Button ini berfungsi untuk membawa kita ke Screen Register. Dimana user

sebelum melakukan proses login terlebih dahulu harus melakukan registrasi dengan cara memasukkan username dan password.

4. Untuk keperluan Register, tambahkan sebuah Android Layout dengan nama Register.axml. lalu ganti sintaks layout tersebut seperti dibawah

```
<?xml version= 1.0 encoding= utf-8 ?>
<RelativeLayout xmlns:android= http://schemas.android.com/apk/res/android
    android:layout_width= match_parent
    android:layout_height= match_parent
    android:background= #2579BF
    android:orientation= vertical >
    <LinearLayout
        android:layout_width= fill_parent
        android:layout_height= wrap_content
        android:layout_centerInParent= true
        android:layout_margin= 10dp
        android:orientation= vertical >
        <ImageView
            android:layout_width= wrap_content
            android:layout_height= wrap_content
            android:layout_gravity= center
            android:layout_marginBottom= 20dp
            android:src= @drawable/splash_logo_arsyifa />
        <TextView
            android:text= Register User
            android:textAppearance= ?android:attr/textAppearanceLarge
            android:layout_width= match_parent
            android:layout_height= wrap_content
            android:gravity= center
            android:layout_marginBottom= 10dp
            android:id= @+id/lblRegister />
        <EditText
            android:id= @+id/txtRegUsername
            android:layout_width= fill_parent
            android:layout_height= 45dp
            android:padding= 5dp
            android:textColor= @android:color/black
            android:background= @android:color/white />
        <View
            android:layout_width= fill_parent
            android:layout_height= 3dp
            android:background= #2579BF />
        <EditText
            android:id= @+id/txtRegPassword
            android:layout_width= fill_parent
            android:layout_height= 45dp
            android:padding= 5dp
            android:background= @android:color/white
            android:textColor= @android:color/black
            android:inputType= textPassword />
```

```
<Button
    android:id="@+id/btnCreate"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/txtRegPassword"
    android:layout_marginTop="10dp"
    android:background="#307FC1"
    android:text="Create"
    android:textColor="@android:color/white" />
</LinearLayout>
</RelativeLayout>
```



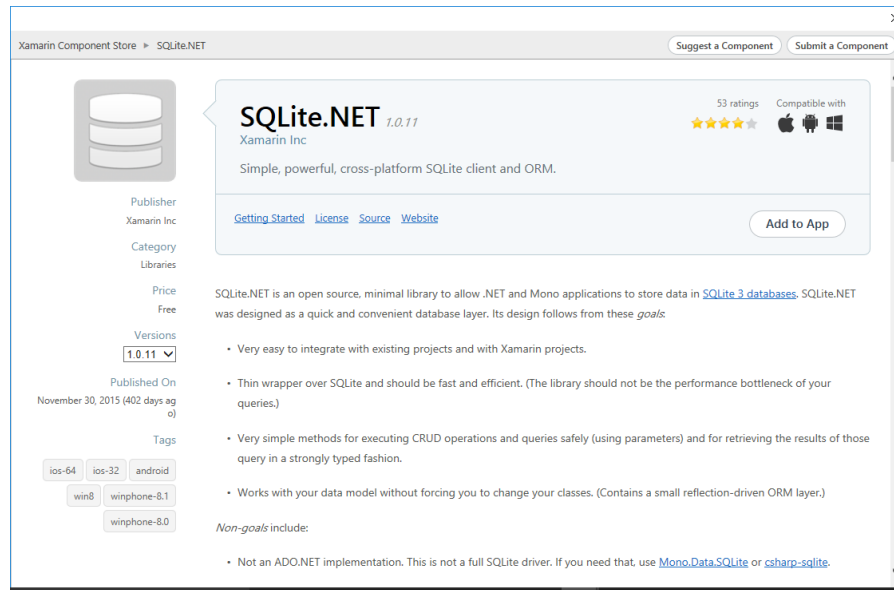
Lalu pada Layout Main.xml ganti sintaks nya menjadi seperti dibawah.


```
<?xml version= 1.0 encoding= utf-8 ?>
<RelativeLayout xmlns:android= http://schemas.android.com/apk/res/android
    android:layout_width= match_parent
    android:layout_height= match_parent
    android:background= #2579BF
    android:orientation= vertical >
    <LinearLayout
        android:layout_width= fill_parent
        android:layout_height= wrap_content
        android:layout_margin= 10dp
        android:layout_centerInParent= true
        android:id= @+id/login_layout
        android:orientation= vertical >
        <ImageView
            android:layout_width= wrap_content
            android:layout_height= wrap_content
            android:layout_gravity= center
            android:layout_marginBottom= 20dp
            android:src= @drawable/splash_logo_arsyifa />
        <TextView
            android:text= Home
            android:id= @+id/lblWelcome
            android:textAppearance= ?android:attr/textAppearanceLarge
            android:layout_width= match_parent
            android:layout_height= wrap_content
            android:gravity= center
            android:layout_marginBottom= 10dp />
    </LinearLayout>
</RelativeLayout>
```

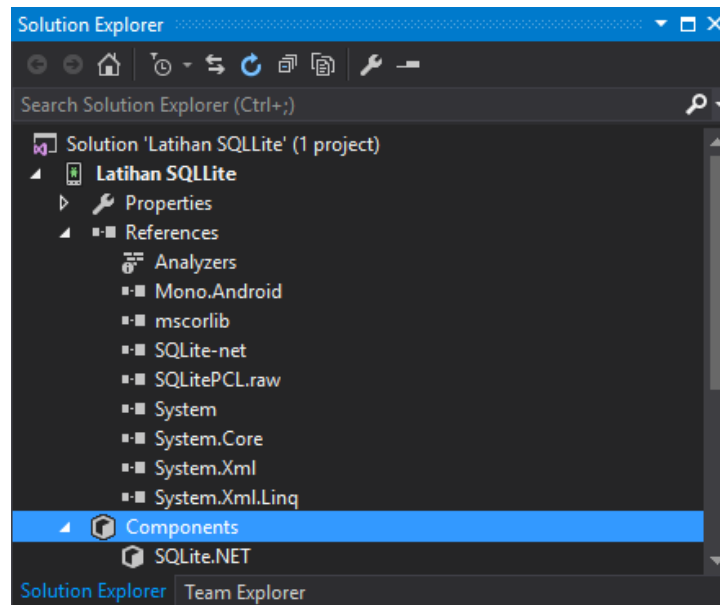
Jika dilihat dari ketiga layout diatas, semua layout memiliki ImageView yang menampilkan image dari folder “drawable”. Tambahkan sebuah image kedalam folder tersebut. Lalu ganti sintaks untuk memanggil gambar nya seperti dibawah “android:src="@drawable/namafilename".

5. Kemudian kita akan membuat Class untuk bekerja dengan SQLITE.

Sebelumnya kita harus menambahkan component SQLITE pada project. Pada solution Explorer, klik kanan Components > Get More Components. Lalu cari dengan kata kunci SQLITE.



Klik button Add to APP, sehingga pada project akan bertambah reference nya seperti dibawah.



Lalu tambahkan sebuah folder dengan nama DAO, lalu tambahkan sebuah class dengan nama UserTable.cs. Class ini digunakan untuk membuat table, dimana pada class ini akan di deklarasikan nama-nama column berserta tipe datanya. Sebelumnya import terlebih dahulu SQLITE name space. Dan ketikkan sintaks dibawah.

```
[PrimaryKey, AutoIncrement, Column("_Id")]  
public int id { get; set; } // AutoIncrement and set primarykey  
  
[MaxLength(25)]  
public string username { get; set; }  
  
[MaxLength(15)]  
public string password { get; set; }
```

Terdapat 3 field pada UserTable, id sebagai Primary Key dengan tipe data nya integer dan auto increament. Lalu ada username dan password dengan maximal karakternya masing-masing 25 dan 15.

Selanjutnya tambahkan sebuah class UserDatabase pada folder DAO. Tambahkan namespace System.IO, System.Linq dan SQLite.

Tambahkan sebuah SQLiteConnection, property dan function seperti dibawah.

```
private SQLiteConnection _connection;  
public string Message { get; set; }  
  
private static bool TableExists<T>(SQLiteConnection connection)  
{  
    const string cmdText = "SELECT name FROM sqlite_master WHERE  
    type='table' AND name=?";  
    var cmd = connection.CreateCommand(cmdText, typeof(T).Name);  
    return cmd.ExecuteScalar<string>() != null;  
}
```

Auto property ini akan digunakan untuk mendapatkan hasil dari operasi-operasi yang akan kita buat, sedangkan function TableExists berfungsi untuk mengecek apakah sebuah table itu sudah ada atau tidak pada database. Tambahkan sebuah procedure untuk membuat database seperti dibawah.

```
public void CreateDatabase()  
{  
    string dbPath = Path.Combine(System.Environment.GetFolderPath(  
        System.Environment.SpecialFolder.Personal),  
        "user.db");  
    _connection = new SQLiteConnection(dbPath);  
    if (!TableExists<UserTable>(_connection))  
    {  
        _connection.CreateTable<UserTable>();  
    }  
}
```

Procedure diatas digunakan untuk membuat Database dan TableUser. Database akan disimpan pada Special Folder dengan nama user.db, Sedangkan untuk table sebelum membuat table akan di cek terlebih dahulu apakah table tersebut exist atau tidak.

Selanjutnya dengan membuat function AddUser, function ini akan digunakan pada screen register.

```
public bool AddUser(string struser, string strpassword)
{
    var bLogin = false;
    var data = _connection.Table<UserTable>().ToList();

    if (data.Any())
    {
        Message = "Anda telah melakukan registrasi";
    }
    else
    {
        var user = new UserTable
        {
            username = struser,
            password = strpassword
        };
        _connection.Insert(user);
        Message = "Registrasi sukses";
        bLogin = true;
    }
    return bLogin;
}
```

Jika proses registrasi sukses maka return yang didapatkan berupa nilai Boolean = true. Sebelum memasukkan data kedalam table, akan dicek apakah sudah ada data pada TableUser. (aplikasi ini dibuat untuk single user saja) Jika sudah ada makan user akan menerima pesan “Anda Telah melakukan proses registrasi” dengan return false. Jika masih kosong maka pesan yang didapat “Registrasi Sukses”.

Sedangkan untuk proses login, kita akan membuat sebuah function. Jika password dan username benar, maka return yang didapat adalah true, jika salah maka akan ada pesan “Username atau password salah” dan return false.

```
public bool CheckLogin(string struser, string strpassword)
{
    var bLogin = false;

    var data = _connection.Table<UserTable>().ToList();
    var result = data.FirstOrDefault(x =>
    x.username == struser && x.password == strpassword);
    if (result != null)
    {
        bLogin = true;
    }
    else
    {
        Message = "Username atau Password salah, " +
        "silahkan ulangi kembali";
    }

    return bLogin;
}
```

Dan terakhir buat sebuah constructor seperti dibawah.

```
public UserDatabase()
{
    Message = "";
    CreateDatabase();
}
```

Dengan adanya constructor seperti diatas, pada setiap kali kita mengakses class ini, maka procedure CreateDatabase pun akan di eksekusi.

6. Setelah selesai dengan SQLite, selanjutnya kita akan membuat activity untuk masing-masing screen. Activiy merupakan representasi dari screen pada aplikasi. Sebelumnya kita telah ada 3 buah screen, maka untuk aplikasi ini kita juga memerlukan 3 activity. Tambahkan dua buah Activity masing dengan nama RegisterActivity.cs dan MainActivity.cs.

Buka LoginActivity tambahkan sintaks seperti dibawah.

```
private Button btnLogin;
private Button btnRegister;
EditText txtusername;
EditText txtPassword;
UserDatabase userdb;
```

Sintaks diatas dapat dilihat, kita mendklarasikan beberapa control yaitu 2 buah button , EditText dan class UserDatabase. Pada OnCreate ketikkan sintaks dibawah,

```
RequestWindowFeature(WindowFeatures.NoTitle);
base.OnCreate(bundle);

// Set our view from the "main" layout resource
SetContentView(Resource.Layout.Login);

userdb = new UserDatabase();
btnLogin = FindViewById<Button>(Resource.Id.btnlogin);
btnRegister = FindViewById<Button>(Resource.Id.btnregister);
txtusername = FindViewById<EditText>(Resource.Id.txtuserName);
txtPassword = FindViewById<EditText>(Resource.Id.txtpassword);

btnLogin.Click += btnLogin_Click;

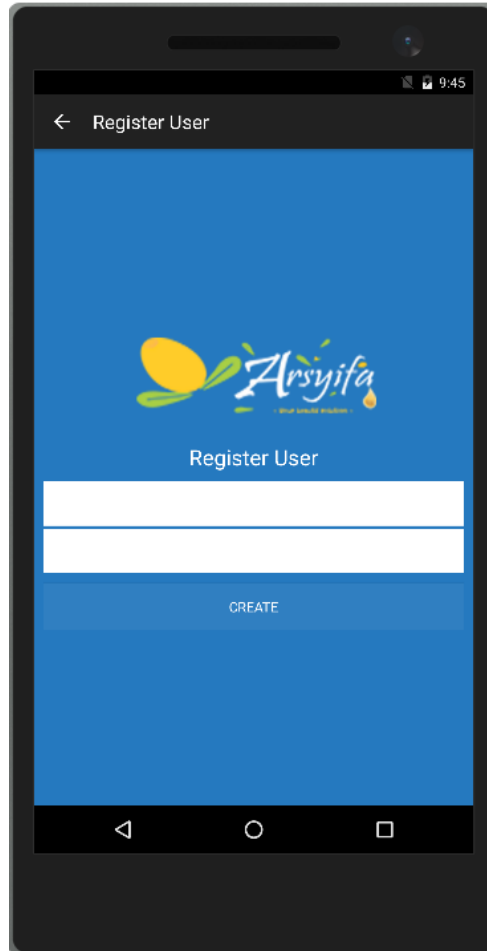
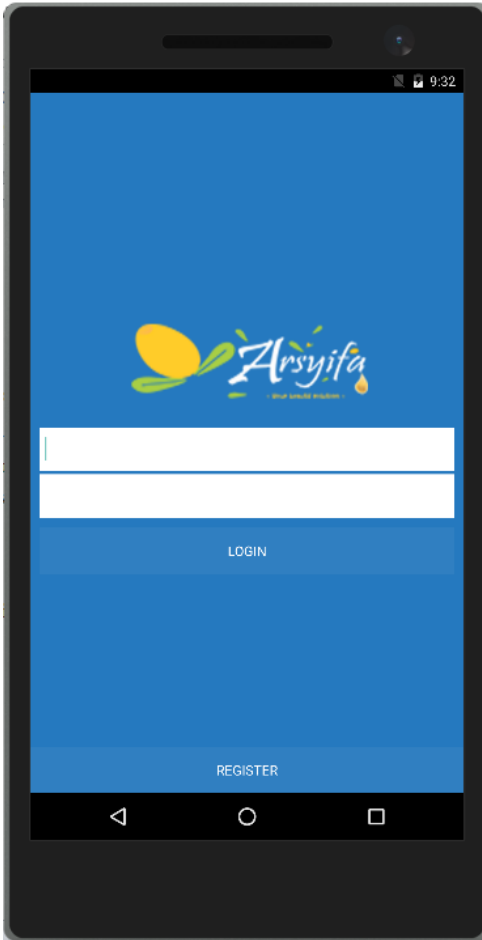
btnRegister.Click += delegate
{
    StartActivity(typeof(RegisterActivity));
};
```

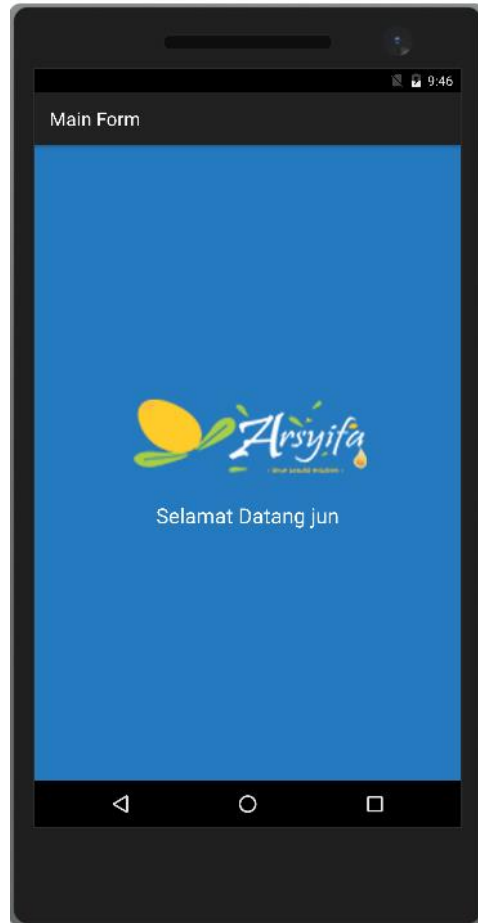
Lalu buat sintaks berikut yang berfungsi pada button login

```
private void btnLogin_Click(object sender, EventArgs e)
{
    try
    {
        var blogin = userdb.CheckLogin(txtusername.Text, txtPassword.Text);

        if (blogin)
        {
            var mainAct = new Intent(this, typeof(MainActivity));
            mainAct.PutExtra("UserWelcome", "Selamat Datang " +
txtusername.Text);
            StartActivity(mainAct);
        }
        else
        {
            Toast.MakeText(this, userdb.Message, ToastLength.Short).Show();
        }
    }
    catch (Exception ex)
    {
        Toast.MakeText(this, ex.ToString(), ToastLength.Short).Show();
    }
}
```

Jalankan Program. Dan liat hasilnya





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.co.id/2017/01/membuat-aplikasi-android-login-screen.html>

Referensi

1. www.msdn.microsoft.com
2. www.planetsourcecode.com
3. www.codeproject.com
4. www.aspnet.com

Masih banyak lagi referensi yang ada di Internet. Anda tinggal cari di www.Google.com. Dengan kata kunci “**tutorial VB.Net**”

Biografi Penulis.



Junindar Lahir di Tanjung Pinang, 21 Juni 1982. Menyelesaikan Program S1 pada jurusan Teknik Inscrenatika 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**”.