

1. Android Studio Kurulumu

- Windows işletim sistemi için <https://developer.android.com/studio/> adresinden android-studio-ide-181.5056338-windows.exe isimli dosyayı direk indirebilirsiniz.
- Linux işletim sistemi için <https://developer.android.com/studio/> adresinden android-studio-ide-181.5056338-linux.zip isimli dosyayı indirip aşağıdaki adımları takip ediniz:

a. Öncelikle aşağıdaki kütüphanelerin bilgisayarınızda yüklü olduğuna emin olunuz.

```
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386
```

b. İlgili kütüphaneler yüklendikten sonra aşağıdaki adımları takip ediniz.

```
betul@betul-VirtualBox:~$ cd /home/betul/Downloads
betul@betul-VirtualBox:~/Downloads$ cd ..
betul@betul-VirtualBox:~$ sudo mv /home/betul/Downloads/android-studio
/usr/local/
[sudo] password for betul:
betul@betul-VirtualBox:~$ cd /usr/local/android-studio/bin/
betul@betul-VirtualBox:/usr/local/android-studio/bin$ ./studio.sh
```

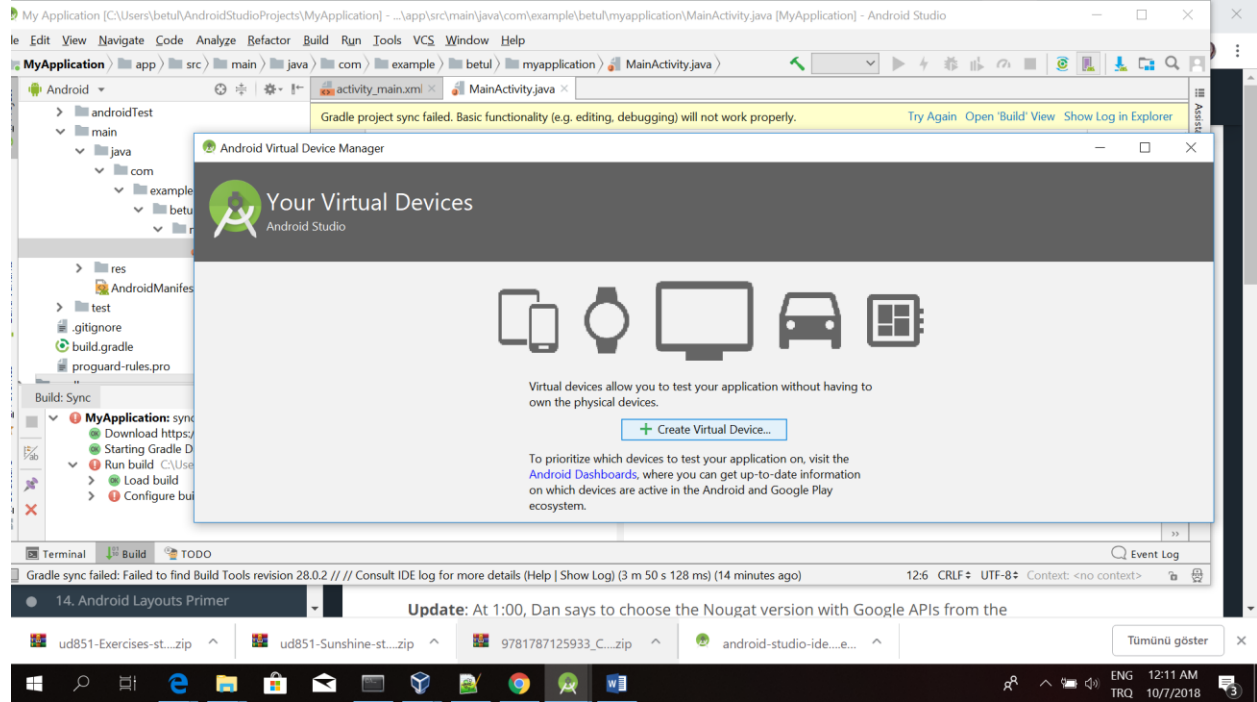
2. Android Studio Giriş ve Sanal Cihaz Kurulumu

Android Studio, Android applicationlar (app) için kullanılan bir IDE. IDE, bir code etidörü ve framework'tür. Android'in temeli hardwares, drivers, power management gibi düşük seviyeli taskleri işleyen bir linux kerneldir. Linux kernel üzerinde bazı core C/C++ kütüphaneleri ve bu core kütüphaneler ile birlikte android runtime vardır. Application framework'ta sağlanan servisler ve classları kullanarak run time'ın kendi instance'ınız üzerinde app'leriniz çalışır. En üstte application layer vardır kendi app'lerimiz ve diğer yüklenen app'ler buradadır. Apps android framework ile, ve birbirleri ile çalışan bileşenler topluluğudur. Apps oluşturmak için dört component vardır:

- Activities
- Services
- Broadcast receivers
- Content providers.



Android studio üzerinde koştuğunda kodunuz, ilk olarak device üzerinde run time içinde çalışabilen byte koduna derlenir. Android studio bunu Grandle kullanarak gerçekleştirir. Grandle, dependencies' leri yöneten bir build tool kitidir. Android Studio da projemizi derledikten sonra uygulamamızı ekran (Screen) üzerinde görebilmek için ilk olarak bir emulator AVD (sanal cihaz-virtual device) oluşturmamız gerekmektedir.



Bu ders için Nexus 5X cihazını tercih ediyoruz. Emulator sadece simulator değil, tüm Android stack çalıştırır. Farklı hardware ya da software üzerinde uygulamanızı debug edebilirsiniz. Device üzerinde çalışmak çok basittir.

Settings-about the device-build number a üç kez tıklıyorsun. Developer seçenekleri görünecek. Bir donanım üzerinde çalışmak için aşağıdaki adresi takip ediniz:

<https://developer.android.com/studio/run/device>

View Nedir?

Bir görüntü, metin ya da bir buton gibi bazı contentleri içeren screen üzerinde bir diktörge. Bu viewler bizim screen layout dediğimiz şeyi oluşturur. App içinde gördüğün her şey user interface – UI olarak adlandırılır. Pek çok view çeşidi vardır: Textview, imageview ve buton. Her bir app ayrı viewlere bölünür:



Kısaca bir view, kendi layout'umuzu oluşturmak için kullandığımız temel bir bloktur.

IDE nedir?

Integrated Development Environment. Kodumuzu yazmak için kullandığımız bir yer (space).

XML nedir?

Extensible Markup Language. Android app nasıl görüneceğini tanımlamanın bir yolu. Bir şeyler yerleştireceğimiz layoutları oluşturmak için kullanıyoruz. (Bilgisayarla konuşmak için belirli bir dil kullanmak zorundayız!)

Not: Bu uygulama dersi için takip edilecek kaynaklara aşağıdaki adresten ulaşabilirsiniz.

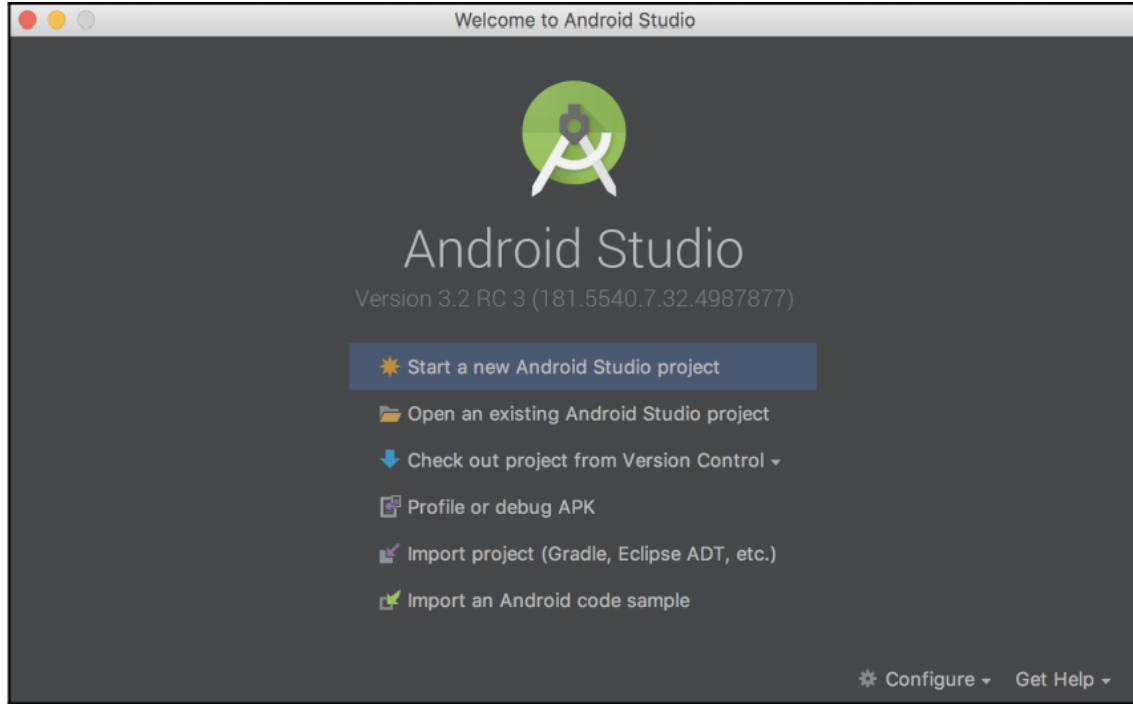
<https://github.com/betulays/AndroidStudioLectureNotes>

3. Aktivite (Activity)

Android SDK, mobil cihazları programlamak için güçlü bir araç sunar ve böyle bir aracı yönetmenin en iyi yolu uygulamalara doğrudan hakim olmaktır. Activity sınıfı, uygulama ve ekran arasında arayüz sağladığından, çoğu Android uygulamasının temel yapı taşıdır. Çoğu Android uygulamasında, birkaç tane olmasa bile en az bir activity olacaktır (ancak gerekli değildir). Bir arka plan servis uygulaması, kullanıcı arayüzü yoksa bir activity gerektirmez.

Uygulama

Aktiviteler ve diğer uygulama bileşenleri (servisler gibi) AndroidManifest.xml dosyasında bulunmaktadır. İlk örneğimiz için öncelikle yeni bir proje oluşturacağız:



- 1- **Start a new Android Studio Project** : Yeni bir Android Studio Projeye başlamak için. Biz bu seçenekle devam edeceğiz.
- 2- **Open an existing Android Studio Project** : Önceden oluşturulmuş bir Android Studio projesini açmak için.
- 3- **Check out project from Version Control** : Git , svn gibi herhangi bir version control sisteminden çekeceksek bu seçenekle devam ediyoruz.
- 4- **Import project (Eclipse ADT , Gradle ,etc..)** : Farklı bir ide yada farklı bir build sistemi ile yazılmış bir projeyi Android Studio projesine çevirerek açmak için. Özellikle önceden yazılmış bir **eclipse projesini açmak** için kullanılır.
- 5- **Import an Android code sample** : Android'in bize sunmuş olduğu hazır kodları açmak için.

Create New Project

Create Android Project

Application name
ActivityExp

Company domain
betul.example.com

Project location
C:\Users\betul\ActivityExp2

Package name
com.example.betul.activityexp Edit

☐ Include C++ support
☐ Include Kotlin support

Previous Next Cancel Finish

Bu ekranda uygulama adımızı ve paket adımızı giriyoruz:

Application Name: Uygulama Adımız (Türkçe karakter kullanılmaması önerilir). Cihazlarda ikonların altında gözükecektir.

Company Domain: Uygulamayı geliştirdiğiniz şirketin domain adresi.

Package Name : Company domain ve package name'e göre otomatik oluşur. Değiştirilmek istenirse sağ tarafındaki edit ile değiştirilebilir. Package name Google Play'deki unique id'niz gibi düşünebilirsiniz. Aynı package name ile başka bir uygulama yüklenemez.

Project Location: Projemizin oluşturulacağı klasör.

Gerekli alanları doldurduktan sonra Next diyerek devam ediyoruz.

Create New Project

Target Android Devices

Select the form factors and minimum SDK
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**
API 15: Android 4.0.3 (IceCreamSandwich)
By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)
☐ Include Android Instant App support

☐ **Wear OS**
API 23: Android 6.0 (Marshmallow)

☐ **TV**
API 21: Android 5.0 (Lollipop)

☐ **Android Auto**

☐ **Android Things**
API 24: Android 7.0 (Nougat)

Previous

Next

Cancel

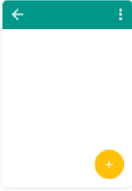
Finish

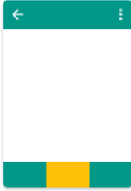
Bu ekranda uygulamamızın en düşük hangi Android sürümünü destekleyeceğini seçiyoruz. Diğer seçenekler (Wear OS, TV...) ile ilgili uygulama yazmayacağımız için next diyerek ilerliyoruz.

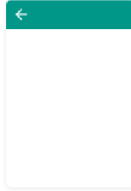
Create New Project

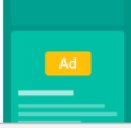
Add an Activity to Mobile

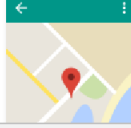
Add No Activity


Basic Activity


Bottom Navigation Activity


Empty Activity



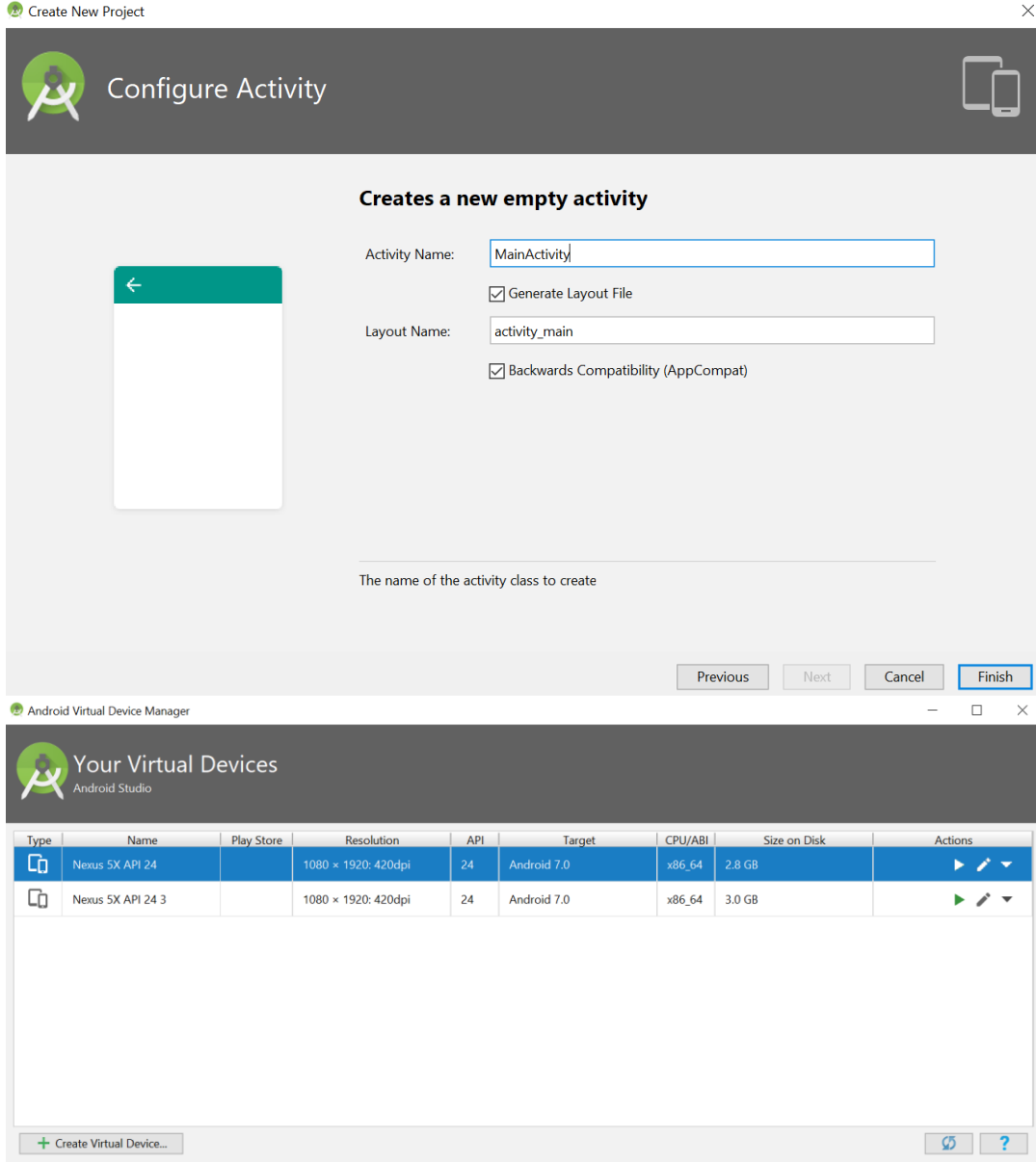


Previous

Next

Cancel

Finish



Yeni projeniz oluşturulduktan sonra aşağıdaki adımları takip edeceğiz:

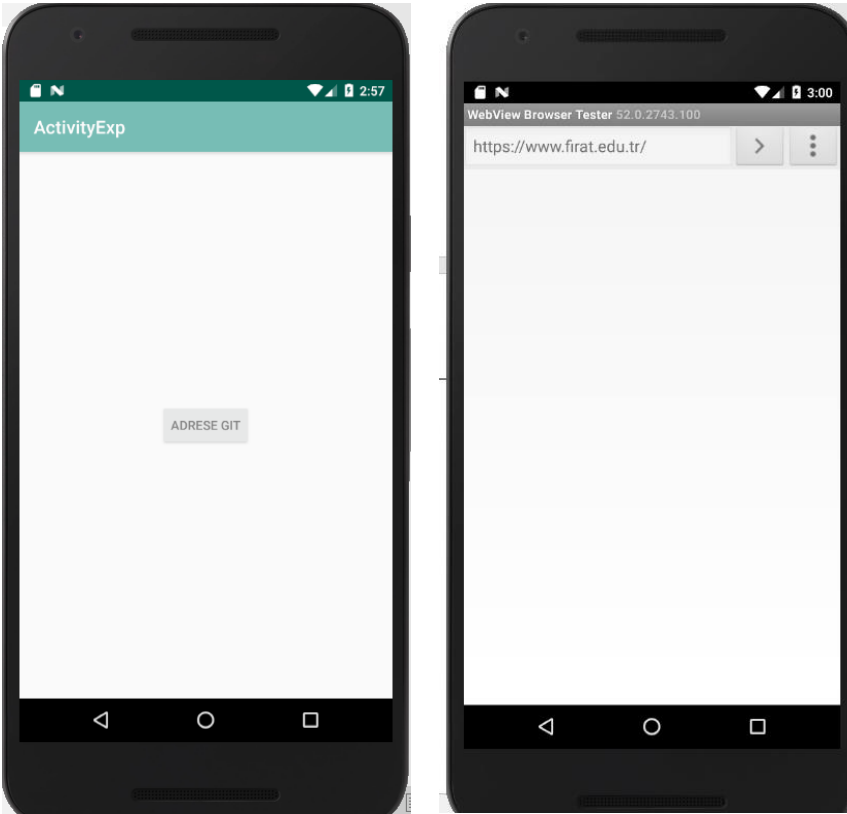
1. MainActivity.java dosyasını açınız ve aşağıdaki fonksiyonu ekleyin.

```
public void launchIntent(View view) {  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setData(Uri.parse("https://www.firat.edu.tr/"));  
    startActivity(intent);  
}
```

2. Activity_main.xml dosyasınız açınız ve <TextView /> bloğu yerine aşağıdaki XML kodunu ekleyin.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Adrese Git"
    android:id="@+id/button"
    android:onClick="launchIntent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
```

3. Uygulamayı çalıştırın ve sanal cihazınızı seçin, aşağıdaki ekran görüntüsünü alacaksınız.



4. Aktivite Değişimi

Bu uygulamada bir aktiviteyi bir diğer aktiviteden etkinleştireceğiz. Bu işlem için iki tane aktivite sınıfı oluşturmamız gerekmektedir. Bir aktiviteden diğerine geçmek için de buton kullanacağız.

Uygulama

Yeni bir activity oluşturulduktan sonra aşağıdaki adımları takip edeceğiz:

1. MainActivity.java dosyasını açınız ve aşağıdaki fonksiyonu ekleyin.

```
public void onClickSwitchActivity(View view) {  
    Intent intent = new Intent(this, SecondActivity.class);  
    startActivity(intent);  
}
```

2. Activity_main.xml (res/layout klasörü altında) dosyasınız açınız ve <TextView /> bloğu yerine aşağıdaki XML kodunu ekleyin.

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerVertical="true"  
    android:layout_centerHorizontal="true"  
    android:text="Activity 2"  
    android:onClick="onClickSwitchActivity"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent"/>
```

3. SecondActivity.java dosyasını açınız ve aşağıdaki fonksiyonu ekleyin.

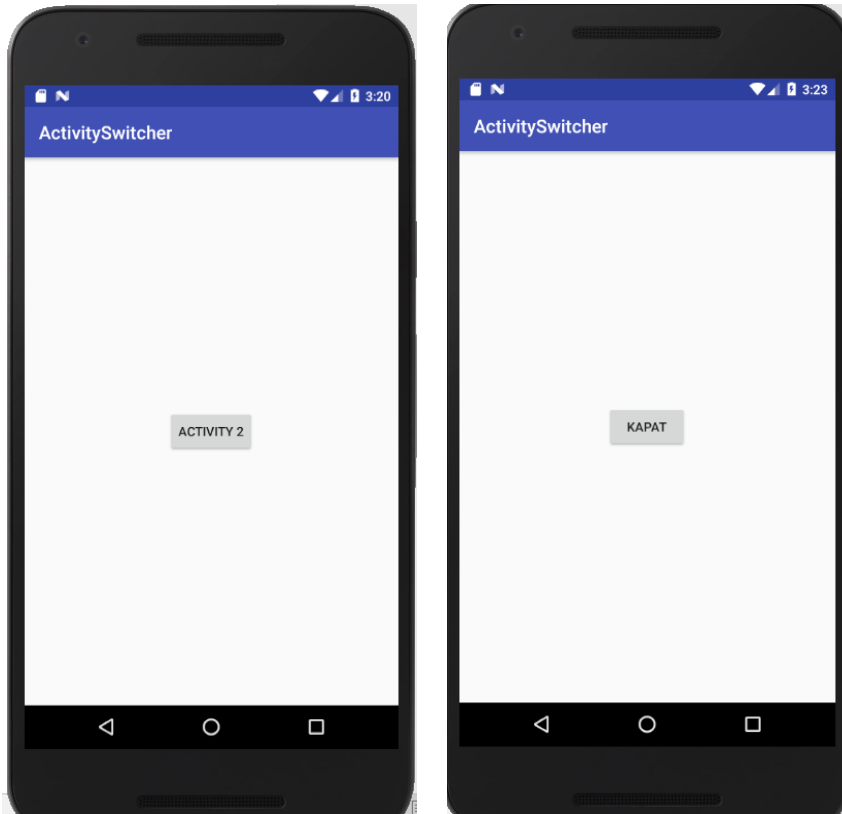
```
public void onClickClose(View view) {  
    finish();  
}
```

4. Activity_second.xml (res/layout klasörü altında) dosyasınız açınız ve <TextView /> bloğu yerine aşağıdaki XML kodunu ekleyin.

```
<Button
```

```
android:id="@+id/buttonClose"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Kapat"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"
android:onClick="onClickClose"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"/>
```

5. Uygulamayı çalıştırın ve sanal cihazınızı seçin, aşağıdaki ekran görüntüsünü alacaksınız.



5. Veri Gönderme

Intent nesnesi bir mesajlaşma nesnesidir. Bir mesaj nesnesinin amacı uygulamanın diğer bileşenleriyle iletişime geçmesidir. Bu amaçla, intent nesnesi ile bir bilginin ya da verinin bir aktiviteden diğerine geçişi nasıl olur bu uygulamada göreceğiz.

Uygulama

Yeni bir activity oluşturulduktan sonra aşağıdaki adımları takip edeceğiz:

1. MainActivity.java dosyasını açınız ve aşağıdaki fonksiyonu ekleyin.

```
public void onClickSwitchActivity(View view) {  
    EditText editText = (EditText)findViewById(R.id.editTextData);  
    String text = editText.getText().toString();  
    Intent intent = new Intent(this, SecondActivity.class);  
    intent.putExtra(Intent.EXTRA_TEXT,text);  
    startActivity(intent);  
}
```

2. Activity_main.xml (res/layout klasörü altında) dosyasınız açınız ve bir önceki button kodununun üstüne aşağıdaki XML kodunu ekleyiniz.

```
<EditText  
    android:id="@+id/editTextData"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintBottom_toTopOf="@+id/button" />
```

```
<Button  
  
    android:id="@+id/button"  
  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:layout_centerVertical="true"  
  
    android:layout_centerHorizontal="true"  
  
    android:text="Activity 2"  
  
    android:onClick="onClickSwitchActivity"  
  
    app:layout_constraintBottom_toBottomOf="parent"  
  
    app:layout_constraintLeft_toLeftOf="parent"
```

```
app:layout_constraintRight_toRightOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"/>
```

3. SecondActivity.java dosyasını açınız ve aşağıdaki fonksiyonu ekleyin.

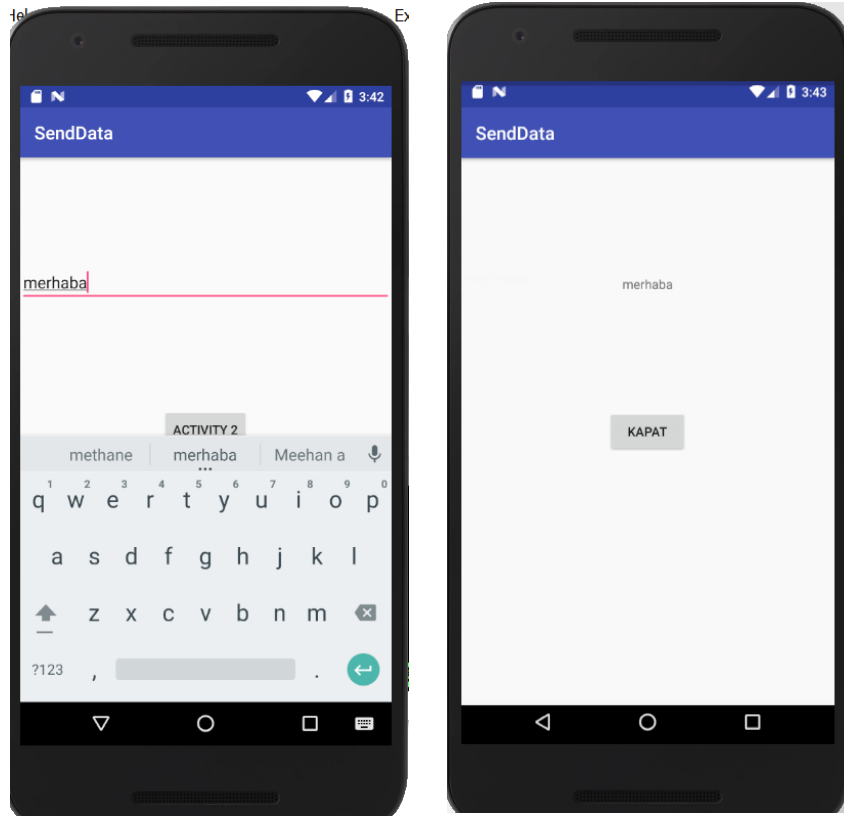
```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_second);  
    TextView textView = (TextView) findViewById(R.id.textViewText);  
    if (getIntent() != null &&  
        getIntent().hasExtra(Intent.EXTRA_TEXT)) {  
        textView.setText(getIntent().getStringExtra(Intent.EXTRA_TEXT));  
    }  
}
```

4. Activity_second.xml (res/layout klasörü altında) dosyasınız açınız ve aşağıdaki XML kodunu ekleyin.

```
<TextView  
    android:id="@+id/textViewText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintBottom_toTopOf="@id/buttonClose"/>
```

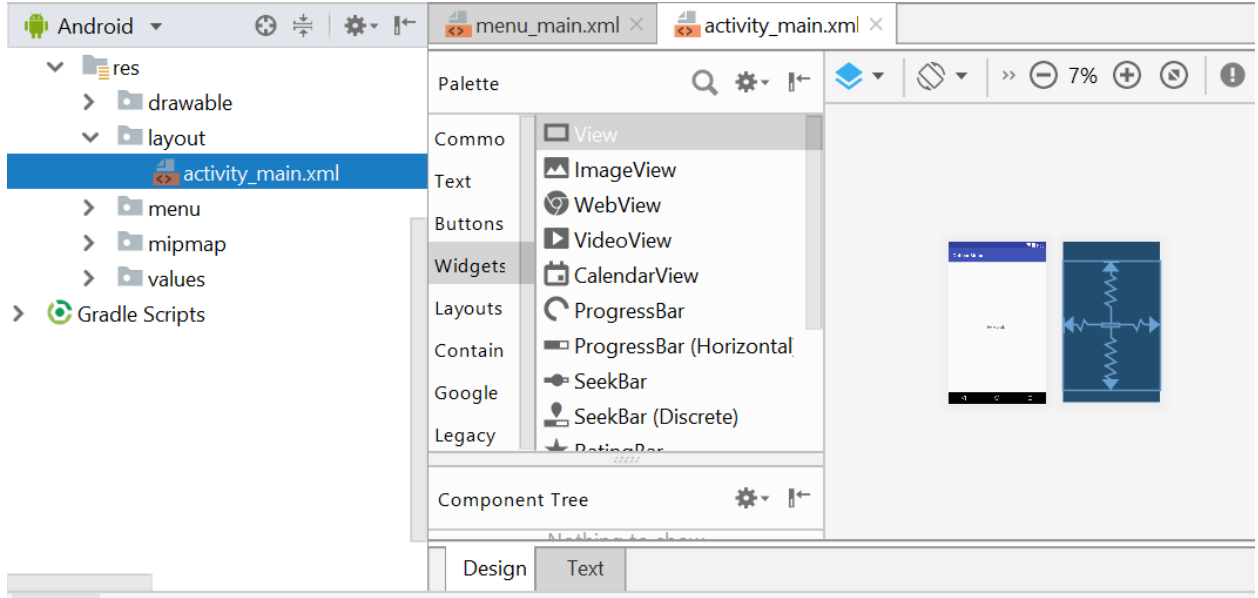
```
<Button  
  
    android:id="@+id/buttonClose"  
  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:text="Kapat"  
  
    android:layout_centerVertical="true"  
  
    android:layout_centerHorizontal="true"  
  
    android:onClick="onClickClose"  
  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent"/>
```

5. Uygulamayı çalıştırın ve sanal cihazınızı seçin, aşağıdaki ekran görüntüsünü alacaksınız.



6. Widgets

Kullanıcı arayüzü Layout içinde tanımlanmaktadır. Bir layout XML içinde ya da kod içerisinde dinamik olarak oluşturulur. Layout dosyalarında özel views'ler ise widget olarak adlandırılır. Android SDK tarafından sağlanan widget listesini görmek için bir layout dosyası açınız ve Design sekmesine tıklayınız.

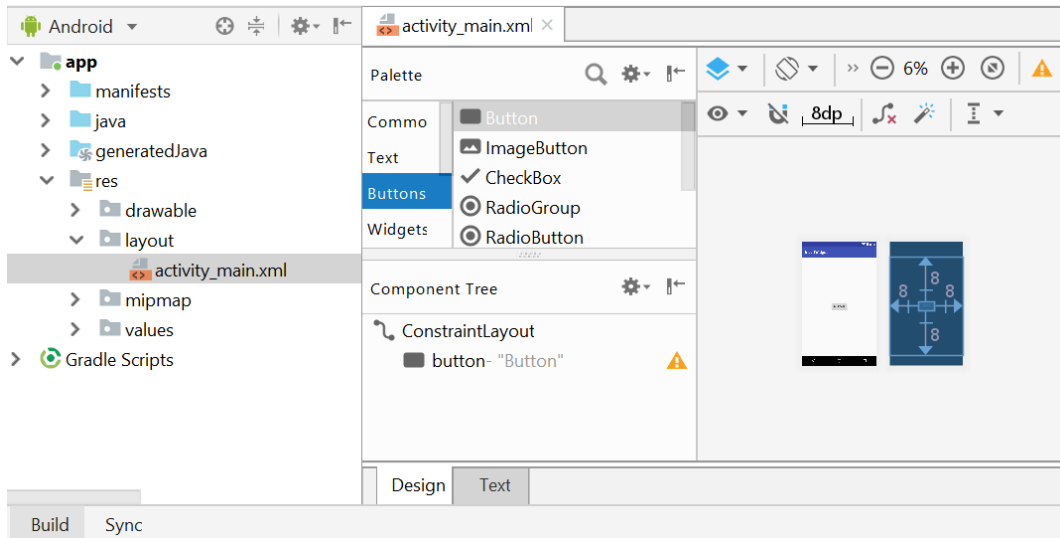


Text, Buttons gibi sekmeler widget içinde yer almasa da tanım olarak widget içerisinde yer almaktadır.

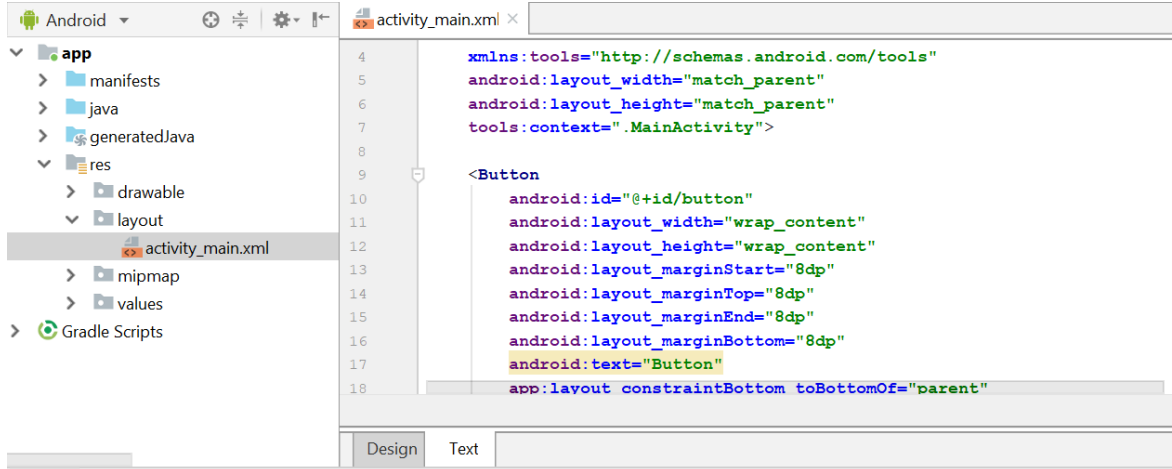
Uygulama

Widget'lar bir layout dosyası içerisinde tanımlanmaktadır. Bu uygulamada Android Studio Designer ile bir buton oluşturacağız. Buton oluşturduktan sonra, `onClickListener()` kullanarak buton click events'leri alacak bir method oluşturacağız.

1. Activity_main.xml dosyasını açınız ve Design sekmesine tıklayınız. Widget listesinden Button sekmesini bulunuz ve sağdaki activity ekranının merkezine sürükleyiniz.



2. Text sekmesine tıkladığınızda Button için bir XML kodunun default olarak oluştuğunu göreceksiniz:



3. MainActivity.java dosyasını açın ve aşağıdaki kodu ekleyiniz.

```
Button button = (Button)findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(MainActivity.this,"Clicked",
            Toast.LENGTH_SHORT).show();
    }
});
```

4. Uygulamayı çalıştırınız...

