



Perbaikan Aplikasi: Text Input

www.esaunggul.ac.id

Validasi Perangkat Lunak Mobile (CRI-562)
Pertemuan 7

Dosen Pengampu: Harry Kurniawan
Prodi Teknik Informatika - Fakultas Ilmu Komputer

Tujuan

- Melakukan perbaikan aplikasi yang sudah teridentifikasi memiliki kesalahan
- Target awal perbaikan adalah memperbaiki hasil validasi **Teks Input**

Persiapan

Persiapan

- Source Code Aplikasi
 - Source code kedua aplikasi yang divalidasi
- Hardware
 - Handphone dengan OS Android
- Software
 - Driver USB perangkat android sesuai handphone
 - Android Studio

Langkah Kerja

Langkah Kerja

1. Download aplikasi dalam ekstensi .APK.
 - Aplikasi harus sesuai dengan telah didaftarkan untuk divalidasi.
 - Setiap Mahasiswa wajib memiliki 2 Aplikasi
 - Aplikasi dibawah top 100 Play Store

Langkah Kerja

2. Ekstrak APK sehingga mendapatkan source codenya

– Tool: www.javadecompilers.com/apk

Langkah Kerja

2. Ekstrak APK sehingga mendapatkan source codenya

– Tool: www.javadecompilers.com/apk

Langkah Kerja

Decompilers online

Android Apk decompiler

Java decompilers

APK decompiler

ApkTool online

Download Jad

AdMob by Google [Learn more](#)

[Decompile Apk and Dex Android files to Java](#)

Choose File No file chosen

Upload and Decompile [G+](#)

[Twitter](#) [Facebook](#) [Google+](#) [Stumbleupon](#) [LinkedIn](#)

Select a decompiler

Jadx decompiler for Android

FBS \$123

BONUS SELAMAT DATANG

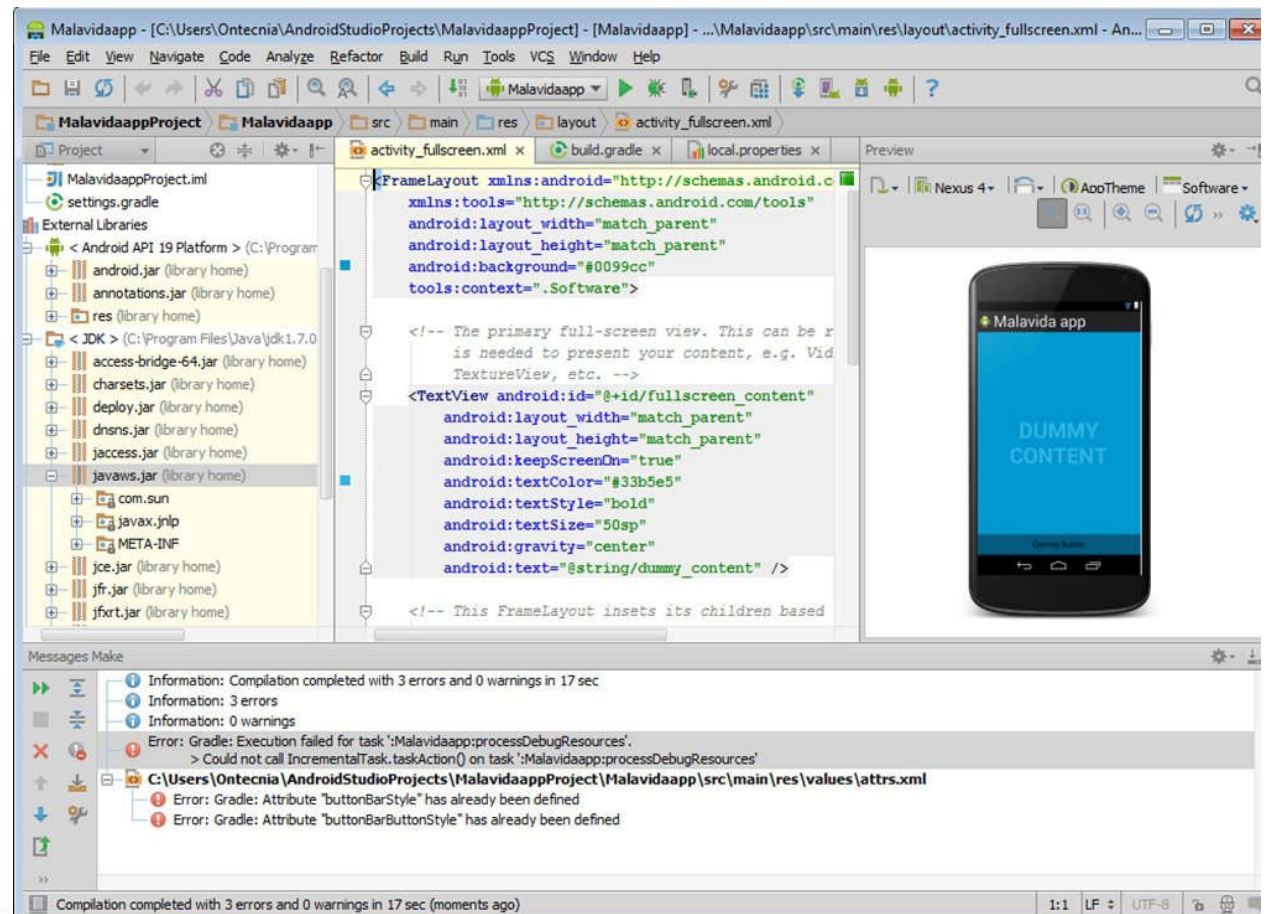
MULAI TRADING

TRADING TANPA DEPOSIT DAN MENGHASILKAN PROFIT

Langkah Kerja

3. Install Android Studio

<https://developer.android.com/studio/index.html>



Langkah Kerja

4. Jalankan Android Studio. Masalah yang sering dihadapi ketika Android Studio tidak bisa dijalankan:

- Konfigurasi letak SDK & JDK
- Konfigurasi Environment

Tutorial:

<https://stackoverflow.com/questions/36638657/run-android-studio-with-jre-1-8-or-later-version>

Langkah Kerja

The screenshot shows an IDE interface with the following components:

- Toolbar:** Contains various icons for file operations, navigation, and execution.
- Project Explorer (Left):** Shows the project structure:
 - Android
 - app
 - manifests
 - java
 - com.company.my_app
 - activities
 - HomeActivity
 - PlaceActivity
 - components
 - dataModels
 - elements
 - lib
 - Application
 - res
 - layout
 - home.xml
 - list_tile1.xml
 - list_tile11.xml
 - place.xml
 - menu
 - values
 - Gradle Scripts

- Editor (Right):** Displays the code for `HomeActivity.java`:


```
package com.company.my_app.activities;

import ...

/**
 * The Activity for the page: Home. This page has the following elements:
 * TextField1, TextButton1, List1; and the following components: StatusBar,
 * AppBar.
 */
public class HomeActivity extends AppCompatActivity implements DropShadowViewProvider {
    /**
     * This page's layout container
     */
    LinearLayout home;

    /**
     * This page's current variant
     */
    private String variant;

    public DSAppBar appBar;

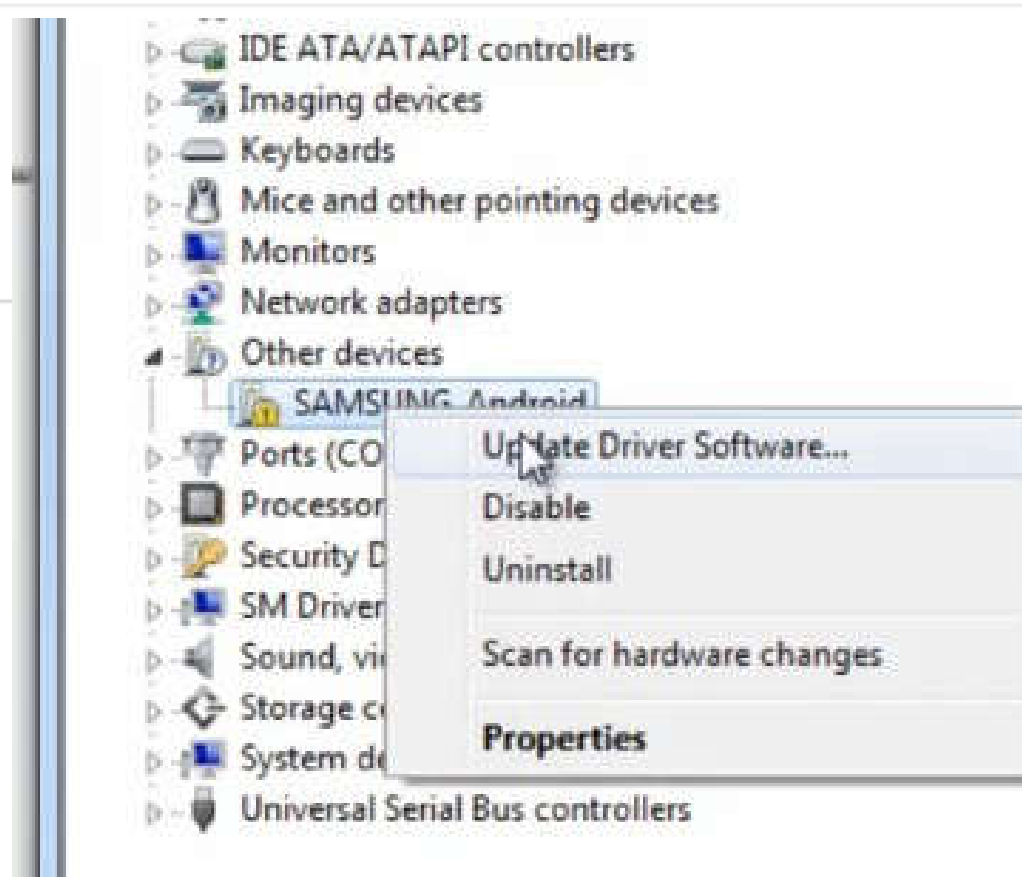
    /**
     * {@link android.support.v4.app.FragmentActivity#onCreate(Bundle savedInstanceState)}
     * If the activity is being re-initialized from a previous state,
     * this Bundle contains the data it most recently saved in
     * {@link #onSaveInstanceState}. <b><i>{@link #onSaveInstanceState}</i></b>
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

Langkah Kerja

5. Sambungkan handphone ke PC dengan kabel data dan Install driver handphone di PC

- Tutorial: <https://www.recovery-android.com/install-android-usb-driver-on-computer.html>
- *Driver juga dapat didownload di website resmi produsen masing-masing handphone*

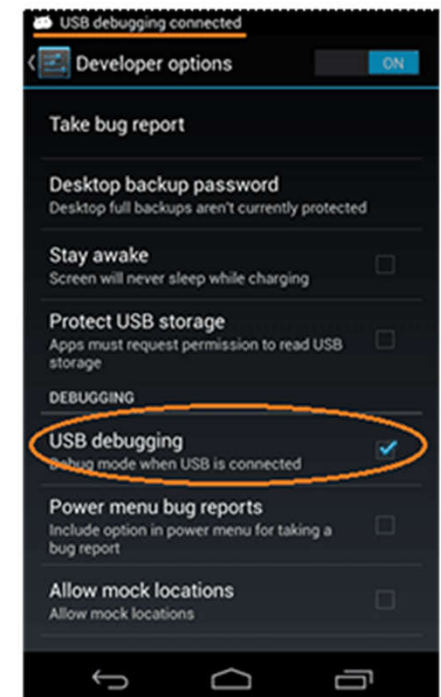
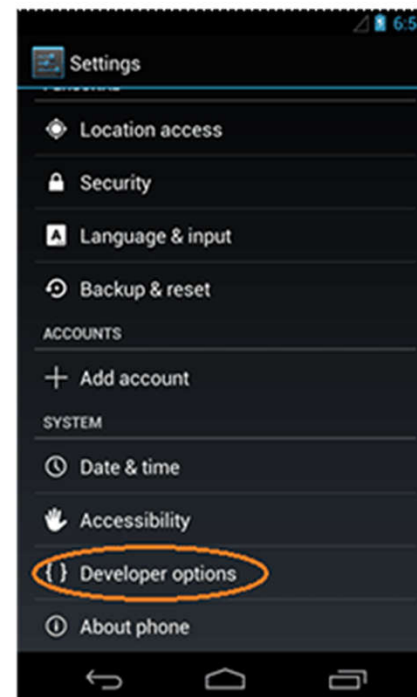
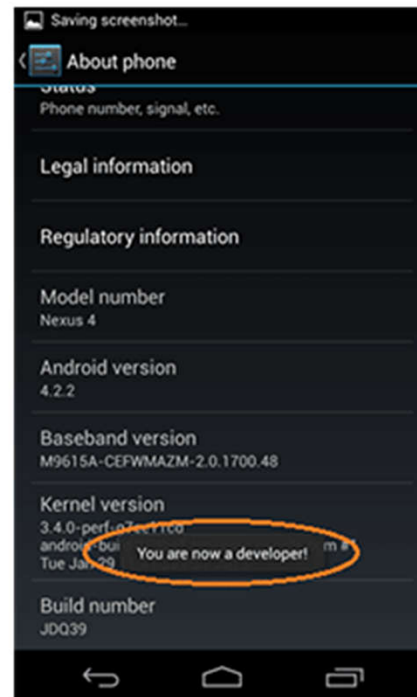
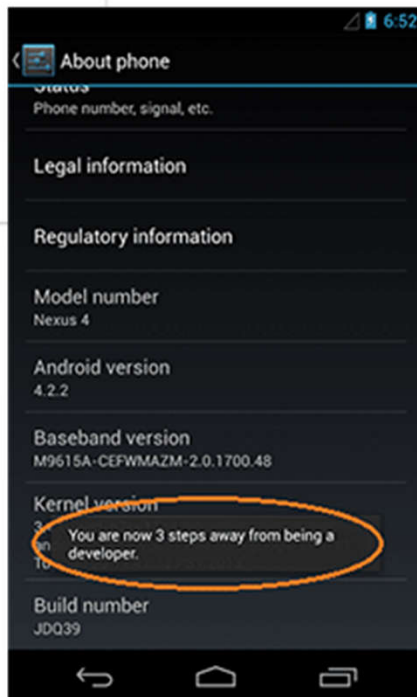
Langkah Kerja



Langkah Kerja

6. Aktifkan “developer mode” di Android anda.
 - Tutorial: <https://www.kingoapp.com/root-tutorials/how-to-enable-usb-debugging-mode-on-android.htm>

Langkah Kerja



Langkah Kerja



6. Sekarang jalankan Android Studio dan coba membuat aplikasi pertama

- Referensi: [Android-developer-fundamentals-course-practicals-idn.pdf](#)
- Tersedia di google drive dalam folder Mini Praktikum

Tugas

Tugas

- Mahasiswa melakukan perbaikan pada Teks Edit yang dianggap masih salah atau kurang optimum, sesuai dengan laporan validasi yang sudah dilakukan oleh mahasiswa.
- Wajib melaporkan hasil perbaikan dengan menunjukkan User Interface (UI) & source code sebelum dan sesudah perubahan
- Tugas ini dapat diabaikan jika Text Input di kedua aplikasi sudah benar atau tidak memiliki Text Input, namun wajib melaporkan bukti UI dan source code bahwa aplikasi sudah benar atau tidak memiliki teks input.

Tugas

Contoh: APLIKASI 1: Cilukba!

(level 1)	(level 2)	Jenis Input	Benar? (Y/T)	Catatan
Log In	Password	Teks	T	Seharusnya input dilakukan masking sehingga password tidak terlihat

The screenshot displays the Android Studio interface for editing the `content_main.xml` layout file. The breadcrumb path at the top is `MyApplication2 > app > src > main > res > layout > content_main.xml`. The toolbar shows the current theme is `Custom` for an `Android 26` device with `NoActionBar`.

The **Project** view on the left shows the following structure:

- app
 - manifests
 - AndroidManifest.xml
 - java
 - res
 - drawable
 - layout** (highlighted with a red box)
 - activity_display_message.xml
 - activity_main.xml
 - content_main.xml
 - menu
 - menu_main.xml
 - mipmap
 - ic_launcher.png (5)
 - ic_launcher_round.png (5)
 - values
 - Gradle Scripts
 - build.gradle (Project: MyApplication2)
 - build.gradle (Module: app)
 - gradle-wrapper.properties (Gradle Version)
 - proguard-rules.pro (ProGuard Rules for app)
 - gradle.properties (Project Properties)
 - settings.gradle (Project Settings)
 - local.properties (SDK Location)

The **Widget Palette** in the center lists various UI components. Under the **Text** category, the **Password** widget is selected. Other visible widgets include Plain Text, Password (Numeric), E-mail, Phone, Postal Address, Multiline Text, Time, Date, Number, Number (Signed), Number (Decimal), autoCompleteTextView, and MultiAutoCompleteTextView.

The **Component Tree** at the bottom shows the hierarchy of the selected widget:

- ConstraintLayout
 - textView - "@string/edit_message"
 - button - "@string/button_send"
 - editText
 - editText2
 - autoCompleteTextView - "AutoCompleteTe"
 - editText3
 - editText4

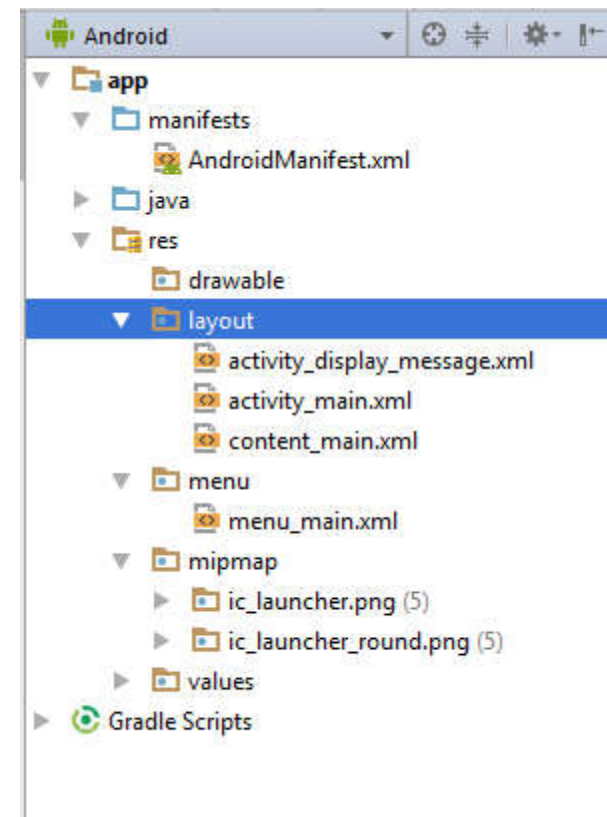
The **Design** view on the right shows a preview of the layout on a mobile device screen, with a blue background and several text input fields.

Tugas

Contoh: APLIKASI 1: Cilukba!

Direktori:

App/res/layout/activity_main.xml



MyApplication2 | app | src | main | res | layout | content_main.xml

Android | DisplayMessageActivity.java | AndroidManifest.xml | content_main.xml | strings.xml

Project Structure: app, manifests, java, res, layout, menu, mipmap, values, Gradle Scripts

Palette: All, Widgets, Text, Layouts, Containers, Images, Date, Transitions, Advanced, Google, Design, AppCompat

Component Tree: ConstraintLayout, textView, button, editText, editText2, autoCompleteTextView, editText3, **editText4**

The screenshot shows the Android Studio interface. The top bar displays the project name 'MyApplication2' and the current file 'content_main.xml'. The left sidebar shows the project structure, including the 'res/layout' directory. The main workspace is in 'Design' mode, showing a mobile app layout with a 'Password' field highlighted by a red rectangle. The 'Component Tree' on the right shows a 'ConstraintLayout' containing several widgets, with 'editText4' highlighted by a red rectangle. The 'Palette' on the left shows the 'Text' category selected, with 'Password' and 'Password (Numeric)' visible.

The screenshot shows the Android Studio IDE interface. The top toolbar indicates the current view is 'Text'. The 'Palette' on the left lists various widget categories, with 'Text' selected. The 'Component Tree' on the right shows a hierarchy of views, with 'editText4' selected. The main editor area displays a visual representation of the widget in text mode, showing its internal structure and constraints. A red arrow points from the 'Design' tab to the 'Text' tab, indicating the transition.

MyApplication2 | app | src | main | res | layout | content_main.xml

Android | DisplayMessageActivity.java | AndroidManifest.xml | content_main.xml | strings.xml

Project Structure: app, manifests, java, res, layout, menu, mipmap, values, Gradle Scripts

Palette: All, Widgets, Text, Layouts, Containers, Images, Date, Transitions, Advanced, Google, Design, AppCompat

Component Tree: ConstraintLayout, textView, button, editText, editText2, autoCompleteTextView, editText3, editText4

Design | Text

Pindah dari tampilan design ke text

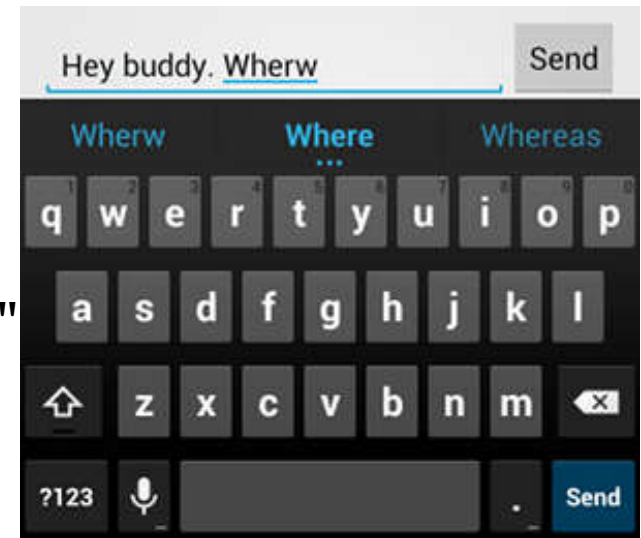
Tugas

Contoh: APLIKASI 1: Cilukba!

```
<EditText
```

```
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:inputType="textPersonName"
```

```
/>
```



Tugas

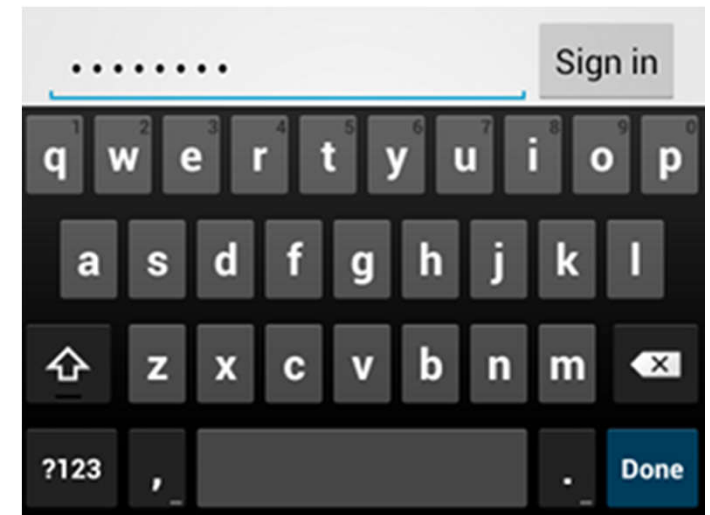
Contoh: APLIKASI 1: Cilukba!

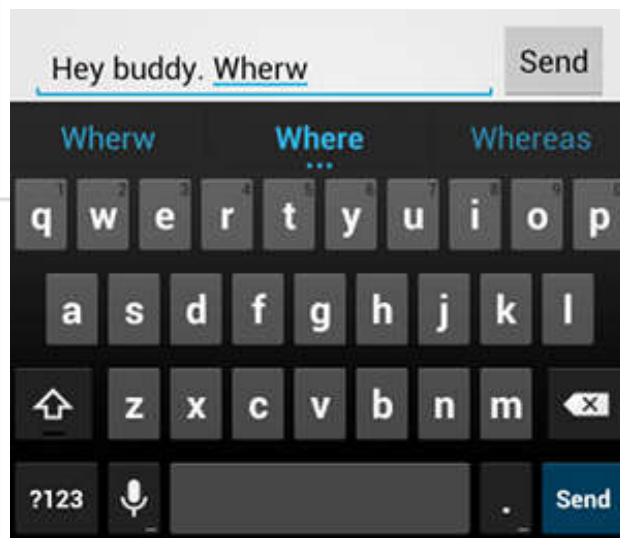
Ganti input text yang tadinya
`textPersonName` menjadi
`textPassword`

Tugas

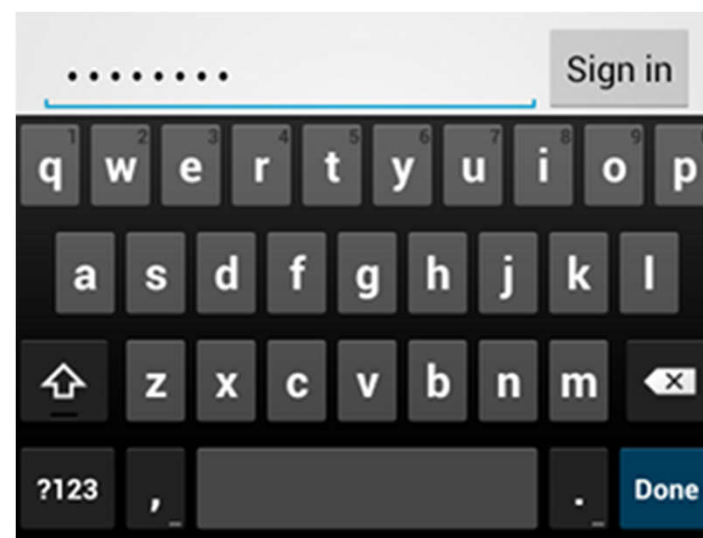
Contoh: APLIKASI 1: Cilukba!

```
<EditText  
    android:id="@+id/editText3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="Password"  
    android:inputType="textPassword"  
>
```





Sebelum



Sesudah

Tutorial

- Referensi untuk teknis detail coding android
 - Website:
<https://developer.android.com/training/index.html>
 - Buku:
<https://www.gitbook.com/download/pdf/book/google-developer-training/android-developer-fundamentals-course-practicals?lang=idn>