

**MODUL PRAKTIKUM PEMROGRAMAN GAWAI (*MOBILE
PROGRAMMING*) BERBASIS ANDROID**

Diajukan Sebagai Prasyarat Dalam Kegiatan Perkuliahan di Universitas Esa
Unggul Menggunakan Aplikasi Android Studio

Disusun Oleh :

Tri Ismardiko Widyawan, S.Kom, M.Kom

&

Asisten Laboratorium Komputer



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER**

UNIVERSITAS ESA UNGGUL

JAKARTA

2018

Apa itu Android?



Android adalah sumber terbuka dan **Sistem Operasi** berbasis Linux untuk perangkat bergerak seperti smartphone dan komputer tablet. Android dikembangkan oleh *Open Handset Alliance* , yang dipimpin oleh Google, dan perusahaan lain.

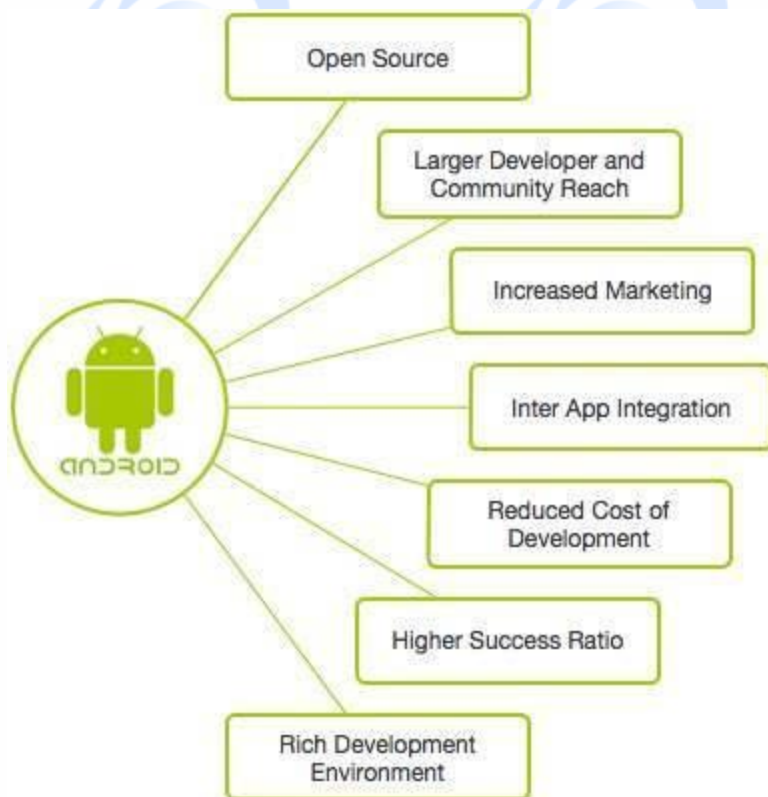
Android menawarkan pendekatan terpadu untuk pengembangan aplikasi untuk perangkat seluler yang berarti pengembang hanya perlu mengembangkan untuk Android, dan aplikasi mereka harus dapat berjalan di berbagai perangkat yang didukung oleh Android.

Versi beta pertama dari Android Software Development Kit (SDK) dirilis oleh Google pada tahun 2007 di mana sebagai versi komersial pertama, Android 1.0, dirilis pada bulan September 2008.

Pada tanggal 27 Juni 2012, di konferensi Google I / O, Google mengumumkan versi Android berikutnya, 4.1 **Jelly Bean** . Jelly Bean adalah pembaruan tambahan, dengan tujuan utama meningkatkan antarmuka pengguna, baik dari segi fungsionalitas dan kinerja.

Kode sumber untuk Android tersedia di bawah lisensi perangkat lunak gratis dan sumber terbuka. Google menerbitkan sebagian besar kode di bawah Lisensi Apache versi 2.0 dan sisanya, kernel Linux berubah, di bawah GNU General Public License versi 2.

Mengapa Android?



Fitur Android

Android adalah sistem operasi yang kuat yang bersaing dengan Apple 4GS dan mendukung fitur-fitur hebat. Beberapa dari mereka tercantum di bawah ini -

Sr.No.	Fitur & Deskripsi
1	UI yang indah Layar dasar OS Android menyediakan antarmuka pengguna yang cantik dan intuitif.
2	Konektivitas GSM / EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC dan WiMAX.
3	Penyimpanan SQLite, database relasional ringan, digunakan untuk tujuan penyimpanan data.

4	<p>Dukungan media</p> <p>H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, dan BMP.</p>
5	<p>Messaging</p> <p>SMS dan MMS</p>
6	<p>Peramban web</p> <p>Berdasarkan mesin tata letak WebKit open-source, dipasangkan dengan mesin JavaScript V8 Chrome yang mendukung HTML5 dan CSS3.</p>
7	<p>Multi-touch</p> <p>Android memiliki dukungan asli untuk multi-touch yang pada awalnya tersedia di handset seperti HTC Hero.</p>
8	<p>Multi-tasking</p> <p>Pengguna dapat melompat dari satu tugas ke tugas lainnya dan pada saat yang sama berbagai aplikasi dapat berjalan secara bersamaan.</p>
9	<p>Widget yang dapat diubah ukurannya</p> <p>Widget dapat diubah ukurannya, sehingga pengguna dapat memperluasnya untuk menampilkan lebih banyak konten atau menyusutkannya untuk menghemat ruang.</p>
10	<p>Banyak bahasa</p> <p>Mendukung arah tunggal dan teks dua arah.</p>
11	<p>GCM</p> <p>Google Cloud Messaging (GCM) adalah layanan yang memungkinkan pengembang mengirim data pesan singkat ke pengguna mereka di perangkat Android, tanpa memerlukan solusi sinkronisasi kepemilikan.</p>

12	<p>Wi-Fi Direct</p> <p>Teknologi yang memungkinkan aplikasi menemukan dan memasang secara langsung, melalui koneksi peer-to-peer bandwidth tinggi.</p>
13	<p>Android Beam</p> <p>Teknologi berbasis NFC populer yang memungkinkan pengguna langsung berbagi, hanya dengan menyentuh dua ponsel yang dilengkapi NFC bersama-sama.</p>

Aplikasi Android

Aplikasi Android biasanya dikembangkan dalam bahasa Java menggunakan Perangkat Pengembangan Perangkat Lunak Android.

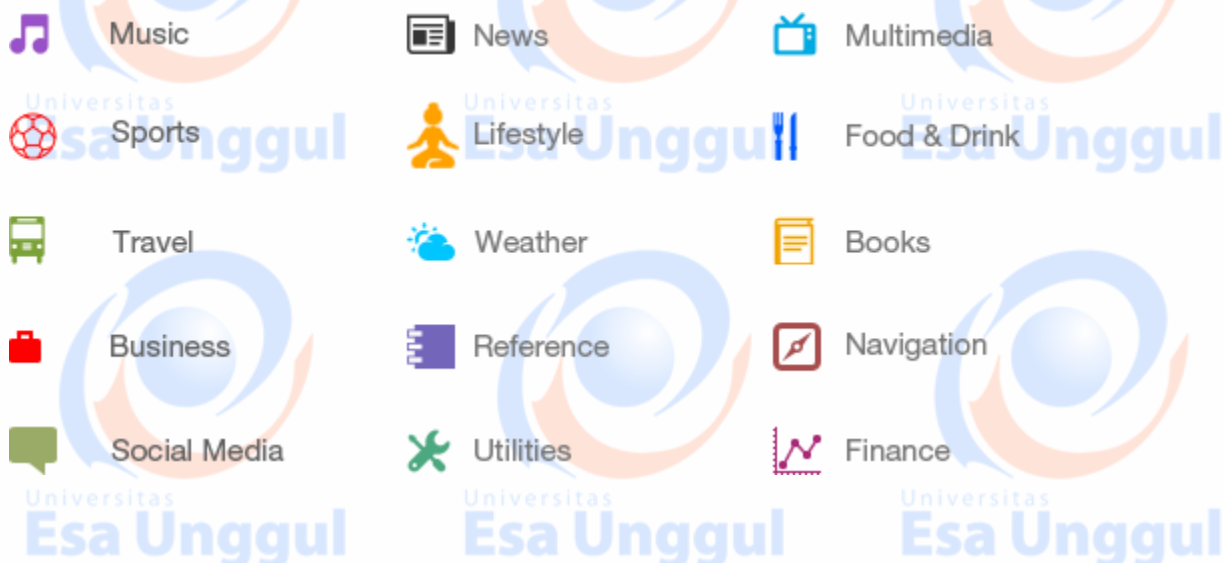
Setelah dikembangkan, aplikasi Android dapat dikemas dengan mudah dan terjual habis baik melalui toko seperti **Google Play**, **SlideME**, **Opera Mobile Store**, **Mobango**, **F-droid** dan **Amazon Appstore**.

Android memberdayakan ratusan juta perangkat seluler di lebih dari 190 negara di seluruh dunia. Ini adalah basis terinstal terbesar dari semua platform seluler dan berkembang pesat. Setiap hari lebih dari 1 juta perangkat Android baru diaktifkan di seluruh dunia.

Tutorial ini telah ditulis dengan tujuan untuk mengajarkan Anda mengembangkan dan mengemas aplikasi Android. Kami akan mulai dari pengaturan lingkungan untuk pemrograman aplikasi Android dan kemudian menelusuri untuk melihat berbagai aspek aplikasi Android.

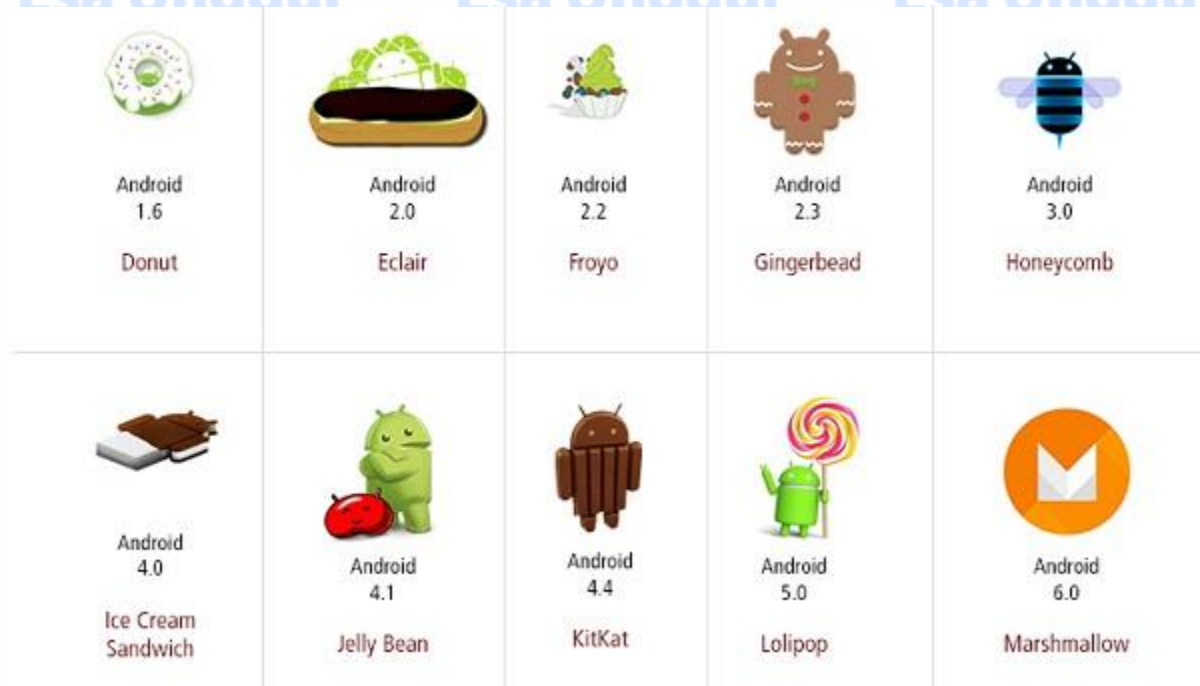
Kategori aplikasi Android

Ada banyak aplikasi android di pasar. Kategori teratas adalah -



Sejarah Android

Nama kode rentang android dari A ke N saat ini, seperti Aestro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, dan Marshmallow. Mari kita memahami sejarah android secara berurutan.



Apakah tingkat API?

Tingkat API adalah nilai integer yang secara unik mengidentifikasi kerangka revisi API yang ditawarkan oleh versi platform Android.

Versi Platform	Tingkat API	VERSION_CODE	
Android 6.0	23	MARSHMALLOW	
Android 5.1	22	LOLLIPOP_MR1	
Android 5.0	21	LOLLIPOP	

Android 4.4W	20	KITKAT_WATCH	KitKat hanya untuk Pakaian
Android 4.4	19	KIT KAT	
Android 4.3	18	JELLY_BEAN_MR2	
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	
Android 4.1, 4.1.1	16	JELLY BEAN	
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	
Android 3.0.x	11	SARANG MADU	
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1	
Android 2.3.2 Android 2.3.1 Android 2.3	9	ROTI JAHE	
Android 2.2.x	8	FROYO	

Android 2.1.x	7	ECLAIR_MR1	
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONAT	
Android 1.5	3	CUPCAKE	
Android 1.1	2	BASE_1_1	
Android 1.0	1	MENDASARKAN	

Memasang Android Studio dan Menjalankan Hello World

Selamat datang di latihan praktis. Anda akan belajar:

- a) Memasang Android Studio, lingkungan development Android.
- b) Mempelajari proses development Android.
- c) Membuat dan menjalankan aplikasi Android Hello World pertama di emulator dan perangkat fisik.
- d) Menambahkan pencatatan log ke aplikasi Anda untuk men-debug.

Yang harus sudah Anda KETAHUI

Untuk praktik ini Anda harus bisa:

- a) Memahami proses development perangkat lunak umum untuk aplikasi berorientasi objek menggunakan IDE (Integrated Development Environment).
- b) Memiliki pengalaman pemrograman berorientasi objek paling sedikit 1-3 tahun, beberapa di antaranya fokus pada bahasa pemrograman Java. (Praktik ini tidak akan menjelaskan pemrograman berorientasi objek atau bahasa Java.)

Yang akan Anda BUTUHKAN

Untuk praktik ini, Anda akan membutuhkan:

- a) Komputer Mac, Windows, atau Linux. Lihat di bagian bawah laman unduhan Android Studio untuk persyaratan sistem terbaru.
- b) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.

Yang akan Anda PELAJARI

Anda akan belajar:

- a) Memasang dan menggunakan Android IDE.
- b) Memahami proses development untuk membangun aplikasi Android.
- c) Membuat proyek Android dari template aplikasi dasar.

Yang akan Anda LAKUKAN

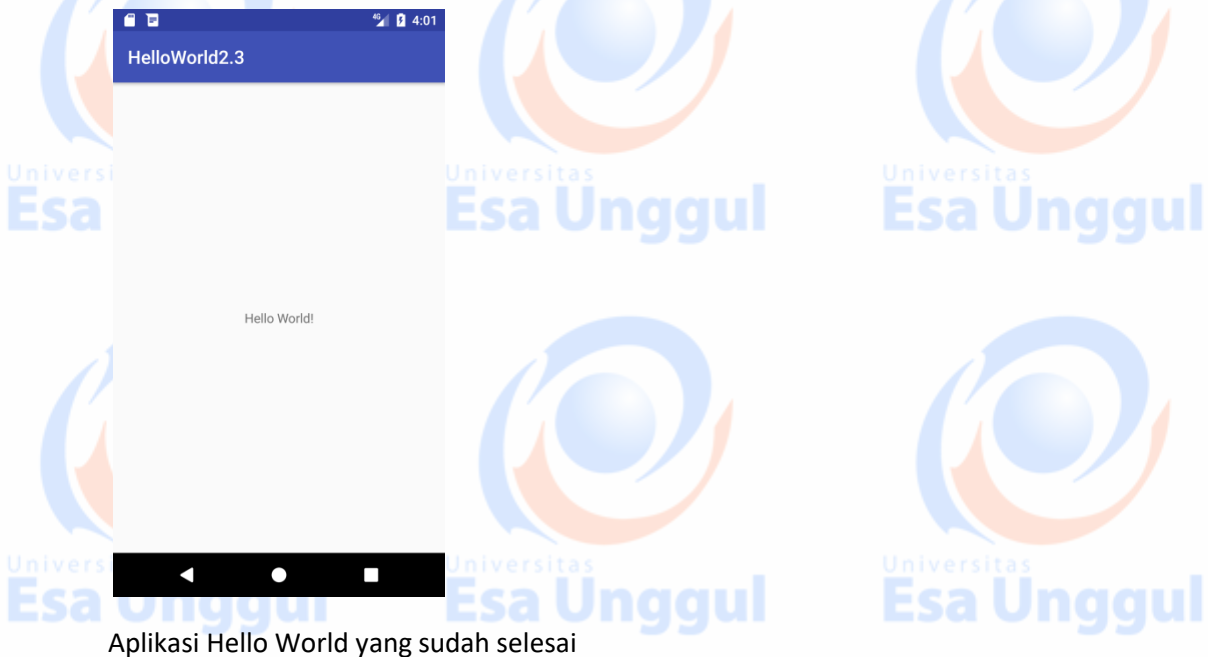
Memasang lingkungan development Android Studio.

- a) Membuat emulator (perangkat virtual) untuk menjalankan aplikasi Anda di komputer.
- b) Membuat dan menjalankan aplikasi Hello World di perangkat virtual dan fisik.
- c) Menjelajahi layout proyek.
- d) Membuat dan menampilkan pernyataan log dari aplikasi Anda.
- e) Menjelajahi file AndroidManifest.xml.

Ringkasan Aplikasi

Setelah berhasil memasang Android Studio IDE, Anda akan membuat proyek Android baru untuk aplikasi 'Hello World' dari sebuah template.

Aplikasi sederhana ini menampilkan string "Hello World" di layar perangkat Android virtual atau fisik. Berikut tampilan aplikasi yang sudah selesai:



Aplikasi Hello World yang sudah selesai

Tugas 1. Memasang Android Studio

Android Studio adalah IDE Google untuk aplikasi Android. Android Studio menyediakan editor kode tingkat lanjut dan serangkaian template aplikasi. Selain itu, Android Studio juga memiliki alat untuk development, men-debug, menguji, dan meningkatkan kinerja yang membuat pengembangan aplikasi lebih cepat dan mudah. Anda bisa menguji aplikasi dengan berbagai macam emulator yang telah dikonfigurasi sebelumnya atau di perangkat seluler Anda sendiri, dan membangun APK produksi untuk publikasi.

Catatan: Android Studio masih terus dikembangkan. Untuk informasi terbaru tentang persyaratan sistem dan petunjuk pemasangan, lihat dokumentasinya di developer.android.com.

Untuk menyiapkan dan mengoperasikan Android Studio:

- a) Anda mungkin perlu memasang Java Development Kit - Java 7 atau versi yang lebih baik.
- b) Memasang Android Studio

Android Studio dapat digunakan di komputer Windows, Mac, dan Linux. Pemasangannya sama untuk semua platform. Perbedaannya akan dijelaskan di bawah ini.

1.1. Memasang Java Development Kit

- 1) Buka jendela terminal di komputer.
- 2) Ketikkan `java -version`

Output termasuk baris:

Java(™) SE Runtime Environment (build1.X.0_05-b13)

X adalah nomor versi yang perlu dilihat.

- a) Jika nomornya 7 atau lebih besar, Anda bisa lanjut memasang Android Studio.
- b) Jika versi Java SE di bawah 7 atau belum dipasang, Anda perlu memasang versi terbaru paket development Java SE sebelum memasang Android Studio.

Untuk mengunduh Java Standard Edition (JSE) Development Kit (JDK):

- 1) Buka laman unduhan Oracle Java SE.
- 2) Klik ikon Java SE Downloads untuk membuka laman Java SE Development Kit 8 Downloads.
- 3) Di kotak paket Java SE Development terbaru, Anda perlu menerima Perjanjian Lisensi agar dapat melanjutkan. Lalu unduh versi yang sesuai untuk komputer yang Anda gunakan untuk mengembangkan.

Penting:Jangan masuk ke demo dan contoh (menunya terlihat sangat mirip, jadi pastikan Anda membaca judul di atas).

- 4) Memasang paket development. Setelah pemasangan JDK selesai, yang seharusnya hanya memakan waktu beberapa menit, Anda bisa mengonfirmasi bahwa pemasangan sudah benar dengan memeriksa versi Java dari baris perintah.
- 5) Buka jendela terminal dan ketikkan `java -version` lagi untuk memverifikasi bahwa pemasangan telah berhasil.
- 6) Setel variabel lingkungan `JAVA_HOME` ke direktori pemasangan JDK.

Windows:

- 1) Setel `JAVA_HOME` ke lokasi pemasangan.
- 2) Start > Control Panel > System > Advanced System Settings > Environment Variables System Variables > New
 - Nama variabel: `JAVA_HOME`
 - Nilai variabel: `C:\Program Files\Java\jdk1.7.0_80` (atau apa pun versi pemasangan Anda)
- 3) Jika variabel sudah ada, perbarui dengan versi JDK ini.
- 4) Verifikasi variabel `JAVA_HOME` dari terminal `cmd.exe`: `echo %JAVA_HOME%`

Lihat juga: https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/

Mac:

- 1) Buka Terminal.
- 2) Konfirmasi bahwa Anda memiliki JDK dengan mengetik "`which java`".
- 3) Periksa bahwa Anda memiliki versi Java yang diperlukan dengan mengetik "`java -version`".
- 4) Setel `JAVA_HOME` menggunakan perintah ini di Terminal: `ekspor JAVA_HOME=`which java``
- 5) masukkan `echo $JAVA_HOME` untuk mengonfirmasi jalurnya.

Linux:

Lihat: https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/

Penting: Jangan pasang Android Studio sampai setelah Java JDK dipasang. Tanpa salinan Java yang berfungsi dengan baik, sisa proses tidak akan bekerja. Jika Anda tidak dapat mengunduh, telusuri pesan kesalahan dan cari solusinya secara online.

Pemecahan Masalah Dasar:

- a) Tidak ada ikon UI, Control Panel, atau Startup yang terkait dengan JDK.
- b) Verifikasi bahwa Anda telah memasang JDK dengan benar dengan membuka direktori tempat Anda memasangnya. Untuk mengidentifikasi tempat JDK, lihat variabel PATH dan/atau cari direktori "jdk" atau "java" atau "javac" yang dapat dijalankan di komputer Anda.

1.2. Memasang Android Studio

- 1) Buka situs developer Android dan ikuti petunjuk untuk mengunduh dan memasang Android Studio.
 - Terima konfigurasi default untuk semua langkah.
 - Pastikan semua komponen dipilih untuk pemasangan.
- 2) Setelah selesai memasang. Wizard Persiapan akan mengunduh dan memasang beberapa komponen tambahan. Bersabarlah karena ini mungkin memakan waktu bergantung kecepatan internet Anda, dan beberapa langkah mungkin tampak berulang.
- 3) Setelah unduhan selesai, Android Studio akan dimulai, dan Anda siap membuat proyek pertama.

Pemecahan masalah: Jika menemui masalah dengan pemasangan, periksa dokumentasi terbaru, forum pemrograman, atau minta bantuan instruktur Anda.

Tugas 2: Membuat aplikasi "Hello World"

Dalam tugas ini, Anda akan mengimplementasikan aplikasi "Hello World" untuk memverifikasi bahwa Android Studio sudah dipasang dengan benar dan mempelajari dasar-dasar pengembangan dengan Android Studio.

2.1 Membuat aplikasi "Hello World"

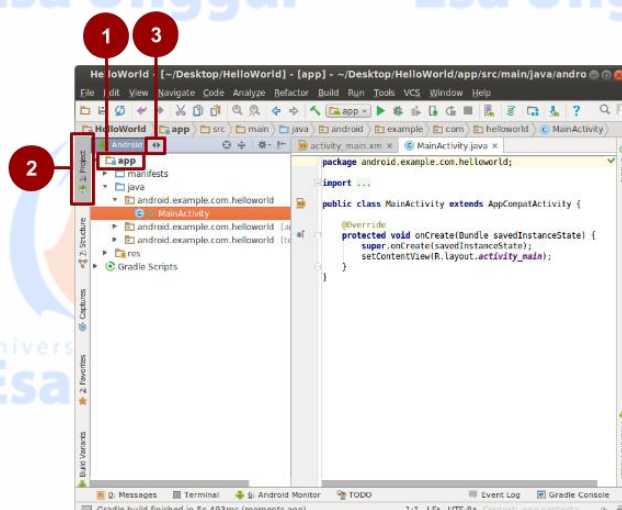
- 1) Buka Android Studio jika belum terbuka.
- 2) Di jendela utama Welcome to Android Studio, klik "Start a new Android Studio project".
- 3) Di jendela New Project, berikan aplikasi Anda Application Name, seperti "Hello World".
- 4) Verifikasi lokasi Proyek, atau pilih direktori yang berbeda untuk menyimpan proyek.
- 5) Pilih Company Domain yang unik.
 - Aplikasi yang dipublikasikan di Google Play Store harus memiliki nama paket yang unik. Karena domain unik, mengawali nama aplikasi dengan nama atau nama domain perusahaan Anda akan menghasilkan nama paket yang unik.
 - Jika tidak berencana mempublikasikan aplikasi, Anda bisa menerima domain contoh default. Perhatikan bahwa mengubah nama paket aplikasi di kemudian hari berarti melakukan pekerjaan ekstra.

- 6) Verifikasi bahwa Project Location default adalah tempat Anda menyimpan aplikasi Hello World dan proyek Android Studio lainnya, atau ubah lokasi ke direktori yang diinginkan. Klik Next.
- 7) Di layar Target Android Devices, "Phone and Tablet" harus dipilih. Dan Anda harus memastikan bahwa API 15: Android 4.0.3 IceCreamSandwich disetel sebagai Minimum SDK. (Perbaiki jika perlu.)
 - Saat buku ini ditulis, pemilihan tingkat API ini menjadikan aplikasi "Hello World" kompatibel dengan 97% perangkat Android yang aktif di Google Play store.
 - Ini adalah setelan yang digunakan oleh contoh di buku ini.
- 8) Klik Next.
- 9) Jika proyek Anda memerlukan komponen tambahan untuk SDK target terpilih, Android Studio akan memasangnya secara otomatis. Klik Next.
- 10) Jendela Customize the Activity. Setiap aplikasi membutuhkan paling tidak satu aktivitas. Aktivitas mewakili satu layar dengan satu antarmuka pengguna dan Android Studio menyediakan template untuk membantu Anda memulai. Untuk proyek Hello World, pilih template termudah (saat ini ditulis, template proyek "Empty Activity" adalah template paling sederhana) yang tersedia.
- 11) Adalah praktik yang biasa untuk menyebut aktivitas utama Anda MainActivity. Ini bukan persyaratan.
- 12) Pastikan kotak Generate Layout file dicentang (jika terlihat).
- 13) Pastikan kotak Backwards Compatibility (App Compat) dicentang.
- 14) Biarkan Layout Name sebagai activity_main. Layout biasanya diberi nama sesuai dengan aktivitasnya. Terima defaultnya dan klik Finish.

Setelah langkah-langkah ini, Android Studio:

- 1) Membuat folder untuk Proyek Android Studio Anda.
- 2) Bangun proyek Anda dengan Gradle (ini mungkin memerlukan beberapa waktu). Android Studio menggunakan Gradle sebagai sistem build-nya. Lihat laman developer Konfigurasi build untuk informasi selengkapnya.
- 3) Membuka editor kode dengan proyek Anda.
- 4) Menampilkan tip hari ini.
 - Android Studio menawarkan banyak pintasan keyboard, dan membaca tip adalah cara yang bagus untuk mempelajarinya dari waktu ke waktu.

Jendela Android Studio harus terlihat mirip dengan diagram berikut:



Anda bisa melihat hierarki file untuk aplikasi Anda dengan beberapa cara.

- 1) Klik di folder Hello World folder untuk meluaskan hierarki file (1),
- 2) Klik di Project (2).
- 3) Klik di menu Android(3).
- 4) Jelajahi opsi tampilan yang berbeda untuk proyek Anda.

Catatan: Buku ini menggunakan tampilan Android dari file proyek, kecuali ditentukan lain.

Tugas 3: Jelajahi struktur proyek

Dalam praktik ini, Anda akan menjelajahi cara file proyek disusun di Android Studio.

Langkah-langkah ini mengasumsikan bahwa proyek Hello World Anda dimulai seperti yang ditampilkan pada diagram di atas.

3.1 Menjelajahi struktur proyek dan layout

Di tampilan Project > Android tugas sebelumnya, ada tiga folder level teratas di bawah folder **app** Anda: **manifests**, **java**, dan **res**.

1. Luaskan folder manifests.

Folder ini berisi AndroidManifest.xml. File ini menjelaskan semua komponen aplikasi Android Anda dan dibaca oleh sistem waktu proses Android saat program dijalankan.

2. Luaskan folder java. Semua file bahasa Java dikelola di folder ini. Folder java berisi tiga subfolder:

- **com.example.hello.helloworld** (atau nama domain yang telah Anda tetapkan): Semua file untuk paket ada di folder yang bernama sama dengan paket tersebut. Untuk aplikasi Hello World, ada satu paket dan paket berisi MainActivity.java (ekstensi file dapat dihilangkan di tampilan Proyek).
- **com.example.hello.helloworld** (androidTest): Folder ini untuk pengujian berinstrumen Anda, dan memulai dengan file pengujian kerangka.
- **com.example.hello.helloworld** (test): Folder ini untuk pengujian unit dan memulai dengan file pengujian unit kerangka yang secara otomatis dibuat.

3. Luaskan folder **res**. Folder ini berisi semua sumber daya untuk aplikasi Anda, termasuk gambar, file layout, string, ikon, dan penataan gaya. Folder ini berisi subfolder berikut:

- **drawable**. Simpan semua gambar aplikasi Anda di folder ini.
- **layout**. Setiap aktivitas memiliki paling tidak satu file layout yang menjelaskan UI di XML. Untuk Hello World, folder ini berisi activity_main.xml.
- **mipmap**. Simpan ikon peluncur Anda di folder ini. Ada subfolder untuk setiap kepadatan layar yang didukung. Android menggunakan kepadatan layar, yaitu, jumlah piksel per inci untuk menentukan resolusi gambar yang diperlukan. Android mengelompokkan kepadatan layar yang sebenarnya ke dalam kepadatan umum seperti medium (mdpi), tinggi (hdpi), atau ekstra-ekstra-ekstra-tinggi (xxxhdpi). Folder ic_launcher.png berisi ikon peluncur default untuk semua kepadatan yang didukung oleh aplikasi Anda.
- **values**. Sebagai ganti melakukan hardcode nilai-nilai seperti string, dimensi, dan warna di file XML dan Java, praktik terbaiknya adalah mendefinisikannya menurut file nilai. Ini membuatnya lebih mudah diubah dan konsisten di seluruh aplikasi Anda.

4. Luaskan subfolder values dalam folder res. Folder ini berisi subfolder berikut:

colors.xml. Tunjukkan warna default untuk tema pilihan Anda, dan Anda bisa menambahkan warna sendiri atau mengubahnya berdasarkan persyaratan aplikasi Anda.

dimens.xml. Simpan ukuran tampilan dan objek untuk resolusi yang berbeda.

strings.xml. Buat sumber daya untuk semua string Anda. Ini memudahkan penerjemahan string ke bahasa lain.

styles.xml. Semua gaya untuk aplikasi dan tema Anda ada di sini. Gaya membantu memberikan aplikasi Anda tampilan yang konsisten untuk semua elemen UI.

3.2 Sistem pembangunan Gradle

Android Studio menggunakan **Gradle** sebagai sistem pembangunan. Seiring dengan kemajuan Anda sepanjang praktik ini, Anda akan mempelajari lebih banyak tentang gradle dan apa yang dibutuhkan untuk membangun dan menjalankan aplikasi Anda.

1. Luaskan folder ***Gradle Scripts**. Folder ini berisi semua file yang dibutuhkan oleh sistem pembangunan.
2. Cari file **build.gradle(Module:app)**. Saat Anda menambahkan dependensi khusus aplikasi, seperti menggunakan pustaka tambahan, dependensi masuk ke dalam file ini.

Tugas 4: Membuat perangkat virtual (emulator)

Pada tugas ini, Anda akan menggunakan Android Virtual Device (AVD) manager untuk membuat perangkat virtual atau emulator yang menyimulasikan konfigurasi untuk jenis perangkat Android tertentu.

Menggunakan AVD Manager, Anda mendefinisikan karakteristik perangkat keras sebuah perangkat dan API level-nya, serta menyimpannya sebagai konfigurasi perangkat virtual.

Saat Anda memulai emulator Android, emulator membaca konfigurasi yang ditetapkan dan membuat perangkat emulasi yang bertindak persis seperti versi fisik perangkat tersebut, namun berada di komputer Anda.

Mengapa: Dengan perangkat virtual, Anda bisa menguji aplikasi di perangkat yang berbeda (tablet, ponsel) dengan API level berbeda untuk memastikan aplikasi terlihat bagus dan berfungsi untuk sebagian besar pengguna. Anda tidak perlu bergantung pada perangkat fisik yang tersedia untuk development aplikasi.

4.1 Membuat perangkat virtual

Untuk menjalankan emulator di komputer, Anda perlu membuat konfigurasi yang menjelaskan perangkat virtual.

1. Di Android Studio, pilih Tools > Android > AVD Manager atau klik ikon AVD Manager Ikon

AVD Manager  di bilah alat.

2. Klik **+Create Virtual Device....** (Jika Anda telah membuat perangkat virtual sebelumnya, jendela akan menunjukkan semua perangkat yang ada dan tombolnya ada di bagian bawah.) Layar Select Hardware muncul menunjukkan daftar perangkat keras yang telah dikonfigurasi sebelumnya. Untuk setiap perangkat, tabel menunjukkan ukuran tampilan (Size), resolusi layar dalam piksel (Resolution), dan kepadatan piksel (Density).


Untuk perangkat Nexus 5 , kepadatan pikselnya adalah xxhdpi, yang berarti aplikasi Anda menggunakan ikon peluncur di folder xxhdpi dari folder mipmap. Aplikasi Anda juga akan menggunakan layout dan drawable dari folder yang didefinisikan untuk kepadatan itu juga.

3. Pilih perangkat keras Nexus 5 dan klik **Next**.
4. Pada layar **System Image**, dari tab **Recommended**, pilih versi sistem Android yang akan dijalankan pada perangkat virtual. Anda bisa memilih gambar sistem terbaru. Ada lebih banyak versi yang tersedia dari yang ditunjukkan di tab **Recommended**. Lihat tab **x86 Images** dan **Other Images** untuk melihatnya.
5. Jika tautan **Download** terlihat di samping versi gambar sistem, berarti versi tersebut belum terpasang dan Anda perlu mengunduhnya. Jika perlu, klik tautan untuk mulai mengunduh dan klik **Finish** setelah selesai.
6. Pada layar **System Image**, pilih gambar sistem dan klik **Next**.
7. Verifikasi konfigurasi Anda dan klik **Finish**. (Jika jendela **AVD Manager Your Android Devices** tetap terbuka, Anda bisa melanjutkan dan menutupnya.)

Tugas 5: Menjalankan aplikasi Anda di emulator

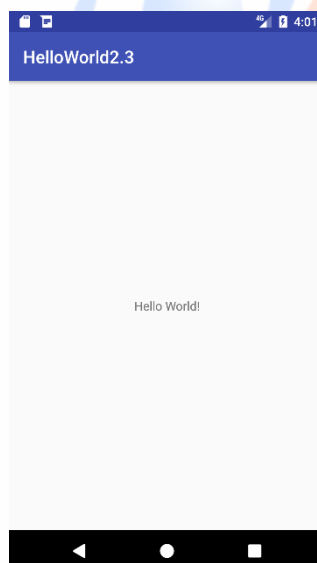
Pada tugas ini Anda akhirnya akan menjalankan aplikasi Hello World.

5.1 Jalankan aplikasi Anda di emulator

1. Di Android Studio, pilih Run > Run app atau klik **ikon Run**  Android Studio di bilah alat.
2. Di jendela Select Deployment Target, di bawah Available Emulators, pilih Nexus 5 API 23 dan klik **OK**.

Emulator dimulai dan di-boot, seperti perangkat fisik. Tergantung kecepatan komputer Anda, ini mungkin akan memerlukan beberapa saat. Aplikasi Anda dibangun, dan setelah emulator siap, Android Studio akan mengunggah aplikasi ke emulator dan menjalankannya.

Anda seharusnya melihat aplikasi Hello World seperti yang ditampilkan di tangkapan layar berikut.



Catatan: Saat menguji emulator, praktik terbaiknya adalah dengan memulainya sekali, di awal sesi Anda. Jangan tutup emulator sampai Anda selesai menguji aplikasi, agar aplikasi tidak perlu melalui proses booting lagi.

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional dan tidak diwajibkan untuk praktik selanjutnya.

Tantangan: Anda bisa sepenuhnya menyesuaikan perangkat virtual.

- Pelajari [dokumentasi AVD Manager](#).
- Buat satu atau beberapa perangkat virtual khusus.

Anda mungkin melihat bahwa tidak semua kombinasi perangkat dan versi sistem berfungsi saat menjalankan aplikasi. Ini karena tidak semua gambar sistem dapat berjalan di semua perangkat keras.

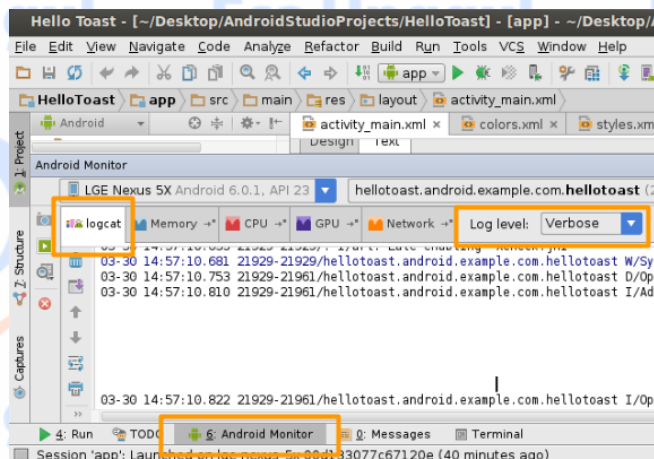
Tugas 6. Menambahkan pernyataan log ke aplikasi Anda

Dalam praktik ini, Anda akan menambahkan pernyataan log ke aplikasi, yang ditampilkan di jendela pencatatan log Android Monitor.

Mengapa: Pesan log adalah alat debug bermanfaat yang bisa Anda gunakan untuk memeriksa nilai, jalur eksekusi, dan melaporkan pengecualian.

Android Monitor menampilkan informasi tentang aplikasi Anda.

1. Klik tombol **Android Monitor** di bawah Android Studio untuk membuka Android Monitor. Secara default, ini membuka tab **logcat**, yang menampilkan informasi tentang aplikasi Anda saat sedang dijalankan. Jika Anda menambahkan pernyataan log ke aplikasi, pernyataan dicetak di sini juga. Anda juga bisa memantau Memori, CPU, GPU, dan kinerja jaringan aplikasi dari tab lain Android Monitor. Ini dapat membantu debug dan kinerja menyempurnakan kode.
2. Level log default adalah **Verbose**. Di menu tarik-turun, ubah level log ke **Debug**.



Pernyataan log yang Anda tambahkan ke kode aplikasi mencetak pesan yang ditentukan oleh Anda di tab logcat Android Monitor. Misalnya:

```
Log.d("MainActivity", "Hello World");
```

Bagian dari pesan tersebut adalah:

- Log – [Kelas Log](#). API untuk mengirim pesan log.
- d – Tingkat Log. Digunakan untuk memfilter tampilan pesan di logcat. "d" untuk debug. Tingkat log lainnya adalah "e" untuk error, "w" untuk warning, dan "i" untuk info.
- "MainActivity" – Argumen pertama adalah tag yang dapat digunakan untuk memfilter pesan di logcat. Ini biasanya adalah nama aktivitas asal pesan. Tetapi, Anda dapat menjadikannya apa saja yang berguna untuk men-debug.

Sesuai ketentuan, tag log didefinisikan sebagai konstanta:

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

"Hello world" – Argumen kedua adalah pesan yang sebenarnya.

6.1 Menambahkan pernyataan log ke aplikasi Anda

1. Buka aplikasi Hello World di Android studio, dan buka file MainActivity.
2. **File > Settings > Editor > General > Auto Import (Mac: Android Studio > Preferences > Editor > General > Auto Import)**. Pilih semua kotak centang dan setel **Insert imports on paste** ke **All**. Impor yang jelas sekarang ditambahkan secara otomatis ke file Anda. Catat bahwa opsi "add unambiguous imports on the fly" penting bagi beberapa fitur Android seperti NumberFormat. Jika tidak dicentang, NumberFormat menampilkan kesalahan. Klik 'Apply' lalu klik tombol 'Ok'.
3. Dalam metode onCreate, tambahkan pernyataan log berikut:

```
Log.d("MainActivity", "Hello World");
```
4. Jika Android Monitor belum dibuka, klik tab Android Monitor di bawah Android Studio untuk membukanya. (Lihat tangkapan layar.)
5. Pastikan level Log di logcat Android Monitor disetel ke Debug atau Verbose (default).
6. Jalankan aplikasi Anda.

Kode solusi:

```
package com.example.hello.helloworld;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.activity_main);

Log.d("MainActivity", "Hello World");

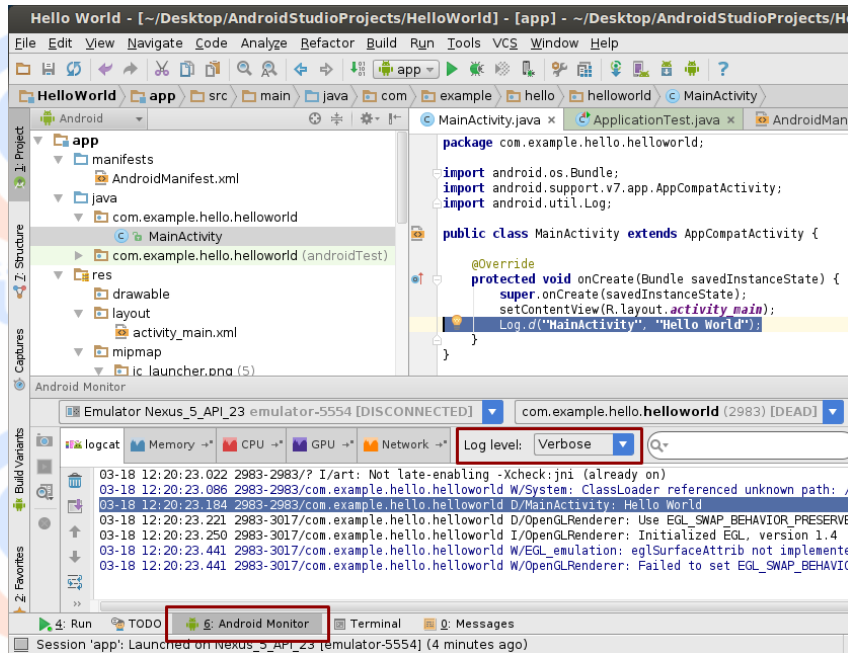
}

}

```

Pesan Log Output

3-18 12:20:23.184 2983-2983/com.example.hello.helloworld D/MainActivity: Hello World



Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk bab berikutnya.

Tantangan: Penggunaan umum kelas Log adalah mencatat log pengecualian Java saat terjadi di program Anda. Ada beberapa metode berguna dalam kelas Log yang dapat digunakan untuk tujuan ini. Gunakan dokumentasi kelas Log untuk mencari tahu metode apa yang bisa Anda gunakan untuk menyertakan pengecualian dengan pesan log. Lalu, tulis kode di file MainActivity.java untuk memicu dan mencatat pengecualian.

Tugas 7: Menjelajahi file AndroidManifest.xml

Setiap aplikasi menyertakan file Manifest Android (AndroidManifest.xml). File manifest berisi informasi penting tentang aplikasi Anda dan menghadirkan informasi ini ke sistem waktu proses Android. Android harus memiliki informasi ini sebelum dapat menjalankan kode aplikasi apa pun.

Dalam praktik ini Anda akan menemukan dan membaca file AndroidManifest.xml untuk aplikasi Hello World.

Mengapa: Saat aplikasi menambahkan lebih banyak fungsionalitas dan pengalaman pengguna jadi lebih menarik dan interaktif, file AndroidManifest.xml berisi lebih banyak lagi informasi. Pada pelajaran berikutnya, Anda akan memodifikasi file ini untuk menambahkan fitur dan izin fitur.

7.1 Jelajahi file AndroidManifest.xml

1. Buka aplikasi Hello World di Android studio, dan di folder **manifests** folder, buka **AndroidManifest.xml**.
2. Baca file dan pertimbangkan apa yang ditunjukkan oleh setiap kode. Kode di bawah ini dianotasi untuk memberi Anda petunjuk.

Kode beranotasi:

```
<!-- XML version and character encoding -->

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

<!-- Required starting tag for the manifest -->

<manifest

<!-- Defines the android namespace. Do not change. -->

xmlns:android="http://schemas.android.com/apk/res/android"

<!-- Unique package name of your app. Do not change once app is
published. -->

package="com.example.hello.helloworld">

<!-- Required application tag -->

<application

<!-- Allow the application to be backed up and restored. -->

android:allowBackup="true"

<!-- Icon for the application as a whole,
and default icon for application components. -->

android:icon="@mipmap/ic_launcher"

<!-- User-readable for the application as a whole,
and default icon for application components. Notice that Android
Studio first shows the actual label "Hello World".

Click on it, and you will see that the code actually refers to a string
resource. Ctrl-click @string/app_name to see where the resource is
specified. This will be covered in a later practical . -->
```



```
    android:label="@string/app_name"

    <!-- Whether the app is willing to support right-to-left layouts.-->

    android:supportsRtl="true"

    <!-- Default theme for styling all activities. -->

    android:theme="@style/AppTheme">

    <!-- Declares an activity. One is required.

    All activities must be declared,

    otherwise the system cannot see and run them. -->

    <activity

    <!-- Name of the class that implements the activity;

    subclass of Activity. -->

    android:name=".MainActivity">

    <!-- Specifies the intents that this activity can respond to.-->

    <intent-filter>

    <!-- The action and category together determine what

    happens when the activity is launched. -->

    <!-- Start activity as the main entry point.

    Does not receive data. -->

    <action android:name="android.intent.action.MAIN" />

    <!-- Start this activity as a top-level activity in

    the launcher . -->

    <category android:name="android.intent.category.LAUNCHER" />

    <!-- Closing tags -->

    </intent-filter>

    </activity>

</application>

</manifest>
```

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional.

Tantangan: Ada banyak elemen lain yang dapat disetel di Android Manifest. Jelajahi dokumentasi Android Manifest dan pelajari tentang elemen tambahan di Android Manifest.

Tugas 8. Menjelajahi file build.gradle

Android Studio menggunakan sistem pembangunan bernama Gradle. Gradle melakukan pembangunan bertahap, yang mengizinkan siklus edit-uji yang lebih singkat.

Untuk mengetahui selengkapnya tentang Gradle, buka:

- [Situs Gradle](#)
- Dokumentasi developer [Mengonfigurasi pembangunan](#)
- Telusuri "tutorial gradle" di internet.

Dalam tugas ini, Anda akan menjelajahi file build.gradle.

Mengapa: Saat menambahkan pustaka baru ke proyek Android, Anda mungkin juga harus memperbarui **file build.gradle**. Mengetahui di mana file tersebut dan struktur dasarnya akan berguna bagi Anda.

8.1 Jelajahi file build.gradle(Module.app)

Di hierarki proyek, cari **Gradle Scripts** dan luaskan. Ada beberapa file build.gradle. Satu dengan arahan untuk seluruh proyek dan satu untuk setiap modul aplikasi. Modul untuk aplikasi Anda disebut "app". Pada tampilan Proyek, modul ini diwakili oleh folder **app** di tingkat atas tampilan Proyek.

Buka **build.gradle (Module.app)**.

Baca file dan pelajari apa yang ditunjukkan oleh setiap baris kode.

```
// Add Android-specific build tasks

apply plugin: 'com.android.application'

// Configure Android specific build options.

android {

    // Specify the target SDK version for the build.

    compileSdkVersion 23

    // The version of the build tools to use.

    buildToolsVersion "23.0.2"
```

```

// Core settings and entries. Overrides manifest settings!

defaultConfig {

    applicationId "com.example.hello.helloworld"

    minSdkVersion 15

    targetSdkVersion 23

    versionCode 1

    versionName "1.0"
}

// Controls how app is built and packaged.

buildTypes {

    // Another common option is debug, which is not signed by default.

    release {

        // Code shrinker. Turn this on for production along with

        // shrinkResources.

        minifyEnabled false

        // Use ProGuard, a Java optimizer.

        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

    }

}

// This is the part you are most likely to change as you start using

// other libraries.

dependencies {

    // Local binary dependency. Include any JAR file inside app/libs.

    compile fileTree(dir: 'libs', include: ['*.jar'])

    // Configuration for unit tests.

    testCompile 'junit:junit:4.12'

    // Remote binary dependency. Specify Maven coordinates of the Support

```

```
// Library needed. Use the SDK Manager to download and install such  
  
// packages.  
  
compile 'com.android.support:appcompat-v7:23.2.1'  
  
}
```

- Untuk lebih memahami Gradle baca [Ringkasan Sistem Pembangunan](#) dan dokumentasi [Mengonfigurasi Build Gradle](#).
- Terdapat alat yang dapat membantu Anda [meringkas kode](#), menghapus pustaka/sumber daya tidak penting dan bahkan menyamarkan program untuk mencegah rekayasa balik yang tidak diinginkan.
- Android Studio sendiri menyediakan beberapa fitur yang berguna. Pelajari lebih lanjut tentang alat sumber terbuka yang bermanfaat yang bernama [ProGuard](#).

Tugas 9. [Opsional] Menjalankan aplikasi Anda di perangkat

Pada tugas terakhir ini, Anda akan menjalankan aplikasi pada perangkat seluler fisik seperti ponsel atau tablet.

Mengapa: Pengguna akan menjalankan aplikasi di perangkat fisik. Anda harus selalu menguji aplikasi pada perangkat fisik dan virtual.

Yang Anda butuhkan:

- Perangkat Android seperti ponsel atau tablet.
- Kabel data untuk menghubungkan perangkat Android ke komputer melalui port USB.
- Jika menggunakan OS Linux atau Windows, Anda mungkin perlu melakukan langkah tambahan untuk menjalankannya di perangkat keras. Periksa dokumentasi [Menggunakan Perangkat Keras](#). Pada Windows, Anda mungkin perlu memasang driver USB yang sesuai untuk perangkat. Lihat [Driver US OEM](#).

Tugas 9.1. [Opsional] Menjalankan aplikasi Anda di perangkat


Untuk mengizinkan Android Studio berkomunikasi dengan perangkat, Anda harus mengaktifkan USB Debugging di perangkat Android. Ini diaktifkan di setelan Developer options perangkat. Perlu diingat bahwa ini tidak sama dengan melakukan rooting perangkat.

Pada Android 4.2 dan yang lebih tinggi, layar Developer options disembunyikan secara default.

Untuk menampilkan Developer options dan mengaktifkan USB Debugging:

- Pada perangkat, buka Settings > About phone dan ketuk Build number tujuh kali.
- Kembali ke layar sebelumnya (Settings). Developer options muncul di bawah daftar. Klik Developer options.
- Pilih USB Debugging.

Sekarang Anda dapat menghubungkan perangkat dan menjalankan aplikasi dari Android Studio.

1. Hubungkan perangkat ke mesin development dengan kabel USB.
2. Di Android Studio, di bawah jendela, klik tab Android Monitor. Anda seharusnya melihat perangkat terdaftar di menu tarik-turun kiri atas.
3. Klik tombol Run  Android Studio di bilah alat. Jendela **Select Deployment Target** terbuka dengan daftar emulator yang tersedia dan perangkat yang terhubung.
4. Pilih perangkat dan klik **OK**.

Android Studio seharusnya memasang dan menjalankan aplikasi di perangkat Anda.

Pemecahan Masalah

Jika Android Studio tidak mengenali perangkat, coba langkah berikut:

- Cabut dan colokkan kembali perangkat.
- Mulai ulang Android Studio.
- Jika komputer masih tidak mengenali perangkat dan mendeklarasikannya "unauthorized":
 1. Cabut perangkat.
 2. Pada perangkat, buka Settings->Developer Options.
 3. Ketuk Revoke USB Debugging authorizations.
 4. Hubungkan kembali perangkat ke komputer.
 5. Saat diminta, berikan otorisasi.

Anda mungkin perlu memasang driver USB yang sesuai untuk perangkat. Baca [dokumentasi Menggunakan Perangkat Keras](#).

Periksa dokumentasi terbaru, forum pemrograman, atau minta bantuan instruktur Anda.

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional.

Tantangan: Sekarang karena Anda sudah siap dan mengenal alur kerja development, lakukan hal berikut:

1. Buat proyek baru di Android Studio.
2. Ganti sapaan dengan "Happy Birthday to " dan seseorang yang baru saja berulang tahun.
3. Ganti latar belakang aplikasi menggunakan gambar bertema ulang tahun.
4. Ambil tangkapan layar dari aplikasi yang telah selesai dan kirimkan lewat email ke seseorang yang ulang tahunnya Anda lupakan.

Rangkuman

Pada bab ini, Anda telah mempelajari cara:

- Memasang Android Studio
- ✍ Mendapatkan pemahaman dasar tentang alur kerja development setelah membukanya di Android Studio.

- ✍ Mendapatkan pemahaman dasar tentang struktur aplikasi Android di lingkungan pembangunannya.
- ✍ Mendapatkan pemahaman dasar tentang Manifes Android dan kegunaannya.
- ✍ Menambahkan pernyataan log ke kode yang memberi Anda alat dasar untuk proses debug.
- Menerapkan aplikasi Hello World pada emulator Android dan [opsional] perangkat seluler.

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Cara Menampilkan Welcome Screen Dengan Text Android

Welcome screen digunakan dalam pengembangan aplikasi Android, untuk menampilkan sebuah pesan text ataupun gambar ketika user (pengguna) membuka pertama kali Aplikasi yang telah diinstall. Welcome screen bisa juga berisi pesan panduan penggunaan Aplikasi yang ditujukan untuk user (pengguna).

Pada modul ini, kita akan mencoba membuat sebuah Contoh penerapan welcome screen dengan menggunakan TextView, kalian juga bisa menambahkan gambar dengan ImageView jika kalian mau.

1. Pertama kita terapkan baris instruksi (coding dibawah) ke file activity_main.xml

```
<?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:orientation="vertical">

    <RelativeLayout
        android:id="@+id/welcome_pesanan"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="#036e64"
        android:orientation="vertical"
        android:layout_margin="16dp"
        android:padding="16dp"
        android:visibility="visible">

        <TextView
            android:id="@+id/welcome_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:layout_marginTop="16dp"
            android:text="Selamat datang di Okedroid"
            android:textColor="#ddd"
            android:textSize="28sp"
            android:textStyle="bold" />
```

```

<TextView
    android:id="@+id/welcome_messgae"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/welcome_title"

    android:text="Okedroid.com adalah Blog yang membahas tentang tutorial
belajar membuat dan mengembangkan Aplikasi Android dengan Android Studio serta
Info Teknologi lainnya. \n\n Kalian bisa belajar banyak disini baik pemula maupun expert"

    android:textColor="#ddd"
    android:textSize="18sp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_gravity="bottom"
    android:background="#3b978d"
    android:onClick="dismissWelcomeMessageBox"
    android:paddingLeft="30dp"
    android:paddingRight="30dp"
    android:text="Got It"
    android:textColor="#fff" />

</RelativeLayout>

<LinearLayout
    android:id="@+id/content_layout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#95a5a6"
    android:orientation="vertical"
    android:padding="20dp"
    android:visibility="invisible">

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pesan ini akan tampil setelah kalian melihat welcome screen
dengan text "
    android:textColor="#fff"
    android:textSize="25sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Okedroid.com"
    android:textSize="30sp" />

</LinearLayout>

</RelativeLayout>

```

2. Kedua kita terapkan pada file java MainActivity.java
MainActivity.java

```

package com.okedroid.myapplication3;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;

public class MainActivity extends AppCompatActivity {

    RelativeLayout introMessage;
    LinearLayout appContent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        introMessage = (RelativeLayout) findViewById(R.id.welcome_pesan);
    }
}

```

```
appContent = (LinearLayout) findViewById(R.id.content_layout);  
}  
public void dismissWelcomeMessageBox(View view) {  
    introMessage.setVisibility(View.INVISIBLE);  
    appContent.setVisibility(View.VISIBLE);  
}  
}
```

3. Sekarang kita coba jalankan Aplikasinya lewat Android Studio
Kurang lebih hasilnya akan seperti gambar dibawah:



Hasil Welcome Screen



Hasil Pesan Konten

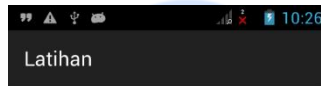


Cara Menghilangkan Title Bar Pada Activity

Title Bar merupakan judul pada tiap-tiap activity dalam Aplikasi Android. Title Bar berbeda dengan Action bar bila title bar hanya menampilkan judul. Bila ActionBar menampilkan menu-menu untuk melakukan suatu tindakan menuju activity-activity kita.

Biasanya Title Bar di hilangkan agar tampilan lebih leluasa dan memperindah tampilan, biasanya dalam pembauatn game semua tampilan Title Bar di hilangkan.

Sebelum Title Bar dihilangkan



10:26

Title Bar sebelum hilang

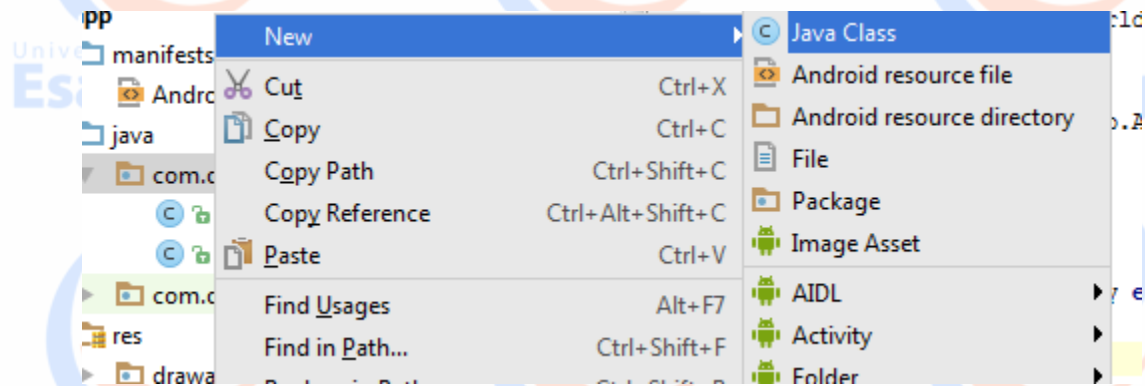
Sesudah Title Bar dihilangkan



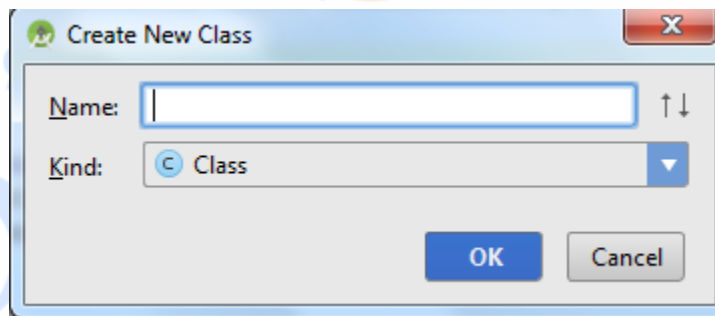
Title Bar Sesudah Hilang

Buat file java

Klik kanan package pada folder **app>Java>com.okedroid.helloworld** (package disesuaikan dengan yang kamu buat)



Buat file java



Beri nama file java

Berikut codingan pemrogramannya.

Letakan Code berikut di MainActivity.java (Default Activity di Aplikasi Android sesuaikan dengan Activity kalian)

```
this.requestWindowFeature(Window.FEATURE_NO_TITLE);
```

tepatnya pada seperti code berikut :

```
package example.com.latihan;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import android.view.Window;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
this.requestWindowFeature(Window.FEATURE_NO_TITLE);  
setContentView(R.layout.activity_main);  
}  
}
```

Dan untuk settingan AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="example.com.latihan" >  
    <application  
        android:allowBackup="true"  
        android:icon="@drawable/ic_launcher"  
        android:label="@string/app_name"  
        android:theme="@android:style/Theme.NoTitleBar">  
        <activity  
            android:name=".MainActivity"  
            android:label="@string/app_name" >  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

Cara Menerapkan Contoh Drop Down Android

Drop Down adalah salah satu menu pada suatu komponen User Interface (Antar Muka), yang ketika di klik atau dipilih akan muncul menu navigasi yang memanjang kebawah. Drop down biasa digunakan untuk menu navigasi yang memanjang kebawah. Drop down biasa digunakan untuk menu navigasi dari sebuah blog atau website, untuk memudahkan pengunjung dalam menampilkan isi konten halaman blog atau website tersebut. Nah kali ini kita akan coba terapkan pada Aplikasi Android.

Pada modul ini, kita akan mencoba membuat/menerapkan menu drop down pada aplikasi Android. Seperti yang biasa digunakan pada sebuah Blog atau website. Disini kita menggunakan widget Spinner untuk membuatnya :

1. Pertama kita akan menerapkan baris instruksi (codingan) di bawah pada file layout activity_main.xml
Copy baris instruksi ke activity_main.xml
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:text="Contoh daftar Android Drop Down"
        android:textSize="24sp" />
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:backgroundTint="#2ecc71"
        android:entries="@array/android_dropdown_arrays"
        android:padding="5dp" />
```

```
<Spinner
    android:id="@+id/spinner1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:backgroundTint="#2ecc71"
    android:entries="@array/android_dropdown_arrays"
    android:padding="5dp" />
```

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_marginBottom="20dp"
    android:gravity="bottom|center"
    android:text="okedroid.com"
    android:textSize="24sp"
    android:textStyle="bold" />
```

```
</LinearLayout>
```

2. Kedua kita akan menerapkan (codingan) di bawah pada file layout MainActivity.java
MainActivity.java

```
package com.okedroid.fathur.myapplication;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.AdapterView;

import android.widget.Spinner;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private Spinner spinner1;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        spinner1 = (Spinner) findViewById(R.id.spinner);

        spinner1.setOnItemSelectedListener(new ItemSelectedListener());

    }

    public class ItemSelectedListener implements AdapterView.OnItemSelectedListener {

        //get strings of first item

        String firstItem = String.valueOf(spinner1.getSelectedItem());

        public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {

            if (firstItem.equals(String.valueOf(spinner1.getSelectedItem()))) {

                // ToDo when first item is selected

            } else {

                Toast.makeText(parent.getContext(),

                    "Kamu telah memilih : " + parent.getItemAtPosition(pos).toString(),

                    Toast.LENGTH_LONG).show();

                // Todo when item is selected by the user

            }

        }

    }

}
```



```

@Override

public void onNothingSelected(AdapterView<?> arg) {

}

}

}

```

3. Ketiga kita akan menerapkan (codingan) bawah pada file String.xml

```

<resources>

<string name="app_name">Contoh Drop Down</string>

<string-array name="android_dropdown_arrays">

<item>Pilih Item</item>

<item>Daftar Drop Down 1</item>

<item>Daftar Drop Down 2</item>

<item>Daftar Drop Down 3</item>

<item>Daftar Drop Down 5</item>

<item>Daftar Drop Down 6</item>

<item>Daftar Drop Down 7</item>

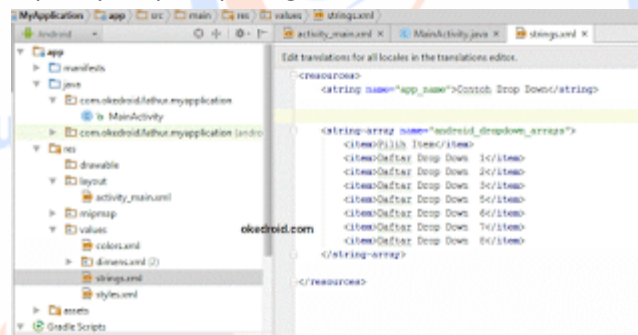
<item>Daftar Drop Down 8</item>

</string-array>

</resources>

```

Kira-kira penerapannya seperti pada gambar di bawah :



Preview String.xml di Android Studio

- Setelah semua selesai, mari kita uji coba dengan menjalankan Aplikasinya di Android Studio. Hasilnya mungkin akan seperti ini;



Hasil contoh Drop Down Android

Cara Menerapkan Contoh Popup Menu (Dropdown Menu) Android

Popoup menu adalah salah satu komponen menu yang akan tampil, secara baris ke bawah. Ketika user (pengguna) mengklik tombol atau item tertentu. Popup menu juga bisa disebut drop down menu, karena tampilannya sama seperti bentuk dropdown, yang biasa kita temui seperti pada navigasi website.

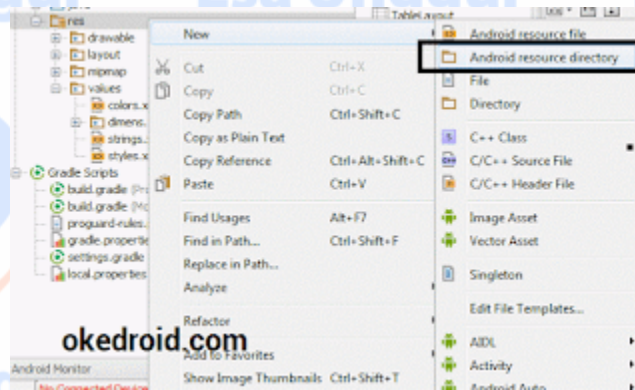
Pada modul ini, kita akan mencoba membuat dan menerapkan contoh PopUp Menu Android. Disini kita akan membuat widget button yang dimana ketika diklik akan muncul pop up menu (drop down menu), yang berisi pilihan menu seperti copy, delete dan rename. Dengan menggunakan fungsi atau method `OnMenuItemClickListener`.

1. Pertama buka project **Android Studio** kalian.
2. Lalu kita akan membuat file xml pada folder `app/res/menu/file.xml`

Jika tidak ada folder menu, kita bisa membuat foldernya secara manual dengan cara :

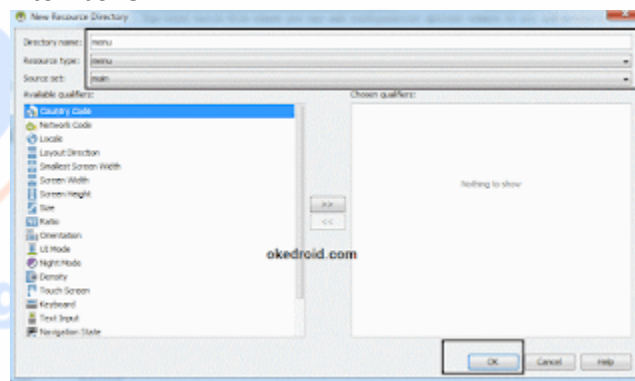
- ✍ Membuat folder menu

Pada folder res di struktur project Android Studio kalian, klik kanan lalu pilih **New > pilih Android Resource Directory** seperti pada gambar di bawah :



Android Resource Directory

Pilih menu lalu pilih tombol OK.



Buat folder menu resource

Buat file xlmnya saya memberikan nama `menu.xml` (terserah kalian) pada folder menu. Salin baris instruksi (codingan) di bawah :

menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/one"
        android:title="Rename"/>
    <item
        android:id="@+id/two"
        android:title="Delete"/>
    <item
        android:id="@+id/three"
        android:title="Copy"/>
</menu>
```

3. Pada file activity_main.xml. kalian bisa menerapkan baris instruksi (codingan) di bawah :

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="KLIK DISINI"
        android:id="@+id/button" android:layout_gravity="center_horizontal"/>
</LinearLayout>
```

4. Setelah itu kita coba terapkan pada class Activity di file MainActivity.java.

```
package com.okedroid.aplikasisaya;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.PopupMenu;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    // create variable
    Button btnclickme;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // load control
        btnclickme = (Button) findViewById(R.id.button);
        btnclickme.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {

                // Create the instance of Menu
                PopupMenu popup = new PopupMenu(MainActivity.this, btnclickme);

                // Inflating menu using xml file
                popup.getMenuInflater().inflate(R.menu.menu, popup.getMenu());
```



```
// registering OnMenuItemClickListener

popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {

    public boolean onMenuItemClick(MenuItem item) {

        Toast.makeText(MainActivity.this,

            "Kamu telah memilih : " + item.getTitle(),

            Toast.LENGTH_SHORT).show();

        return true;

    }

});

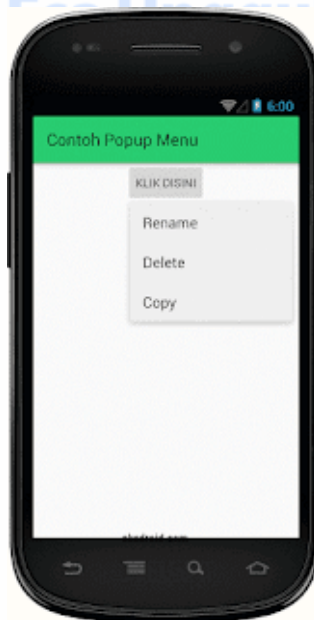
popup.show();

}

}

}
```

5. Setelah itu kita coba uji dan jalankan aplikasinya lewat Android Studio. Hasilnya kurang lebih seperti ini :



Hasil Popup Menu

Cara Memasang SplashScreen di Aplikasi Android

SplashScreen merupakan start screen (layar pembuka) pada aplikasi android, atau tampilan loading awal, saat aplikasi android di jalankan untuk pertama kali. Saat aplikasi android dibuat, splashscreen berguna untuk mempercantik aplikasi android kita agar terlihat profesional. Umumnya, splashscreen di gunakan untuk menampilkan logo-logo perusahaan, logo-logo nama developer.

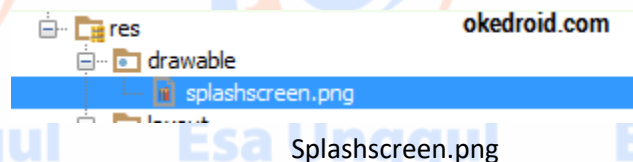
Pada modul android kali ini, kita akan mencoba membuat dan menerapkan SplashScreen pada aplikasi android.

1. Pertama buka Project di Android Studio kalian.
2. Siapkan terlebih dahulu gambar kalian, yang akan digunakan untuk Splashscreen. Sebagai contoh saya menggunakan gambar berikut dengan ukuran 480x800:

okedroid.com
-Blog Android and Teknologi-

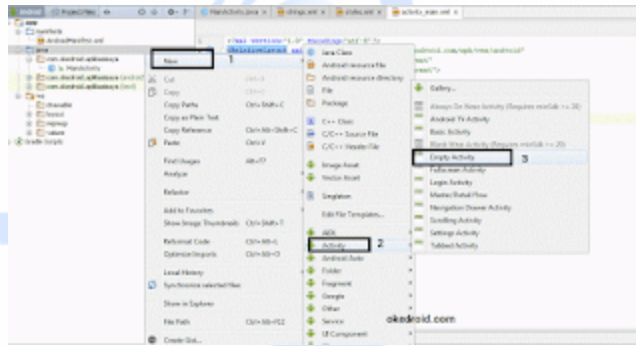
Contoh SplashScreen

3. Kemudian simpan gambar splashscreen, pada folder res/drawable, pada struktur folder project Android Studio kalian.



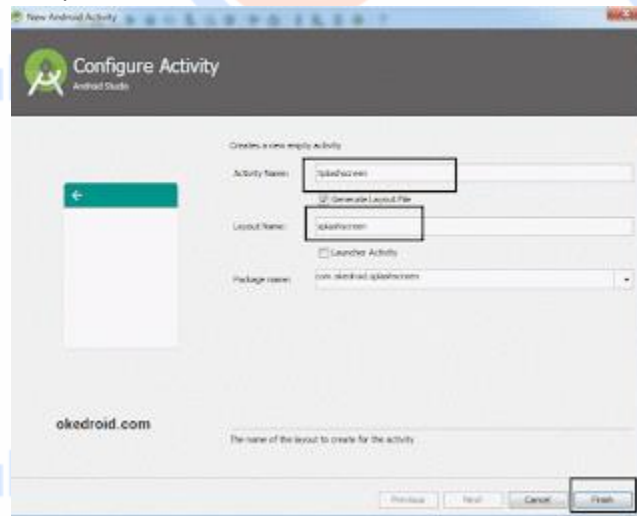
Splashscreen.png

4. Disini kita akan membuat Activity baru, untuk splashscreen yang terdiri dari file .java dan file .xml.
Pada folder java atau res kalian bisa klik kanan pilih New > pilih Activity > lalu EmptyActivity.



Membuat Activity Baru

Pada bagian New Android Activity kalian isikan nama activity atau file java dan nama layout (Terserah kalian), sebagai contoh saya akan membuat file class Activity dengan nama SplashScreen.java dan splashscreen.xml



Membuat Activity SplashScreen

- Setelah itu salin baris instruksi (codingan) di bawah pada file SplashScreen.java dan splashscreen.xml.

```
<?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" >

    <ImageView
        android:src="@drawable/splashscreen"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="fitXY"/>
```

```

<ProgressBar

    android:id="@+id/progressBar1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentBottom="true"

    android:layout_centerHorizontal="true"

    android:layout_marginBottom="58dp" />

</RelativeLayout>

```

Splashscreen.java

```

package com.okedroid.splashscreen;
import android.support.v7.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;
import android.view.WindowManager;
//Created by Fathur (okedroid.com)
public class Splashscreen extends AppCompatActivity {

    //Set waktu lama splashscreen
    private static int splashInterval = 2000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.splashscreen);
        new Handler().postDelayed(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub
                Intent i = new Intent(Splashscreen.this, MainActivity.class);
                startActivity(i); // menghubungkan activity splashscreen ke main activity dengan
                intent
                //jeda selesai Splashscreen
                this.finish();
            }
            private void finish() {
                // TODO Auto-generated method stub
            }
        }, splashInterval);
    }
}

```

6. Setelah disetting agar Activity Splashscreen, dapat dijalankan pertama kali saat Aplikasi dibuka, maka kita perlu mensetting pada bagian file (app/manifest/AndroidManifest.xml).
AndroidManifest.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.okedroid.splashscreen" >
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <!-- Splash screen -->
        <activity
            android:name="com.okedroid.splashscreen.Splashscreen"
            android:label="@string/app_name"
            android:theme="@style/Splashscreen"
            android:screenOrientation="portrait"
            android:noHistory="true" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <!-- Main activity -->
        <activity
            android:name="com.okedroid.splashscreen.MainActivity"
            android:label="@string/app_name" >
        </activity>
    </application>
</manifest>
```

7. Pada library AppCompatActivity, agar tampilan SplashScreen menjadi fullscreen, kita perlu mensettingnya pada bagian theme di res/values/styles.xml.

```
<resources>

    <!-- Base application theme. -->

    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

        <!-- Customize your theme here. -->

        <item name="colorPrimary">@color/colorPrimary</item>

        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>

        <item name="colorAccent">@color/colorAccent</item>

    </style>

    <style name="Splashscreen" parent="Theme.AppCompat.Light.DarkActionBar">

        <item name="windowActionBar">false</item>

        <item name="windowNoTitle">true</item>

    </style>

</resources>
```

8. Setelah itu kita coba jalankan Aplikasinya lewat Android Studio

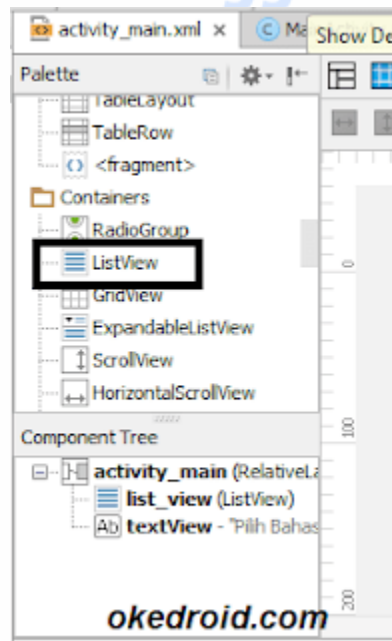
Cara membuat ListView sederhana di Android

listview adalah salah satu komponen User Interface pada aplikasi Android yang berfungsi menampilkan atau memuat data, dalam tampilan list/daftar yang memungkinkan kalian melihatnya bergulir secara vertikal di Aplikasi Android. ListView berguna jika kita membuat Aplikasi Android dan ingin menampilkan atau memuat data dalam suatu list misal daftar nama, nomor hp, nama-nama binatang dll.

ListView bekerja menggunakan Adapter yang berfungsi untuk binding data, memproses dan memformat konten (pull konten) dari resource seperti Array atau Database, kemudian di convert setiap item ke dalam tampilan list/daftar.

Nah pada artikel tutorial belajar Android kali ini, kita akan menampilkan daftar/list bahasa pemrograman, yang akan ditampilkan dalam bentuk ListView.

1. Pertama-pertama, kalian bisa buat project baru atau project yang sudah ada yang ingin diterapkan ListView di Android Studio kalian.
2. Lalu pada resource layout pada bagian Palette, sebelah kiri kalian bisa mendrag komponen ListView kedalam layout.



ListView Pallette

3. Maka hasilnya akan seperti ini

```

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

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:id="@+id/activity_main"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context="com.okedroid.simplelistview.MainActivity">

    <!--List View -->

    <ListView

        android:layout_width="match_parent"

        android:layout_height="match_parent"

        android:layout_marginTop="17dp"

        android:id="@+id/list_view"

    />

    <TextView

        android:text="Pilih Bahasa Pemrograman : "

        android:textStyle="bold"

        android:layout_marginBottom="16dp"

        android:layout_alignParentTop="true"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentBottom="true"

        android:layout_alignParentStart="true"

        android:id="@+id/textView"/>

</RelativeLayout>

```

4. Setelah itu untuk mengisi code logicnya kalian bisa menerapkan codingnya seperti ini : MainActivity.java

```
package com.okedroid.simplelistview;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

/*
 * Fathurrahman (okedroid.com)
 */

public class MainActivity extends AppCompatActivity {

    private ListView lvItem;

    private String[] bahasapemrograman = new String[]{
        "ASP.NET", "C++", "C#", "BASIC", "SQL",
        "PHP", "Phyton", "Javascript", "Java",
        "Visual Basic"};

    //mendeklarasikan listview var dan menginisiasi array tipe data string
    //Step 1
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle("ListView Sederhana"); //tampil title
        getSupportActionBar().setSubtitle("okedroid.com"); //tampil subtitle

        lvItem = (ListView) findViewById(R.id.list_view);

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(MainActivity.this,
            android.R.layout.simple_list_item_1, android.R.id.text1, bahasapemrograman);

        /*
```

Step 2

Membinding atau memformat data

```
*/
```

```
lvItem.setAdapter(adapter);
```

```
//menset data di dalam listview
```

```
//Step 3
```

```
lvItem.setOnItemClickListener(new AdapterView.OnItemClickListener() {
```

```
    @Override
```

```
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
```

```
        Toast.makeText(MainActivity.this, "Memilih : "+bahasapemrograman[position],  
        Toast.LENGTH_LONG).show();
```

```
        //memanggil set on Item ClickListener untuk Listview, jadi jika salah satu item list view  
        diklik akan
```

```
        //akan bereaksi menampilkan toast atau aksi lainnya.
```

```
        //Step 4
```

```
    }
```

```
});
```

```
}
```

```
}
```

5. Jalankan atau Run Aplikasinya di Android Studio, lalu lihat hasilnya dengan menggunakan Real Device atau Emulator.

Kira-kira hasilnya akan terlihat seperti ini :



Android - PHP / MYSQL

Dalam bab ini, kita akan menjelaskan, bagaimana Anda dapat mengintegrasikan PHP dan MYSQL dengan aplikasi android Anda. Ini sangat berguna jika Anda memiliki webserver, dan Anda ingin mengakses datanya di aplikasi android Anda.

MYSQL digunakan sebagai database di webserver dan PHP digunakan untuk mengambil data dari database. Aplikasi kami akan berkomunikasi dengan halaman PHP dengan parameter yang diperlukan dan PHP akan menghubungi database MYSQL dan akan mengambil hasilnya dan mengembalikan hasilnya kepada kami.

PHP - MYSQL Membuat Database

Database MYSQL dapat dibuat dengan mudah menggunakan skrip sederhana ini. Pernyataan **CREATE DATABASE** membuat database.

```
<?php
    $con=mysqli_connect("example.com","username","password");
    $sql="CREATE DATABASE my_db";
    if (mysqli_query($con,$sql)) {
        echo "Database my_db created successfully";
    }
?>
```

Membuat Tabel

Setelah database dibuat, waktunya untuk membuat beberapa tabel dalam database. Pernyataan **CREATE TABLE** membuat database.

```
<?php
    $con=mysqli_connect("example.com","username","password","my_db");
    $sql="CREATE TABLE table1(Username CHAR(30),Password CHAR(30),Role CHAR(30))";
    if (mysqli_query($con,$sql)) {
        echo "Table have been created successfully";
    }
```



```
}  
?>
```

Menyisipkan Nilai dalam tabel

Ketika database dan tabel dibuat. Sekarang saatnya memasukkan beberapa data ke dalam tabel. Pernyataan **Sisipkan Ke** membuat basis data.

```
<?php  
  
$con=mysqli_connect("example.com","username","password","my_db");  
  
$sql="INSERT INTO table1 (FirstName, LastName, Age) VALUES ('admin',  
'admin','administrator')";  
  
if (mysqli_query($con,$sql)) {  
    echo "Values have been inserted successfully";  
}  
  
?>
```

PHP - GET dan metode POST

PHP juga digunakan untuk mengambil catatan dari database mysql setelah dibuat. Untuk mengambil catatan beberapa informasi harus dilewatkan ke halaman PHP mengenai catatan apa yang harus diambil.

Metode pertama untuk menyampaikan informasi adalah melalui metode GET di mana perintah \$ **_GET** digunakan. Variabel dilewatkan dalam url dan catatan diambil. Sintaksnya diberikan di bawah ini -

```
<?php  
  
$con=mysqli_connect("example.com","username","password","database name");  
  
if (mysqli_connect_errno($con)) {  
    echo "Failed to connect to MySQL: " . mysqli_connect_error();  
}  
  
  
$username = $_GET['username'];
```

```

$password = $_GET['password'];

$result = mysqli_query($con,"SELECT Role FROM table1 where Username='$username'
    and Password='$password'");

$row = mysqli_fetch_array($result);

$data = $row[0];

if($data){
    echo $data;
}

mysqli_close($con);

?>

```

Metode kedua adalah menggunakan metode POST. Satu-satunya perubahan dalam skrip di atas adalah mengganti \$ _GET dengan \$ **_POST** . Dalam metode Post, variabel tidak dilewatkan melalui URL.

Android - Menghubungkan MYSQL

Menghubungkan Via Dapatkan Metode

Ada dua cara untuk terhubung ke MYSQL melalui halaman PHP. Yang pertama disebut **metode** **Get** . Kami akan menggunakan **HttpGet** dan **HttpClient** kelas untuk menghubungkan. Sintaks mereka diberikan di bawah ini -

```

URL url = new URL(link);

HttpClient client = new DefaultHttpClient();

HttpGet request = new HttpGet();

request.setURI(new URI(link));

```

Setelah itu Anda perlu memanggil **melaksanakan** metode HttpClient kelas dan menerimanya dalam sebuah objek HttpResponse. Setelah itu Anda perlu membuka aliran untuk menerima data.

```

HttpResponse response = client.execute(request);

```

```
BufferedReader in = new BufferedReader  
(new InputStreamReader(response.getEntity().getContent()));
```

Menghubungkan Via Post Method

Dalam metode Post, kelas **URLEncoder** , **URLConnection** akan digunakan. Para urlencoder akan menyandikan informasi dari variabel yang lewat. Ini sintaks diberikan di bawah ini -

```
URL url = new URL(link);  
  
String data = URLEncoder.encode("username", "UTF-8")  
+ "=" + URLEncoder.encode(username, "UTF-8");  
  
data += "&" + URLEncoder.encode("password", "UTF-8")  
+ "=" + URLEncoder.encode(password, "UTF-8");  
  
URLConnection conn = url.openConnection();
```

Hal terakhir yang perlu Anda lakukan adalah menulis data ini ke tautan. Setelah menulis, Anda perlu membuka aliran untuk menerima data yang direspon.

```
OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());  
  
wr.write( data );  
  
BufferedReader reader = new BufferedReader(new  
InputStreamReader(conn.getInputStream()));
```

Contoh

Contoh di bawah ini adalah contoh lengkap untuk menghubungkan aplikasi android Anda dengan database MYSQL melalui halaman PHP. Ini menciptakan aplikasi dasar yang memungkinkan Anda untuk masuk menggunakan metode GET dan POST.

PHP - bagian MYSQL

Dalam contoh ini, database dengan nama temp telah dibuat di 000webhost.com. Dalam database itu, sebuah tabel telah dibuat dengan nama tabel1. Tabel ini memiliki tiga bidang. (Nama Pengguna, Sandi,

Peran). Tabel hanya memiliki satu catatan yaitu ("admin", "admin", "administrator").

Halaman php telah diberikan di bawah ini yang mengambil parameter dengan metode posting.

```
<?php

$con=mysqli_connect("mysql10.000webhost.com","username","password","db_name");

if (mysqli_connect_errno($con)) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$username = $_POST['username'];
$password = $_POST['password'];
$result = mysqli_query($con,"SELECT Role FROM table1 where
Username='$username' and Password='$password'");
$row = mysqli_fetch_array($result);
$data = $row[0];

if($data){
    echo $data;
}

mysqli_close($con);

?>
```

Bagian Android

Untuk bereksperimen dengan contoh ini, Anda perlu menjalankan ini pada perangkat yang sebenarnya di mana wifi internet terhubung.

Tangga	Deskripsi
1	Anda akan menggunakan Android Studio IDE untuk membuat aplikasi Android dan menamakannya sebagai PHPMYSQL di bawah paket com.example.phpmysql.
2	Ubah file src / MainActivity.java untuk menambahkan kode Activity.
3	Buat file src / SiginActivity.java untuk menambahkan kode PHPMYSQL.
4	Ubah tata letak file XML res / layout / activity_main.xml tambahkan komponen GUI jika diperlukan.
5	Ubah file res / values / string.xml dan tambahkan komponen string yang diperlukan.
6	Ubah AndroidManifest.xml untuk menambahkan izin yang diperlukan.
7	Jalankan aplikasi dan pilih perangkat android yang berjalan dan instal aplikasi di atasnya dan verifikasi hasilnya.

Berikut adalah isi **src / com.example.phpmysql / MainActivity.java** .

```
package com.example.phpmysql;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends Activity {
```

```
private EditText usernameField,passwordField;

private TextView status,role,method;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    usernameField = (EditText)findViewById(R.id.editText1);
    passwordField = (EditText)findViewById(R.id.editText2);

    status = (TextView)findViewById(R.id.textView6);
    role = (TextView)findViewById(R.id.textView7);
    method = (TextView)findViewById(R.id.textView9);
}

public void login(View view){

    String username = usernameField.getText().toString();

    String password = passwordField.getText().toString();

    method.setText("Get Method");

    new SigninActivity(this,status,role,0).execute(username,password);

}

public void loginPost(View view){

    String username = usernameField.getText().toString();

    String password = passwordField.getText().toString();

    method.setText("Post Method");

    new SigninActivity(this,status,role,1).execute(username,password);

}
```



```
}
```

Berikut adalah isi **src / com.example.phpmysql / SigninActivity.java** .

```
package com.example.phpmysql;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URI;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import android.content.Context;
import android.os.AsyncTask;
import android.widget.TextView;

public class SigninActivity extends AsyncTask{

    private TextView statusField,roleField;

    private Context context;

    private int byGetOrPost = 0;

    //flag 0 means get and 1 means post.(By default it is get.)
```

```

    public SigninActivity(Context context,TextView statusField,TextView roleField,int
flag) {

        this.context = context;

        this.statusField = statusField;

        this.roleField = roleField;

        byGetOrPost = flag;

    }

    protected void onPreExecute(){

    }

    @Override

    protected String doInBackground(String... arg0) {

        if(byGetOrPost == 0){ //means by Get Method

            try{

                String username = (String)arg0[0];

                String password = (String)arg0[1];

                String link =
"http://myphpmysqlweb.hostei.com/login.php?username="+username+"&
password="+password;

                URL url = new URL(link);

                HttpClient client = new DefaultHttpClient();

                HttpGet request = new HttpGet();

                request.setURI(new URI(link));

                HttpResponse response = client.execute(request);

                BufferedReader in = new BufferedReader(new

                    InputStreamReader(response.getEntity().getContent()));

```

```

        StringBuffer sb = new StringBuffer("");
        String line="";

        while ((line = in.readLine()) != null) {
            sb.append(line);
            break;
        }

        in.close();
        return sb.toString();
    } catch (Exception e){
        return new String("Exception: " + e.getMessage());
    }
} else{
    try{
        String username = (String)arg0[0];
        String password = (String)arg0[1];

        String link="http://myphpmysqlweb.hostei.com/loginpost.php";
        String data  = URLEncoder.encode("username", "UTF-8") + "=" +
            URLEncoder.encode(username, "UTF-8");
        data += "&" + URLEncoder.encode("password", "UTF-8") + "=" +
            URLEncoder.encode(password, "UTF-8");

        URL url = new URL(link);
        URLConnection conn = url.openConnection();
    }
}

```

```

conn.setDoOutput(true);

OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());

wr.write( data );
wr.flush();

BufferedReader reader = new BufferedReader(new
    InputStreamReader(conn.getInputStream()));

StringBuilder sb = new StringBuilder();
String line = null;

// Read Server Response
while((line = reader.readLine()) != null) {
    sb.append(line);
    break;
}

return sb.toString();
} catch(Exception e){
    return new String("Exception: " + e.getMessage());
}
}

@Override
protected void onPostExecute(String result){
    this.statusField.setText("Login Successful");

```

```
        this.roleField.setText(result);  
    }  
}
```

Menambahkan konten berikut untuk **build.gradle** dan membangun kembali seluruh proyek.

```
android {  
    useLibrary 'org.apache.http.legacy'  
}
```

Berikut adalah isi **activity_main.xml** .

```
<?xml version="1.0" encoding="utf-8"?>  
  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"  
    tools:context=".MainActivity" >  
  
    <EditText  
        android:id="@+id/editText2"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_alignRight="@+id/editText1"  
        android:layout_below="@+id/editText1"  
        android:layout_marginTop="25dp"
```

```
        android:ems="10"

        android:inputType="textPassword" >

</EditText>

<EditText

    android:id="@+id/editText1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentRight="true"

    android:layout_alignParentTop="true"

    android:layout_marginTop="44dp"

    android:ems="10" >

<requestFocus android:layout_width="wrap_content" />

</EditText>

<TextView

    android:id="@+id/textView1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignBottom="@+id/editText1"

    android:layout_alignParentLeft="true"

    android:text="@string/Username" />

<TextView

    android:id="@+id/textView3"

    android:layout_width="wrap_content"
```



```
android:layout_height="wrap_content"

android:layout_alignParentTop="true"

android:layout_centerHorizontal="true"

android:text="@string/App"

android:textAppearance="?android:attr/textAppearanceLarge" />
```

<TextView

```
android:id="@+id/textView7"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignBottom="@+id/textView5"

android:layout_alignLeft="@+id/textView6"

android:text="@string/Role"

android:textAppearance="?android:attr/textAppearanceMedium"

android:textSize="10sp" />
```

<TextView

```
android:id="@+id/textView5"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_below="@+id/textView6"

android:layout_marginTop="27dp"

android:layout_toLeftOf="@+id/editText1"

android:text="@string/LoginRole" />
```

<TextView

```
android:id="@+id/textView8"

android:layout_width="wrap_content"

android:layout_height="wrap_content"
```

```
android:layout_above="@+id/textView6"

android:layout_alignLeft="@+id/textView5"

android:layout_marginBottom="27dp"

android:text="@string/method" />
```

<TextView

```
android:id="@+id/textView4"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignLeft="@+id/textView8"

android:layout_below="@+id/button1"

android:layout_marginTop="86dp"

android:text="@string/LoginStatus" />
```

<TextView

```
android:id="@+id/textView6"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignTop="@+id/textView4"

android:layout_centerHorizontal="true"

android:text="@string/Status"

android:textAppearance="?android:attr/textAppearanceMedium"

android:textSize="10sp" />
```

<TextView

```
android:id="@+id/textView9"

android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"

android:layout_alignBottom="@+id/textView8"

android:layout_alignLeft="@+id/textView6"

android:text="@string/Choose"

android:textAppearance="?android:attr/textAppearanceMedium"

android:textSize="10sp" />
```

<Button

```
android:id="@+id/button2"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_centerVertical="true"

android:layout_toRightOf="@+id/textView6"

android:onClick="loginPost"

android:text="@string/LoginPost" />
```

<Button

```
android:id="@+id/button1"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignBaseline="@+id/button2"

android:layout_alignBottom="@+id/button2"

android:layout_alignLeft="@+id/textView2"

android:onClick="login"

android:text="@string/LoginGet" />
```

<TextView

```
android:id="@+id/textView2"
```

```
        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignBaseline="@+id/editText2"

        android:layout_alignBottom="@+id/editText2"

        android:layout_alignParentLeft="true"

        android:text="@string/Password" />
```

```
</RelativeLayout>
```

Berikut adalah isi **Strings.xml** .

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

<resources>

    <string name="app_name">PHPMYSQL</string>

    <string name="action_settings">Settings</string>

    <string name="hello_world">Hello world!</string>

    <string name="Username">Username</string>

    <string name="Password">Password</string>

    <string name="LoginGet">Login - Get</string>

    <string name="LoginPost">Login - Post</string>

    <string name="App">Login Application</string>

    <string name="LoginStatus">Login Status</string>

    <string name="LoginRole">Login Role</string>

    <string name="Status">Not login</string>

    <string name="Role">Not assigned</string>

    <string name="method">Login Method</string>

    <string name="Choose">Choose Method</string>
```

</resources>

Berikut adalah konten **AndroidManifest.xml** .

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.phpmysql" >

    <uses-permission android:name="android.permission.INTERNET"/>

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >


        <activity
            android:name="com.example.phpmysql.MainActivity"
            android:label="@string/app_name" >

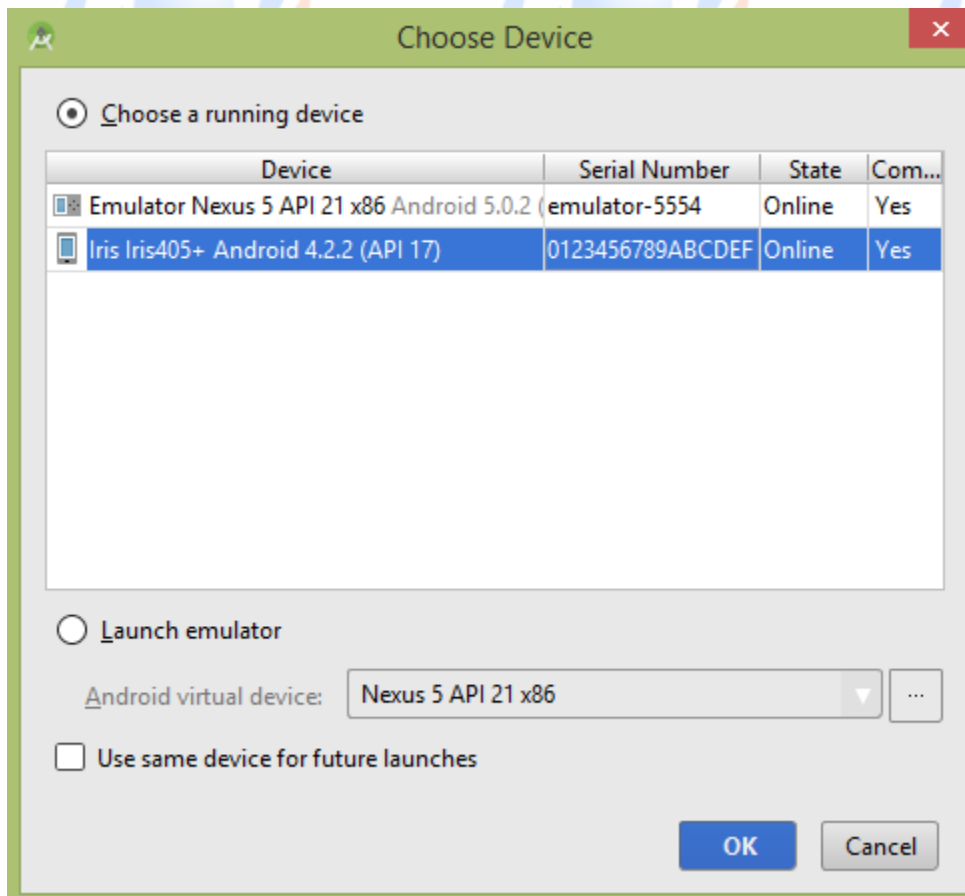
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>


    </application>
```

</manifest>

Mari coba jalankan aplikasi PHPMYSQL Anda. Saya berasumsi Anda telah menghubungkan perangkat Android Mobile Anda yang sebenarnya dengan komputer Anda. Untuk menjalankan aplikasi dari studio Android, buka salah satu file aktivitas proyek Anda dan klik Jalankan  ikon dari bilah alat. Sebelum memulai aplikasi Anda, studio Android akan menampilkan jendela berikut untuk memilih opsi di mana Anda ingin menjalankan aplikasi Android Anda.



Pilih perangkat seluler Anda sebagai opsi dan kemudian periksa perangkat seluler Anda yang akan menampilkan layar berikut -



12:43 AM

PHPMYSQL

Login Application

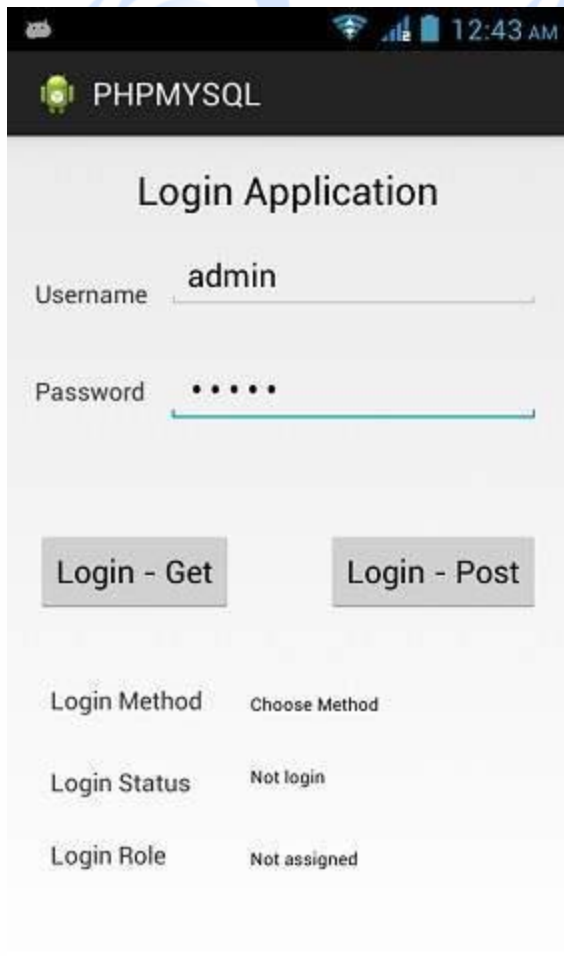
Username

Password

Login - Get Login - Post

Login Method	Choose Method
Login Status	Not login
Login Role	Not assigned

Sekarang cukup ketik nama pengguna dan kata sandi Anda. Dalam kasus saya, saya menetik admin sebagai nama pengguna dan kata sandi. Ini ditunjukkan pada gambar -



PHPMYSQL

Login Application

Username

Password

Login Method	Choose Method
Login Status	Not login
Login Role	Not assigned

Sekarang tekan tombol Get dan tunggu beberapa detik dan respons akan diunduh dan akan ditampilkan kepada Anda. Dalam hal ini, responsnya adalah ROLE yang diambil dalam kasus admin sebagai nama pengguna dan kata sandi. Hal ini ditunjukkan pada gambar di bawah

The screenshot shows a mobile application interface with a status bar at the top displaying the time as 12:42 AM. The app title is 'PHPMYSQL'. The main section is titled 'Login Application'. It contains two input fields: 'Username' with the value 'admin' and 'Password' with masked characters '.....'. Below these fields are two buttons: 'Login - Get' and 'Login - Post'. The 'Login - Post' button is highlighted. At the bottom, there are three rows of text: 'Login Method' with the value 'Get Method', 'Login Status' with the value 'Login Successful', and 'Login Role' with the value 'administrator'.

Field	Value
Username	admin
Password
Login Method	Get Method
Login Status	Login Successful
Login Role	administrator

Sekarang lagi tekan tombol POST dan hasil yang sama akan muncul. Ini ditunjukkan pada gambar di bawah -

12:43 AM

PHPMYSQL

Login Application

Username

Password

Login Method

Login Status

Login Role

Android - Google Maps

Android memungkinkan kami mengintegrasikan peta google dalam aplikasi kami. Anda dapat menunjukkan lokasi apa pun di peta, atau dapat menunjukkan rute yang berbeda di peta dll. Anda juga dapat menyesuaikan peta sesuai dengan pilihan Anda.

Google Map - File tata letak

Sekarang Anda harus menambahkan fragmen peta ke dalam file layout xml. Sintaksnya diberikan di bawah ini -

```
<fragment
    android:id="@+id/map"

    android:name="com.google.android.gms.maps.MapFragment"

    android:layout_width="match_parent"

    android:layout_height="match_parent"/>
```

Google Map - file AndroidManifest

Hal berikutnya yang perlu Anda lakukan adalah menambahkan beberapa izin bersama dengan kunci Google Map API di file AndroidManifest.XML. Sintaksnya diberikan di bawah ini -

```
<!--Permissions-->

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="com.google.android.providers.gsf.permission.
    READ_GSERVICES" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<!--Google MAP API key-->

<meta-data
```

```
android:name="com.google.android.maps.v2.API_KEY"

android:value="AIzaSyDKymeBXNeiFWY5jRUejv6zItpmr2MVyQ0" />
```

Menyesuaikan Google Map

Anda dapat dengan mudah menyesuaikan peta google dari tampilan defaultnya, dan mengubahnya sesuai dengan permintaan Anda.

Menambahkan Marker

Anda dapat menempatkan pembuat dengan beberapa teks di atasnya yang menampilkan lokasi Anda di peta. Ini bisa dilakukan dengan melalui metode **addMarker ()** . Sintaksnya diberikan di bawah ini -

```
final LatLng TutorialPoint = new LatLng(21 , 57);

Marker TP = googleMap.addMarker(new MarkerOptions()

    .position(TutorialPoint).title("TutorialPoint"));
```

Mengejar Jenis Peta

Anda juga dapat mengubah jenis MAP. Ada empat jenis peta yang berbeda dan masing-masing memberikan tampilan peta yang berbeda. Jenis-jenis ini Normal, Hybrid, Satelit dan medan. Anda dapat menggunakannya seperti di bawah ini

```
googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);

googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);

googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

Aktifkan / Nonaktifkan zoom

Anda juga dapat mengaktifkan atau menonaktifkan gerakan zoom di peta dengan memanggil metode **setZoomControlsEnabled (boolean)** . Sintaksnya diberikan di bawah ini -

```
googleMap.getUiSettings().setZoomGesturesEnabled(true);
```


Terlepas dari penyesuaian ini, ada metode lain yang tersedia di kelas GoogleMap, yang membantu Anda lebih menyesuaikan peta. Mereka tercantum di bawah -

Sr.No	Metode & deskripsi
1	addCircle (pilihan CircleOptions) Metode ini menambahkan lingkaran ke peta
2	addPolygon (opsi PolygonOptions) Metode ini menambahkan poligon ke peta
3	addTileOverlay (opsi TileOverlayOptions) Metode ini menambahkan overlay ubin ke peta
4	animateCamera (Pembaruan CameraUpdate) Metode ini Memindahkan peta sesuai dengan pembaruan dengan animasi
5	bersih() Metode ini menghapus semuanya dari peta.
6	getMyLocation () Metode ini mengembalikan lokasi pengguna yang saat ini ditampilkan.
7	moveCamera (Pembaruan CameraUpdate) Metode ini mereposisi kamera sesuai dengan petunjuk yang ditentukan dalam pembaruan
8	setTrafficEnabled (diaktifkan boolean) Metode ini Mengaktifkan atau menonaktifkan lapisan lalu lintas.

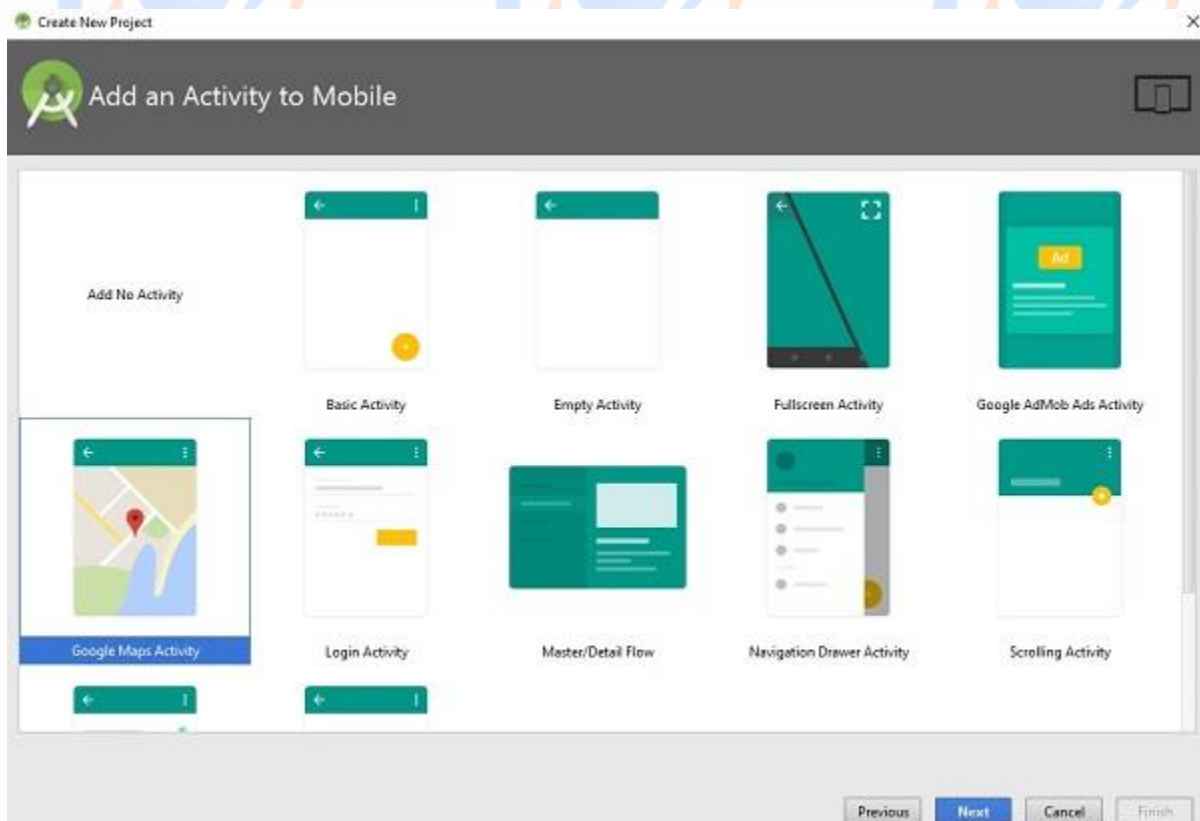
9	snapshot (GoogleMap.SnapshotReadyCallback callback) Metode ini Membawa snapshot dari peta
10	stopAnimation () Metode ini menghentikan animasi kamera jika ada yang sedang berlangsung

Contoh

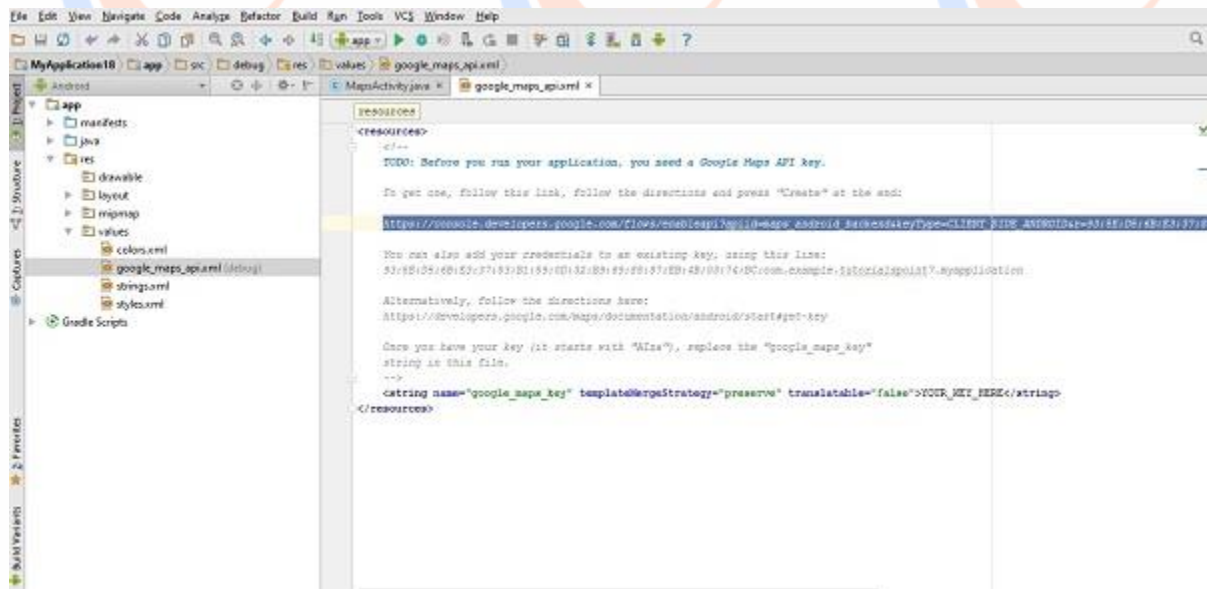
Berikut ini contoh yang menunjukkan penggunaan kelas GoogleMap. Ini menciptakan aplikasi M dasar yang memungkinkan Anda untuk menavigasi melalui peta.

Untuk bereksperimen dengan contoh ini, Anda dapat menjalankan ini di perangkat yang sebenarnya atau di emulator.

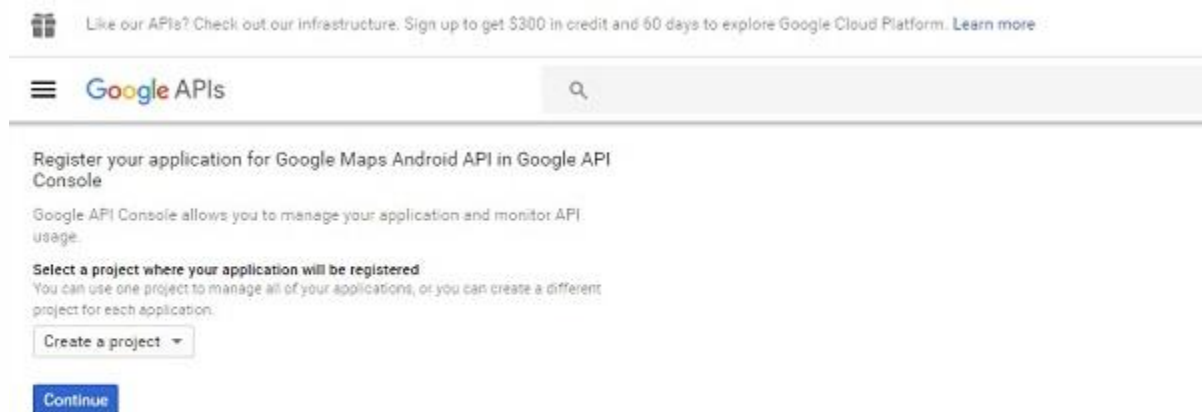
Buat proyek dengan aktivitas google maps seperti yang ditunjukkan di bawah ini -



Ini akan membuka layar berikut dan menyalin url konsol untuk Kunci API seperti yang ditunjukkan di bawah ini -



Salin ini dan tempel ke peramban Anda. Ini akan memberikan layar berikut -



Klik lanjutkan dan klik pada Create API Key maka akan muncul layar berikut

API key created

Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key

AIzaSyAXhBdyKxUo_cb-EkSgWJQTdqR0QjLcques

⚠ Restrict your key to prevent unauthorized use in production.

Close

Restrict key

Berikut adalah isi **activity_main.xml** .

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.tutorialspoint7.myapplication.MapsActivity" />
```

Berikut adalah isi **MapActivity.java** .

Dalam kode di bawah ini kami telah memberikan contoh detail garis lintang dan garis bujur

```
package com.example.tutorialspoint7.myapplication;

import android.support.v4.app.FragmentActivity;
import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
```

```

import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);

        // Obtain the SupportMapFragment and get notified when the map is ready to be
        // used.
        SupportMapFragment mapFragment = (SupportMapFragment)
            getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    /**
     * Manipulates the map once available.
     *
     * This callback is triggered when the map is ready to be used.
     *
     * This is where we can add markers or lines, add listeners or move the camera.
     *
     * In this case, we just add a marker near Sydney, Australia.
     *
     * If Google Play services is not installed on the device.
     *
     * This method will only be triggered once the user has installed
     * Google Play services and returned to the app.
     */
}

```

```

@Override

public void onMapReady(GoogleMap googleMap) {

    mMap = googleMap;

    // Add a marker in Sydney and move the camera
    LatLng Tutorialspoint = new LatLng(21, 57);

    mMap.addMarker(new

        MarkerOptions().position(Tutorialspoint).title("Tutorialspoint.com"));

    mMap.moveCamera(CameraUpdateFactory.newLatLng(Tutorialspoint));

}

}

```

Berikut ini adalah isi file **AndroidManifest.xml** .

```

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.tutorialspoint7.myapplication">

    <!--

        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use

        Google Maps Android API v2, but you must specify either coarse or fine

        location permissions for the 'MyLocation' functionality.

    -->

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <uses-permission android:name="android.permission.INTERNET" />

    <application

        android:allowBackup="true"

        android:icon="@mipmap/ic_launcher"

```

```
android:label="@string/app_name"
```

```
android:supportsRtl="true"
```

```
android:theme="@style/AppTheme">
```

```
<!--
```

The API key for Google Maps-based APIs is defined as a string resource.

(See the file "res/values/google_maps_api.xml").

Note that the API key is linked to the encryption key used to sign the APK.

You need a different API key for each encryption key, including the release
key

that is used to sign the APK for publishing.

You can define the keys for the debug and

release targets in src/debug/ and src/release/.

```
-->
```

```
<meta-data
```

```
    android:name="com.google.android.geo.API_KEY"
```

```
    android:value="AIzaSyAXhBdyKxUo_cb-EkSgWJQTdqR0QjLcques" />
```

```
<activity
```

```
    android:name=".MapsActivity"
```

```
    android:label="@string/title_activity_maps">
```

```
    <intent-filter>
```

```
        <action android:name="android.intent.action.MAIN" />
```

```
        <category android:name="android.intent.category.LAUNCHER" />
```

```
    </intent-filter>
```

```
</activity>
```

```
</application>
```


</manifest>

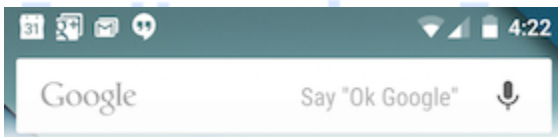
Output harus seperti ini -



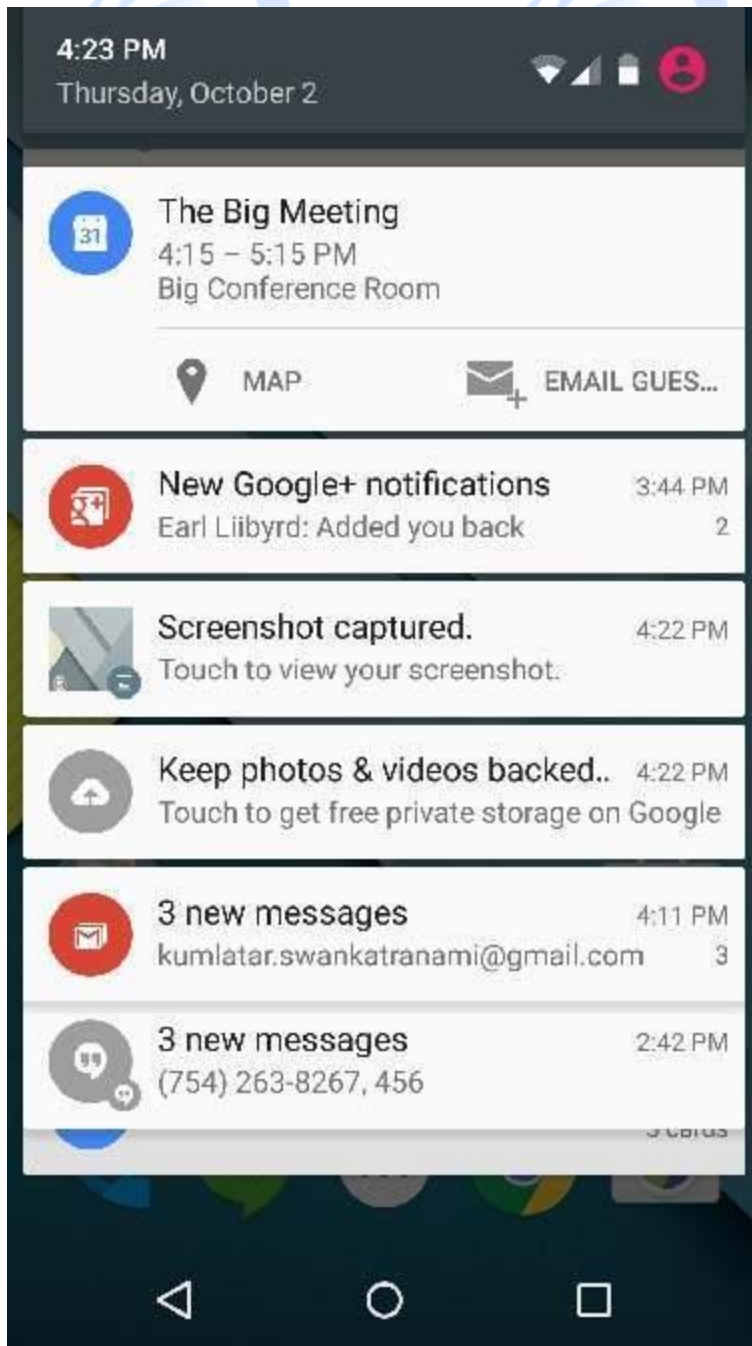
Android - Notifikasi

Sebuah **pemberitahuan** adalah pesan Anda dapat menampilkan kepada pengguna di luar UI yang normal aplikasi Anda. Saat Anda memberi tahu sistem untuk mengeluarkan pemberitahuan, itu pertama kali muncul sebagai ikon di area notifikasi. Untuk melihat detail pemberitahuan, pengguna membuka laci notifikasi. Area notifikasi dan laci notifikasi adalah area yang dikontrol sistem yang dapat dilihat oleh pengguna kapan saja.

Kelas Android **Toast** menyediakan cara yang mudah untuk menampilkan peringatan pengguna, tetapi masalahnya adalah bahwa peringatan ini tidak terus-menerus yang berarti peringatan berkedip di layar selama beberapa detik dan kemudian menghilang.



Untuk melihat rincian pemberitahuan, Anda harus memilih ikon yang akan menampilkan laci notifikasi memiliki detail tentang pemberitahuan tersebut. Saat bekerja dengan emulator dengan perangkat virtual, Anda harus mengklik dan menyeret turun bilah status untuk memperluasnya yang akan memberi Anda detail sebagai berikut. Ini hanya akan **64 dp** tinggi dan disebut pandangan normal.



Di atas formulir yang diperluas dapat memiliki **Tampilan Besar** yang akan memiliki detail tambahan tentang pemberitahuan tersebut. Anda dapat menambahkan hingga enam baris tambahan di notifikasi. Cuplikan layar berikut menunjukkan pemberitahuan tersebut.

Buat dan Kirim Pemberitahuan

Anda memiliki cara sederhana untuk membuat pemberitahuan. Ikuti langkah-langkah berikut di aplikasi Anda untuk membuat pemberitahuan -

Langkah 1 - Buat Pembuat Pemberitahuan

Sebagai langkah pertama adalah membuat pembuat notifikasi menggunakan `NotificationCompat.Builder.build()`. Anda akan menggunakan Pembuat Notifikasi untuk menyetel berbagai properti Pemberitahuan seperti ikon kecil dan besar, judul, prioritas, dll.

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this)
```

Langkah 2 - Mengatur Properti Pemberitahuan

Setelah Anda memiliki objek **Builder**, Anda dapat mengatur properti Pemberitahuan menggunakan objek Builder sesuai kebutuhan Anda. Tapi ini wajib untuk mengatur setidaknya mengikuti -

- Ikon kecil, ditetapkan oleh **setSmallIcon()**
- Judul, ditetapkan oleh **setContentTitle()**
- Teks detail, ditetapkan oleh **setContentText()**

```
mBuilder.setSmallIcon(R.drawable.notification_icon);  
mBuilder.setContentTitle("Notification Alert, Click Me!");  
mBuilder.setContentText("Hi, This is Android Notification Detail!");
```

Anda memiliki banyak properti opsional yang dapat Anda atur untuk pemberitahuan Anda. Untuk mempelajari lebih lanjut tentang mereka, lihat dokumentasi referensi untuk `NotificationCompat.Builder`.

Langkah 3 - Lampirkan Tindakan

Ini adalah bagian opsional dan diperlukan jika Anda ingin melampirkan tindakan dengan pemberitahuan. Suatu tindakan memungkinkan pengguna untuk langsung dari pemberitahuan ke **Kegiatan** di aplikasi Anda, di mana mereka dapat melihat satu atau lebih peristiwa atau melakukan pekerjaan lebih lanjut.

Tindakan didefinisikan oleh **PendingIntent** yang berisi **Intent** yang memulai suatu Kegiatan dalam aplikasi Anda. Untuk mengaitkan `PendingIntent` dengan isyarat, panggil metode `NotificationCompat.Builder` yang sesuai. Misalnya, jika Anda ingin memulai Aktivitas ketika pengguna

mengklik teks pemberitahuan di laci notifikasi, Anda menambahkan `PendingIntent` dengan memanggil **`setContentIntent()`**.

Objek `PendingIntent` membantu Anda melakukan tindakan atas nama aplikasi Anda, seringkali di lain waktu, tanpa peduli apakah aplikasi Anda berjalan atau tidak.

Kami mengambil bantuan objek `stack builder` yang akan berisi tumpukan belakang buatan untuk Kegiatan yang dimulai. Ini memastikan bahwa menavigasi mundur dari Kegiatan mengarah keluar dari aplikasi Anda ke layar Utama.

```
Intent resultIntent = new Intent(this, ResultActivity.class);
TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
stackBuilder.addParentStack(ResultActivity.class);

// Adds the Intent that starts the Activity to the top of the stack
stackBuilder.addNextIntent(resultIntent);
PendingIntent resultPendingIntent =
    stackBuilder.getPendingIntent(0, PendingIntent.FLAG_UPDATE_CURRENT);
mBuilder.setContentIntent(resultPendingIntent);
```

Langkah 4 - Terbitkan pemberitahuan

Akhirnya, Anda melewati objek Pemberitahuan ke sistem dengan memanggil `NotificationManager.notify()` untuk mengirim pemberitahuan Anda. Pastikan Anda memanggil metode **`NotificationCompat.Builder.build()`** pada objek builder sebelum memberi tahu. Metode ini menggabungkan semua opsi yang telah ditetapkan dan mengembalikan objek **Notifikasi** baru.

```
NotificationManager mNotificationManager = (NotificationManager)
    getSystemService(Context.NOTIFICATION_SERVICE);

// notificationID allows you to update the notification later on.

mNotificationManager.notify(notificationID, mBuilder.build());
```

Kelas `NotificationCompat.Builder`

Kelas `NotificationCompat.Builder` memungkinkan kontrol lebih mudah atas semua bendera, serta membantu membangun tata letak pemberitahuan yang khas. Berikut adalah beberapa metode penting dan paling sering digunakan yang tersedia sebagai bagian dari kelas `NotificationCompat.Builder`.

Sr.No.	Konstanta & Deskripsi
1	<p>Pembuatan notifikasi ()</p> <p>Gabungkan semua opsi yang telah ditetapkan dan kembalikan objek Notifikasi baru.</p>
2	<p>NotificationCompat.Builder setAutoCancel (boolean autoCancel)</p> <p>Pengaturan bendera ini akan membuatnya sehingga notifikasi secara otomatis dibatalkan ketika pengguna mengkliknya di panel.</p>
3	<p>NotificationCompat.Builder setContent (Tampilan RemoteViews)</p> <p>Sediakan RemoteViews khusus untuk digunakan, bukan yang standar.</p>
4	<p>NotificationCompat.Builder setContentInfo (Info CharSequence)</p> <p>Setel teks besar di sisi kanan notifikasi.</p>
5	<p>NotificationCompat.Builder setPendingIntent (PendingIntent intent)</p> <p>Berikan PendingIntent untuk dikirim ketika pemberitahuan diklik.</p>
6	<p>NotificationCompat.Builder setContentText (teks CharSequence)</p> <p>Atur teks (baris kedua) pemberitahuan, dalam pemberitahuan standar.</p>
7	<p>NotificationCompat.Builder setContentTitle (judul CharSequence)</p> <p>Atur teks (baris pertama) dari pemberitahuan, dalam pemberitahuan standar.</p>
8	<p>NotificationCompat.Builder setDefaults (default int)</p> <p>Setel opsi pemberitahuan default yang akan digunakan.</p>

9	NotificationCompat.Builder setLargeIcon (ikon Bitmap) Atur ikon besar yang ditampilkan di ticker dan notifikasi.
10	NotificationCompat.Builder setNumber (nomor int) Setel nomor besar di sisi kanan notifikasi.
11	NotificationCompat.Builder setOngoing (boolean sedang berlangsung) Setel apakah ini pemberitahuan berkelanjutan.
12	NotificationCompat.Builder setSmallIcon (ikon int) Atur ikon kecil untuk digunakan dalam tata letak notifikasi.
13	NotificationCompat.Builder setStyle (NotificationCompat.Style style) Tambahkan gaya pemberitahuan yang kaya untuk diterapkan pada waktu pembuatan.
14	NotificationCompat.Builder setTicker (CharSequence tickerText) Atur teks yang ditampilkan di bilah status saat pemberitahuan pertama kali tiba.
15	NotificationCompat.Builder setVibrate (long [] pattern) Setel pola getaran untuk digunakan.
16	NotificationCompat.Builder setWhen (lama kapan) Atur waktu saat peristiwa itu terjadi. Notifikasi di panel diurutkan berdasarkan waktu ini.

Contoh

Contoh berikut menunjukkan fungsi pemberitahuan Android menggunakan **NotificationCompat.Builder** Class yang telah diperkenalkan di Android 4.1.

Langkah	Deskripsi
1	Anda akan menggunakan Android Studio IDE untuk membuat aplikasi Android dan beri nama sebagai <i>tutorialspoint</i> di bawah paket <i>com.example.notificationdemo</i> .
2	Ubah file <i>src / MainActivity.java</i> dan tambahkan kode untuk memberi tahu (""), jika pengguna mengklik tombolnya, itu akan memanggil layanan notifikasi android.
3	Buat file Java <i>src / NotificationView.java</i> baru , yang akan digunakan untuk menampilkan tata letak baru sebagai bagian dari aktivitas baru yang akan dimulai ketika pengguna akan mengklik salah satu notifikasi
4	Ubah tata letak file XML <i>res / layout / activity_main.xml</i> untuk menambahkan tombol Notifikasi dalam tata letak relatif.
5	Buat file XML tata letak baru <i>res / layout / notification.xml</i> . Ini akan digunakan sebagai file layout untuk aktivitas baru yang akan dimulai ketika pengguna akan mengklik salah satu notifikasi.
6	Tidak perlu mengubah konstanta string default. Studio Android menangani konstanta string standar
7	Jalankan aplikasi untuk meluncurkan emulator Android dan verifikasi hasil dari perubahan yang dilakukan dalam aplikasi.

Berikut ini adalah isi dari file aktivitas utama yang dimodifikasi **src / com.example.notificationdemo / MainActivity.java** . File ini dapat mencakup setiap metode siklus hidup dasar.

```
package com.example.notificationdemo;
```

```

import android.app.Activity;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.support.v4.app.NotificationCompat;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

    Button b1;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        b1 = (Button)findViewById(R.id.button);
        b1.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                addNotification();

            }

        });

    }

    private void addNotification() {

        NotificationCompat.Builder builder =

```

```

        new NotificationCompat.Builder(this)

            .setSmallIcon(R.drawable.abc)

            .setContentTitle("Notifications Example")

            .setContentText("This is a test notification");

        Intent notificationIntent = new Intent(this, MainActivity.class);

        PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
notificationIntent,

            PendingIntent.FLAG_UPDATE_CURRENT);

        builder.setContentIntent(contentIntent);

        // Add as notification

        NotificationManager manager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

        manager.notify(0, builder.build());

    }
}

```

Berikut ini adalah isi dari file **res / layout / notification.xml** -

```

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:orientation="vertical"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent" >

    <TextView

        android:layout_width="fill_parent"

        android:layout_height="400dp"

        android:text="Hi, Your Detailed notification view goes here...." />

```

```
</LinearLayout>
```

Berikut ini adalah isi dari file aktivitas utama yang dimodifikasi **src / com.example.notificationdemo / NotificationView.java** .

```
package com.example.notificationdemo;

import android.os.Bundle;
import android.app.Activity;

public class NotificationView extends Activity{

    @Override

    public void onCreate(Bundle savedInstanceState){

        super.onCreate(savedInstanceState);

        setContentView(R.layout.notification);

    }

}
```

Berikut ini adalah isi dari file **res / layout / activity_main.xml** -

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:paddingBottom="@dimen/activity_vertical_margin"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    tools:context="MainActivity">

    <TextView
```

```
android:id="@+id/textView1"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Notification Example"

android:layout_alignParentTop="true"

android:layout_centerHorizontal="true"

android:textSize="30dp" />
```

<TextView

```
android:id="@+id/textView2"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Tutorials point "

android:textColor="#ff87ff09"

android:textSize="30dp"

android:layout_below="@+id/textView1"

android:layout_centerHorizontal="true"

android:layout_marginTop="48dp" />
```

<ImageButton

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:id="@+id/imageButton"

android:src="@drawable/abc"

android:layout_below="@+id/textView2"

android:layout_centerHorizontal="true"

android:layout_marginTop="42dp" />
```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Notification"
    android:id="@+id/button"
    android:layout_marginTop="62dp"
    android:layout_below="@+id/imageButton"
    android:layout_centerHorizontal="true" />

</RelativeLayout>

```

Berikut ini adalah isi **res / values / strings.xml** untuk mendefinisikan dua konstanta baru -

```

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

<resources>

    <string name="action_settings">Settings</string>

    <string name="app_name">tutorialspoint </string>

</resources>

```

Berikut adalah konten default **AndroidManifest.xml** -

```

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.notificationdemo" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

```

```

<activity
    android:name="com.example.notificationdemo.MainActivity"
    android:label="@string/app_name" >

    <intent-filter>

        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />

    </intent-filter>

</activity>

<activity android:name=".NotificationView"
    android:label="Details of notification"
    android:parentActivityName=".MainActivity">


    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity"/>

</activity>

</application>

</manifest>

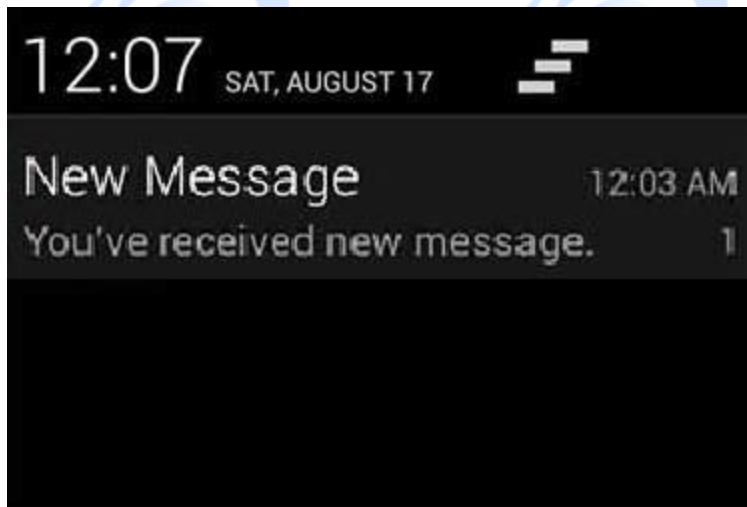
```

Mari mencoba menjalankan aplikasi **tutorialspoint** Anda . Saya berasumsi Anda telah membuat **AVD** Anda saat melakukan pengaturan lingkungan. Untuk menjalankan APP dari Android Studio, buka salah satu file aktivitas proyek Anda dan klik Jalankan  dari bilah alat. Android Studio menginstal aplikasi pada AVD Anda dan memulainya dan jika semuanya baik-baik saja dengan pengaturan dan aplikasi Anda, itu akan menampilkan jendela Emulator berikut -



Sekarang **tombol** klik , Anda akan melihat di bagian atas pesan "Pemberitahuan Pesan Baru!" akan ditampilkan sebentar dan setelah itu Anda akan memiliki layar berikut yang memiliki ikon kecil di sudut kiri atas.

Sekarang mari kita memperluas tampilan, klik lama pada ikon kecil, setelah beberapa detik akan menampilkan informasi tanggal dan ini adalah waktu ketika Anda harus menyeret bar status ke bawah tanpa melepaskan mouse. Anda akan melihat status bar akan mengembang dan Anda akan mendapatkan layar berikut -



Pemberitahuan Tampilan Besar

Cuplikan kode berikut menunjukkan cara mengubah pemberitahuan yang dibuat di cuplikan sebelumnya untuk menggunakan gaya tampilan Inbox yang besar. Saya akan memperbarui metode modifikasi `displayNotification()` untuk menunjukkan fungsi ini -

```
protected void displayNotification() {  
    Log.i("Start", "notification");  
  
    /* Invoking the default notification service */  
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this);  
  
    mBuilder.setContentTitle("New Message");  
    mBuilder.setContentText("You've received new message.");  
    mBuilder.setTicker("New Message Alert!");  
    mBuilder.setSmallIcon(R.drawable.woman);  
  
    /* Increase notification number every time a new notification arrives */  
    mBuilder.setNumber(++numMessages);  
  
    /* Add Big View Specific Configuration */  
}
```

```
NotificationCompat.InboxStyle inboxStyle = new NotificationCompat.InboxStyle();

String[] events = new String[6];
events[0] = new String("This is first line....");
events[1] = new String("This is second line...");
events[2] = new String("This is third line...");
events[3] = new String("This is 4th line...");
events[4] = new String("This is 5th line...");
events[5] = new String("This is 6th line...");

// Sets a title for the Inbox style big view
inboxStyle.setBigContentTitle("Big Title Details:");

// Moves events into the big view
for (int i=0; i < events.length; i++) {
    inboxStyle.addLine(events[i]);
}

mBuilder.setStyle(inboxStyle);

/* Creates an explicit intent for an Activity in your app */
Intent resultIntent = new Intent(this, NotificationView.class);

TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
stackBuilder.addParentStack(NotificationView.class);

/* Adds the Intent that starts the Activity to the top of the stack */
stackBuilder.addNextIntent(resultIntent);
```

```

    PendingIntent resultPendingIntent
    =stackBuilder.getPendingIntent(0, PendingIntent.FLAG_UPDATE_CURRENT);

    mBuilder.setContentIntent(resultPendingIntent);

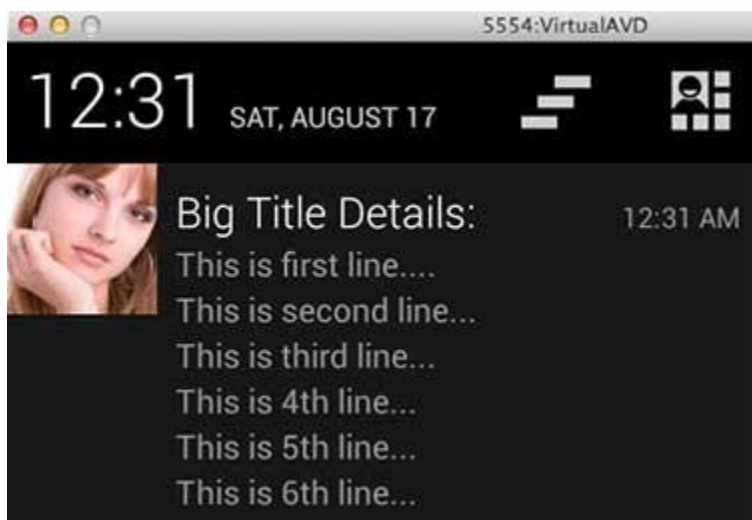
    mNotificationManager = (NotificationManager)
    getSystemService(Context.NOTIFICATION_SERVICE);

    /* notificationID allows you to update the notification later on. */

    mNotificationManager.notify(notificationID, mBuilder.build());
}

```

Sekarang jika Anda akan mencoba menjalankan aplikasi Anda maka Anda akan menemukan hasil berikut dalam bentuk tampilan yang diperluas -



Android - Layar Login

Aplikasi masuk adalah layar yang meminta kredensial Anda untuk masuk ke beberapa aplikasi tertentu. Anda mungkin telah melihatnya ketika masuk ke facebook, twitter dll

Bab ini menjelaskan, cara membuat layar masuk dan cara mengelola keamanan saat upaya palsu dilakukan.

Pertama Anda harus mendefinisikan dua TextView yang menanyakan nama pengguna dan kata sandi pengguna. Kata sandi TextView harus memiliki **inputType** yang diatur ke kata sandi. Sintaksnya diberikan di bawah ini -

```
<EditText

    android:id = "@+id/editText2"

    android:layout_width = "wrap_content"

    android:layout_height = "wrap_content"

    android:inputType = "textPassword" />

<EditText

    android:id = "@+id/editText1"

    android:layout_width = "wrap_content"

    android:layout_height = "wrap_content"

/>
```

Tentukan tombol dengan teks masuk dan setel pada **Klik** Properti. Setelah itu tentukan fungsi yang disebutkan di properti onClick dalam file java.

```
<Button

    android:id = "@+id/button1"

    android:layout_width = "wrap_content"

    android:layout_height = "wrap_content"

    android:onClick = "login"
```

```
android:text = "@string/Login"

/>
```

Dalam file java, di dalam metode onClick dapatkan teks nama pengguna dan kata sandi menggunakan **getText ()** dan **toString ()** metode dan cocokkan dengan teks menggunakan **equals ()** fungsi.

```
EditText username = (EditText)findViewById(R.id.editText1);
EditText password = (EditText)findViewById(R.id.editText2);

public void login(View view){

    if(username.getText().toString().equals("admin") &&
password.getText().toString().equals("admin")){

        //correct password

    }else{

        //wrong password

    }
}
```

Hal terakhir yang perlu Anda lakukan adalah menyediakan mekanisme keamanan, sehingga upaya yang tidak diinginkan harus dihindari. Untuk ini menginisialisasi variabel dan pada setiap upaya palsu, kurangi itu. Dan ketika mencapai 0, nonaktifkan tombol login.

```
int counter = 3;

counter--;

if(counter==0){

    //disble the button, close the application e.t.c

}
```

Contoh

Berikut ini contoh yang menunjukkan aplikasi masuk. Ini menciptakan aplikasi dasar yang memberi Anda hanya tiga upaya untuk masuk ke aplikasi. Untuk bereksperimen dengan contoh ini, Anda dapat menjalankan ini di perangkat yang sebenarnya atau di emulator.

Tangga	Deskripsi
1	Anda akan menggunakan studio Android untuk membuat aplikasi Android di bawah paket com.example.sairamkrishna.myapplication.
3	Ubah file src / MainActivity.java untuk menambahkan kode yang diperlukan.
4	Ubah res / layout / activity_main untuk menambahkan komponen XML masing-masing
5	Jalankan aplikasi dan pilih perangkat android yang berjalan dan instal aplikasi di atasnya dan verifikasi hasilnya

Berikut ini adalah isi dari file aktivitas utama yang dimodifikasi **src / MainActivity.java** .

```
package com.example.sairamkrishna.myapplication;

import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;

import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
```



```

public class MainActivity extends Activity {

    Button b1,b2;

    EditText ed1,ed2;

    TextView tx1;

    int counter = 3;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        b1 = (Button)findViewById(R.id.button);

        ed1 = (EditText)findViewById(R.id.editText);

        ed2 = (EditText)findViewById(R.id.editText2);

        b2 = (Button)findViewById(R.id.button2);

        tx1 = (TextView)findViewById(R.id.textView3);

        tx1.setVisibility(View.GONE);

        b1.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                if(ed1.getText().toString().equals("admin") &&

                    ed2.getText().toString().equals("admin")) {

                    Toast.makeText(getApplicationContext(),

```

```

        "Redirecting...", Toast.LENGTH_SHORT).show();
    }else{
        Toast.makeText(getApplicationContext(), "Wrong
        Credentials", Toast.LENGTH_SHORT).show();

        tx1.setVisibility(View.VISIBLE);
        tx1.setBackgroundColor(Color.RED);
        counter--;
        tx1.setText(Integer.toString(counter));

        if (counter == 0) {
            b1.setEnabled(false);
        }
    }
}

});

b2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}
}

```

Berikut ini adalah konten modifikasi dari **res / layout / activity_main.xml**xml .

Dalam kode berikut **abc** menunjukkan tentang logo tutorialspoint.com

```

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

<RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"

    xmlns:tools = "http://schemas.android.com/tools"
    android:layout_width="match_parent"

    android:layout_height = "match_parent" android:paddingLeft=
"@dimen/activity_horizontal_margin"

    android:paddingRight = "@dimen/activity_horizontal_margin"

    android:paddingTop = "@dimen/activity_vertical_margin"

    android:paddingBottom = "@dimen/activity_vertical_margin" tools:context =
".MainActivity">

    <TextView android:text = "Login" android:layout_width="wrap_content"

        android:layout_height = "wrap_content"

        android:id = "@+id/textview"

        android:textSize = "35dp"

        android:layout_alignParentTop = "true"

        android:layout_centerHorizontal = "true" />

    <TextView

        android:layout_width = "wrap_content"

        android:layout_height = "wrap_content"

        android:text = "Tutorials point"

        android:id = "@+id/textView"

        android:layout_below = "@+id/textview"

        android:layout_centerHorizontal = "true"

        android:textColor = "#ff7aff24"

        android:textSize = "35dp" />

    <EditText

```

```
android:layout_width = "wrap_content"
android:layout_height = "wrap_content"
android:id = "@+id/editText"
android:hint = "Enter Name"
android:focusable = "true"
android:textColorHighlight = "#ff7eff15"
android:textColorHint = "#ffff25e6"
android:layout_marginTop = "46dp"
android:layout_below = "@+id/imageView"
android:layout_alignParentLeft = "true"
android:layout_alignParentStart = "true"
android:layout_alignParentRight = "true"
android:layout_alignParentEnd = "true" />
```

<ImageView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageView"
android:src="@drawable/abc"
android:layout_below="@+id/textView"
android:layout_centerHorizontal="true" />
```

<EditText

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textPassword"
android:ems="10"
android:id="@+id/editText2"
```

```
android:layout_below="@+id/editText"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignRight="@+id/editText"
android:layout_alignEnd="@+id/editText"
android:textColorHint="#ffff299f"
android:hint="Password" />
```

<TextView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Attempts Left:"
android:id="@+id/textView2"
android:layout_below="@+id/editText2"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:textSize="25dp" />
```

<TextView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="New Text"
android:id="@+id/textView3"
android:layout_alignTop="@+id/textView2"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:layout_alignBottom="@+id/textView2"
android:layout_toEndOf="@+id/textview"
```

```
android:textSize="25dp"

android:layout_toRightOf="@+id/textview" />
```

```
<Button
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="login"
    android:id="@+id/button"
    android:layout_alignParentBottom="true"
    android:layout_toLeftOf="@+id/textview"
    android:layout_toStartOf="@+id/textview" />
```

```
<Button
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Cancel"
    android:id="@+id/button2"
    android:layout_alignParentBottom="true"
    android:layout_toRightOf="@+id/textview"
    android:layout_toEndOf="@+id/textview" />
```

```
</RelativeLayout>
```

Berikut ini adalah isi dari **res / values / string.xml** .

```
<resources>

    <string name="app_name">My Application</string>

</resources>
```

Berikut ini adalah isi file **AndroidManifest.xml** .

```

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.sairamkrishna.myapplication" >


    <application

        android:allowBackup="true"

        android:icon="@mipmap/ic_launcher"

        android:label="@string/app_name"

        android:theme="@style/AppTheme" >


        <activity

            android:name=".MainActivity"

            android:label="@string/app_name" >


            <intent-filter>

                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />

            </intent-filter>


        </activity>

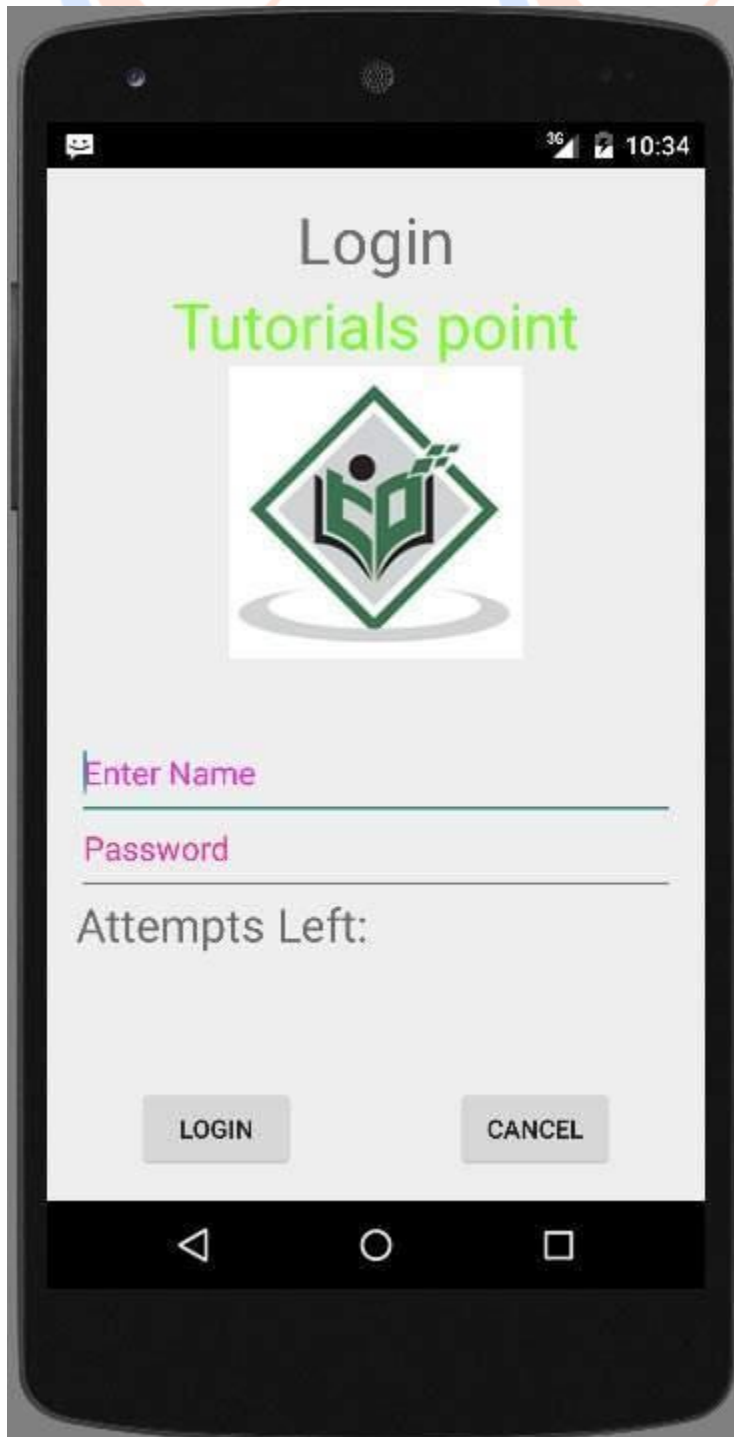

    </application>

</manifest>

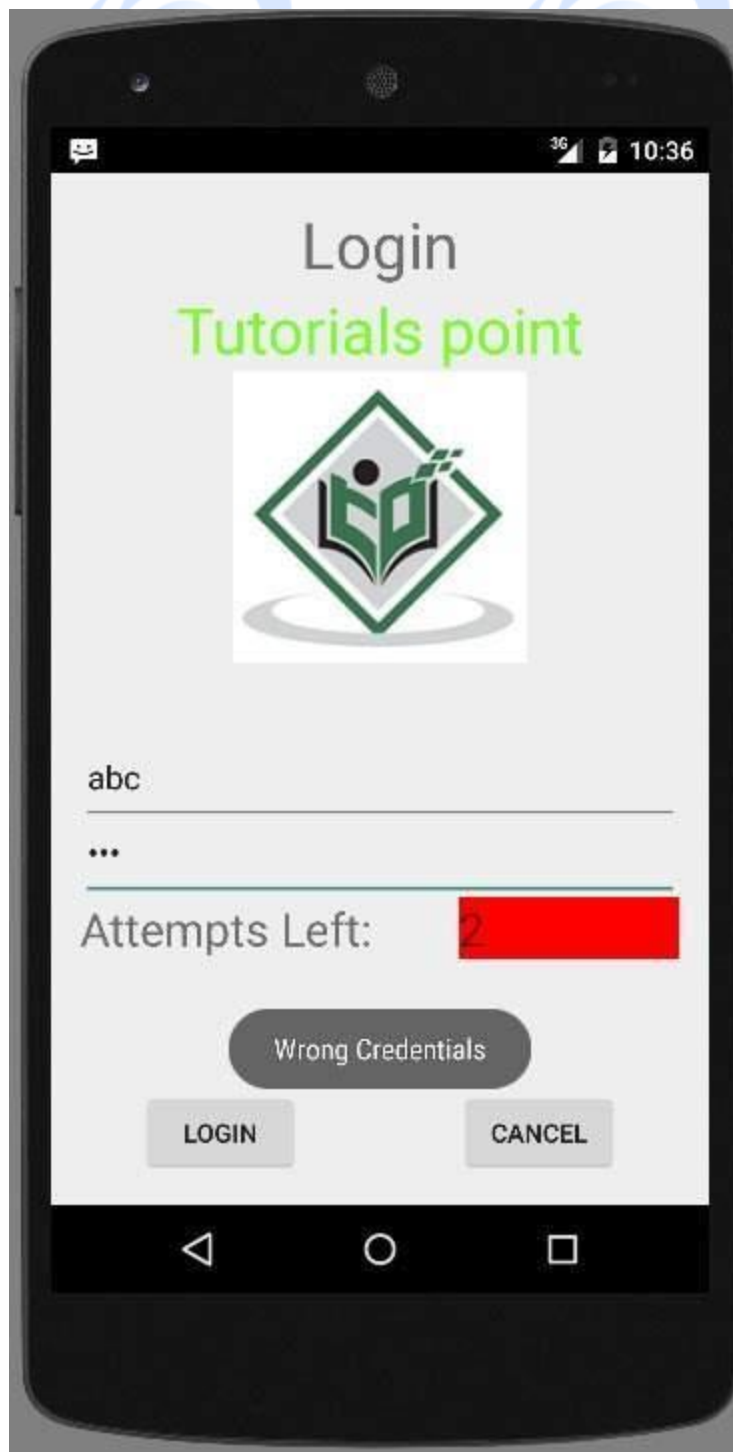
```

Mari coba jalankan aplikasi kita yang baru kita modifikasi. Saya berasumsi Anda telah membuat **AVD** Anda saat melakukan pengaturan lingkungan. Untuk menjalankan aplikasi dari studio Android, buka salah satu file aktivitas proyek Anda dan klik Jalankan 🏃 ikon dari bilah alat. Android studio menginstal aplikasi pada AVD Anda dan memulainya dan jika

semuanya baik-baik saja dengan pengaturan dan aplikasi Anda, itu akan menampilkan jendela Emulator berikut -

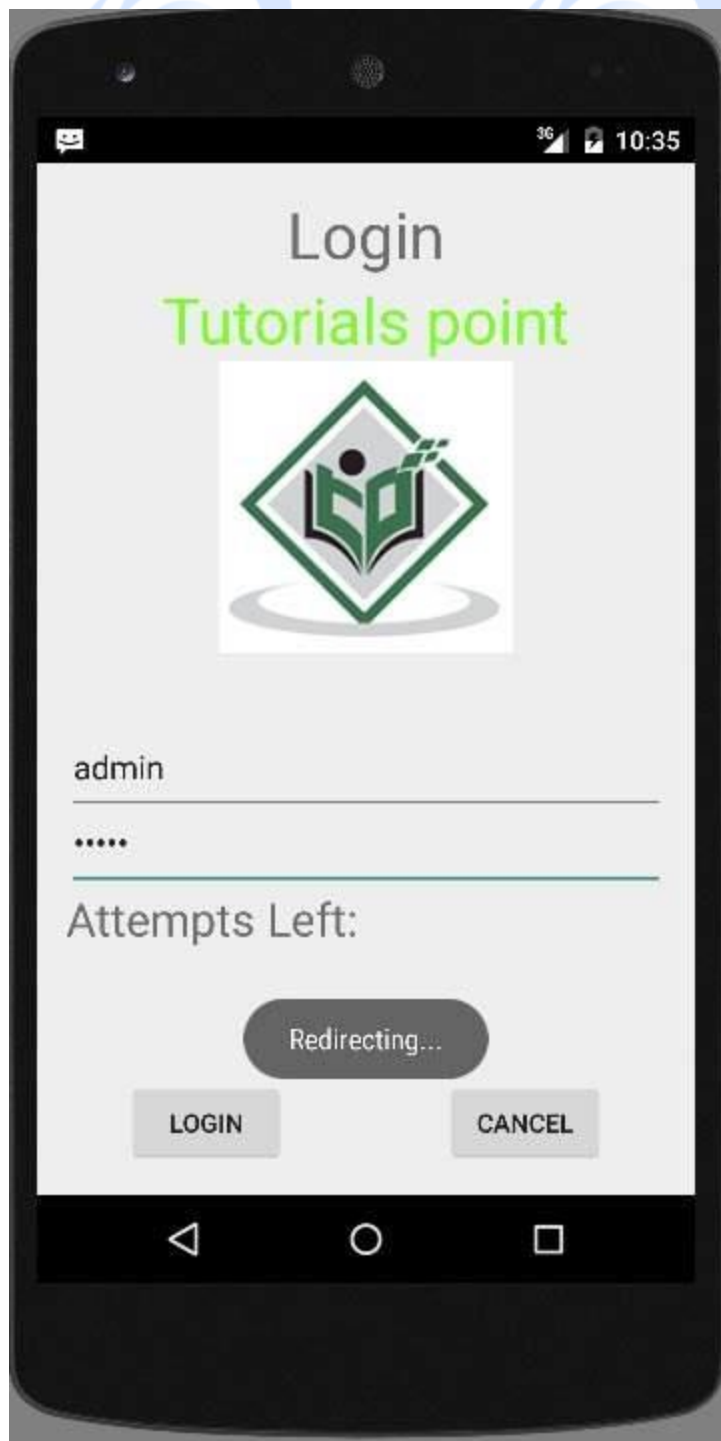


Ketik apa pun di bidang nama pengguna dan kata sandi, lalu tekan tombol login. Saya menempatkan abc di bidang nama pengguna dan abc di bidang kata sandi. Saya mendapat upaya gagal. Ini ditunjukkan di bawah -



Lakukan ini dua kali lagi, dan Anda akan melihat bahwa Anda memiliki 0 upaya login yang tersisa dan tombol login Anda dinonaktifkan.

Sekarang buka aplikasi lagi, dan kali ini masukkan username yang benar sebagai admin dan kata sandi sebagai admin dan klik pada login. Anda akan berhasil masuk.



Jika pengguna menekan tombol batal, maka akan menutup aplikasi layar login.

Tutorial Membuat Aplikasi Android Login dan Register Database MySQL dilengkapi dengan Session menggunakan SharedPreferences

1. Pertama membuat database dengan nama kuncoro_login dengan nama tabel users dan struktur tabelnya seperti berikut :

Column name	Data Type	length	Primary Key	Not Null	Auto Increment
id	int	5		v	v
username	varchar	30	v	v	
password	varchar	30			

2. Membuat web service untuk parsing data aplikasi android.
koneksi.php
Sebagai koneksi aplikasi ke database. Coding-nya [disini](#).
login.php
Untuk akses user terdaftar agar bisa login app. Coding-nya [disini](#).
register.php
Untuk menambah user baru. Coding-nya [disini](#).
3. Buat project baru di Android Studio File ⇒ New Project. Kemudian pilih Blank Activity dan melanjutkannya hingga selesai.

✎ activity_main.xml
Tampilan untuk Main setelah login.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context="com.dedykuncoro.login.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/logo"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:background="@mipmap/ic_launcher" />

        <TextView
            android:id="@+id/TextView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginBottom="5dp"
            android:layout_marginTop="5dp"
            android:text="www.dedykuncoro.com"
            android:textSize="18dip"
            android:textStyle="bold" />

    </LinearLayout>

</RelativeLayout>
```

```

<View
    android:id="@+id/View1"
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:layout_gravity="center"
    android:background="#448AFF" />

<TextView
    android:id="@+id/txt_id"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:text="ID"/>

<TextView
    android:id="@+id/txt_username"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:text="Username"/>

<Button
    android:id="@+id/btn_logout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="50dp"
    android:background="#00555555"
    android:text="Logout"
    android:textStyle="bold" />

</LinearLayout>

</RelativeLayout>

```

login.xml
Sebagai tampilan dari login.

```

<?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:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/logo"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:background="@mipmap/ic_launcher" />

```

```

<TextView
    android:id="@+id/TextView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="5dp"
    android:layout_marginTop="5dp"
    android:text="www.dedykuncoro.com"
    android:textSize="18dip"
    android:textStyle="bold" />

<View
    android:id="@+id/View1"
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:layout_gravity="center"
    android:background="#448AFF" />

<TextView
    android:id="@+id/TextView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="30dip"
    android:text="WELCOME, LOGIN BELOW"
    android:textSize="16dip" />

<EditText
    android:id="@+id/txt_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:hint="Username"
    android:inputType="textEmailAddress"
    android:singleLine="true" />

<EditText
    android:id="@+id/txt_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:hint="Password"
    android:password="true"
    android:singleLine="true" />

<Button
    android:id="@+id/btn_login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="15dp"
    android:text="Login"
    android:background="#448AFF"
    android:textColor="#fff"
    android:textStyle="bold" />

```

```

        <Button
            android:id="@+id/btn_register"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginBottom="15dp"
            android:background="#00555555"
            android:text="Register"
            android:textStyle="bold" />

    </LinearLayout>

</RelativeLayout>

```

✂ register.xml
Sebagai tampilan dari register.

```

<?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:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/logo"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:background="@mipmap/ic_launcher" />

        <TextView
            android:id="@+id/TextView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginBottom="5dp"
            android:layout_marginTop="5dp"
            android:text="www.dedykuncoro.com"
            android:textSize="18dip"
            android:textStyle="bold" />

        <View
            android:id="@+id/View1"
            android:layout_width="fill_parent"
            android:layout_height="1dp"
            android:layout_gravity="center"
            android:background="#448AFF" />
    </LinearLayout>
</RelativeLayout>

```



```
<TextView
    android:id="@+id/TextView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="30dip"
    android:text="REGISTER BELOW"
    android:textSize="16dip" />

<EditText
    android:id="@+id/txt_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:hint="Username"
    android:inputType="textEmailAddress"
    android:singleLine="true" />

<EditText
    android:id="@+id/txt_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:hint="Password"
    android:password="true"
    android:singleLine="true" />

<EditText
    android:id="@+id/txt_confirm_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:hint="Confirmation Password"
    android:password="true" />

<Button
    android:id="@+id/btn_register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="15dp"
    android:background="#448AFF"
    android:textColor="#fff"
    android:text="Register"
    android:textStyle="bold" />

<Button
    android:id="@+id/btn_login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="15dp"
    android:background="#00555555"
    android:text="Login"
    android:textStyle="bold" />
```

```
</LinearLayout>
```

```
</RelativeLayout>
```

✎ Buka build.gradle dan tambahkan volley library didalamnya.

```
compile 'com.android.volley:volley:1.0.0'
```

✎ build.gradle

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    androidTestCompile('com.android.support.test.espresso:espresso-  
core:2.2.2', {  
        exclude group: 'com.android.support', module: 'support-  
annotations'  
    })  
    compile 'com.android.support:appcompat-v7:25.3.0'  
    compile 'com.android.support.constraint:constraint-layout:1.0.2'  
    compile 'com.android.volley:volley:1.0.0' /*tambahan volley library*/  
    testCompile 'junit:junit:4.12'  
}
```

Agar project terstruktur dan terorganisir, buat package **app**. Untuk membuat paket baru, klik kanan pada **src=>New=>Package** dan memberikan nama paket. Contoh : **com.dedykuncoro.login**.

Buat class **AppController.java** didalam package **app** dan tambah coding seperti dibawah ini. Class tunggal yang menginisialisasi class global yang diperlukan. Semua objek yang berhubungan dengan [volley](#) diinisialisasi di sini.

✎ **AppController.java**

```
package com.dedykuncoro.login.app;  
  
import android.app.Application;  
import android.text.TextUtils;  
  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.toolbox.ImageLoader;  
import com.android.volley.toolbox.Volley;  
  
public class AppController extends Application {  
  
    public static final String TAG =  
AppController.class.getSimpleName();  
  
    private RequestQueue mRequestQueue;  
  
    private static AppController mInstance;  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        mInstance = this;  
    }  
}
```

```

    public static synchronized ApplicationController getInstance() {
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue =
                Volley.newRequestQueue(getApplicationContext());
        }

        return mRequestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req, String tag) {
        req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
        getRequestQueue().add(req);
    }

    public <T> void addToRequestQueue(Request<T> req) {
        req.setTag(TAG);
        getRequestQueue().add(req);
    }

    public void cancelPendingRequests(Object tag) {
        if (mRequestQueue != null) {
            mRequestQueue.cancelAll(tag);
        }
    }
}

```

Buat class **Server.java** didalam package **util** dan tambahkan coding seperti dibawah ini. Class ini berfungsi mengatur link server sumber data dari web service yang akan digunakan.

 **Server.java**

```


package com.dedykuncoro.login;

/**
 * Created by KUNCORO on 24/03/2017.
 */

public class Server {
    public static final String URL =
        "http://10.0.2.2/android/kuncoro_login/";
}

```

Buat class **Login.java** dan tambahkan coding seperti dibawah ini. Class ini berfungsi untuk login.

 **Login.java**

```

package com.dedykuncoro.login;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.ConnectivityManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.dedykuncoro.login.app.AppController;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Kuncoro on 03/24/2017.
 */
public class Login extends AppCompatActivity {

    ProgressDialog pDialog;
    Button btn_register, btn_login;
    EditText txt_username, txt_password;
    Intent intent;

    int success;
    ConnectivityManager conMgr;

    private String url = Server.URL + "login.php";

    private static final String TAG = Login.class.getSimpleName();

    private static final String TAG_SUCCESS = "success";
    private static final String TAG_MESSAGE = "message";

    public final static String TAG_USERNAME = "username";
    public final static String TAG_ID = "id";

    String tag_json_obj = "json_obj_req";

    SharedPreferences sharedPreferences;
    Boolean session = false;
    String id, username;
    public static final String my_shared_preferences =
"my_shared_preferences";
    public static final String session_status = "session_status";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login);

        conMgr = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    }

```

```

        if (conMgr.getActiveNetworkInfo() != null
            && conMgr.getActiveNetworkInfo().isAvailable()
            && conMgr.getActiveNetworkInfo().isConnected()) {
        } else {
            Toast.makeText(getApplicationContext(), "No Internet
Connection",
                        Toast.LENGTH_LONG).show();
        }
    }

    btn_login = (Button) findViewById(R.id.btn_login);
    btn_register = (Button) findViewById(R.id.btn_register);
    txt_username = (EditText) findViewById(R.id.txt_username);
    txt_password = (EditText) findViewById(R.id.txt_password);

    // Cek session login jika TRUE maka langsung buka MainActivity
    sharedPreferences = getSharedPreferences(my_shared_preferences,
Context.MODE_PRIVATE);
    session = sharedPreferences.getBoolean(session_status, false);
    id = sharedPreferences.getString(TAG_ID, null);
    username = sharedPreferences.getString(TAG_USERNAME, null);

    if (session) {
        Intent intent = new Intent(Login.this, MainActivity.class);
        intent.putExtra(TAG_ID, id);
        intent.putExtra(TAG_USERNAME, username);
        finish();
        startActivity(intent);
    }

    btn_login.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            String username = txt_username.getText().toString();
            String password = txt_password.getText().toString();

            // mengecek kolom yang kosong
            if (username.trim().length() > 0 &&
password.trim().length() > 0) {
                if (conMgr.getActiveNetworkInfo() != null
                    &&
conMgr.getActiveNetworkInfo().isAvailable()
                    &&
conMgr.getActiveNetworkInfo().isConnected()) {
                    checkLogin(username, password);
                } else {
                    Toast.makeText(getApplicationContext() ,"No
Internet Connection", Toast.LENGTH_LONG).show();
                }
            } else {
                // Prompt user to enter credentials
                Toast.makeText(getApplicationContext() ,"Kolom tidak
boleh kosong", Toast.LENGTH_LONG).show();
            }
        }
    });

```

```

        }
    });

    btn_register.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            intent = new Intent(Login.this, Register.class);
            finish();
            startActivity(intent);
        }
    });

}

private void checkLogin(final String username, final String password)
{
    pDialog = new ProgressDialog(this);
    pDialog.setCancelable(false);
    pDialog.setMessage("Logging in ...");
    showDialog();

    StringRequest strReq = new StringRequest(Request.Method.POST,
url, new Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            Log.e(TAG, "Login Response: " + response.toString());
            hideDialog();

            try {
                JSONObject jsonObj = new JSONObject(response);
                success = jsonObj.getInt(TAG_SUCCESS);

                // Check for error node in json
                if (success == 1) {
                    String username = jsonObj.getString(TAG_USERNAME);
                    String id = jsonObj.getString(TAG_ID);

                    Log.e("Successfully Login!", jsonObj.toString());

                    Toast.makeText(getApplicationContext(),
jObj.getString(TAG_MESSAGE), Toast.LENGTH_LONG).show();

                    // menyimpan login ke session
                    SharedPreferences.Editor editor =
sharedpreferences.edit();
                    editor.putBoolean(session_status, true);
                    editor.putString(TAG_ID, id);
                    editor.putString(TAG_USERNAME, username);
                    editor.commit();

                    // Mengambil main activity
                    Intent intent = new Intent(Login.this,
MainActivity.class);
                    intent.putExtra(TAG_ID, id);

```

```

        intent.putExtra(TAG_USERNAME, username);
        finish();
        startActivity(intent);
    } else {
        Toast.makeText(getApplicationContext(),
            jsonObj.getString(TAG_MESSAGE),
            Toast.LENGTH_LONG).show();
    }
} catch (JSONException e) {
    // JSON error
    e.printStackTrace();
}

}, new Response.ErrorListener() {

    @Override
    public void onErrorResponse(VolleyError error) {
        Log.e(TAG, "Login Error: " + error.getMessage());
        Toast.makeText(getApplicationContext(),
            error.getMessage(), Toast.LENGTH_LONG).show();

        hideDialog();
    }
}) {

    @Override
    protected Map<String, String> getParams() {
        // Posting parameters to login url
        Map<String, String> params = new HashMap<String,
String>();

        params.put("username", username);
        params.put("password", password);

        return params;
    }

};

// Adding request to request queue
AppController.getInstance().addToRequestQueue(strReq,
tag_json_obj);
}

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
}

private void hideDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}
}

```

Buat class Register.java dan tambahkan coding seperti dibawah ini. Class ini berfungsi menambah user baru.

Register.java

```
package com.dedykuncoro.login;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.dedykuncoro.login.app.AppController;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class Register extends AppCompatActivity {

    ProgressDialog pDialog;
    Button btn_register, btn_login;
    EditText txt_username, txt_password, txt_confirm_password;
    Intent intent;

    int success;
    ConnectivityManager conMgr;

    private String url = Server.URL + "register.php";

    private static final String TAG = Register.class.getSimpleName();

    private static final String TAG_SUCCESS = "success";
    private static final String TAG_MESSAGE = "message";

    String tag_json_obj = "json_obj_req";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.register);

        conMgr = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
        {
            if (conMgr.getActiveNetworkInfo() != null
                && conMgr.getActiveNetworkInfo().isAvailable()
                && conMgr.getActiveNetworkInfo().isConnected()) {
            } else {
                Toast.makeText(getApplicationContext(), "No Internet
Connection",
```

```

        Toast.LENGTH_LONG).show();
    }
}

btn_login = (Button) findViewById(R.id.btn_login);
btn_register = (Button) findViewById(R.id.btn_register);
txt_username = (EditText) findViewById(R.id.txt_username);
txt_password = (EditText) findViewById(R.id.txt_password);
txt_confirm_password = (EditText)
findViewById(R.id.txt_confirm_password);

btn_login.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        intent = new Intent(Register.this, Login.class);
        finish();
        startActivity(intent);
    }
});

btn_register.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        String username = txt_username.getText().toString();
        String password = txt_password.getText().toString();
        String confirm_password =
txt_confirm_password.getText().toString();

        if (conMgr.getActiveNetworkInfo() != null
            && conMgr.getActiveNetworkInfo().isAvailable()
            && conMgr.getActiveNetworkInfo().isConnected()) {
            checkRegister(username, password, confirm_password);
        } else {
            Toast.makeText(getApplicationContext(), "No Internet
Connection", Toast.LENGTH_SHORT).show();
        }
    }
});

}

private void checkRegister(final String username, final String
password, final String confirm_password) {
    pDialog = new ProgressDialog(this);
    pDialog.setCancelable(false);
    pDialog.setMessage("Register ...");
    showDialog();

    StringRequest strReq = new StringRequest(Request.Method.POST,
url, new Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            Log.e(TAG, "Register Response: " + response.toString());
            hideDialog();

```

```

        try {
            JSONObject jsonObj = new JSONObject(response);
            success = jsonObj.getInt(TAG_SUCCESS);

            // Check for error node in json
            if (success == 1) {

                Log.e("Successfully Register!", jsonObj.toString());

                Toast.makeText(getApplicationContext(),
                    jsonObj.getString(TAG_MESSAGE),
Toast.LENGTH_LONG).show();

                txt_username.setText("");
                txt_password.setText("");
                txt_confirm_password.setText("");

            } else {
                Toast.makeText(getApplicationContext(),
                    jsonObj.getString(TAG_MESSAGE),
Toast.LENGTH_LONG).show();

            }
        } catch (JSONException e) {
            // JSON error
            e.printStackTrace();
        }
    }
}, new Response.ErrorListener() {

    @Override
    public void onErrorResponse(VolleyError error) {
        Log.e(TAG, "Login Error: " + error.getMessage());
        Toast.makeText(getApplicationContext(),
            error.getMessage(), Toast.LENGTH_LONG).show();

        hideDialog();
    }
}) {

    @Override
    protected Map<String, String> getParams() {
        // Posting parameters to login url
        Map<String, String> params = new HashMap<String,
String>();

        params.put("username", username);
        params.put("password", password);
        params.put("confirm_password", confirm_password);

        return params;
    }

};

// Adding request to request queue
AppController.getInstance().addToRequestQueue(strReq,
tag_json_objj);

```

```

    }

    private void showDialog() {
        if (!pDialog.isShowing())
            pDialog.show();
    }

    private void hideDialog() {
        if (pDialog.isShowing())
            pDialog.dismiss();
    }

    @Override
    public void onBackPressed() {
        intent = new Intent(Register.this, Login.class);
        finish();
        startActivity(intent);
    }
}

```

Buka class MainActivity.java dan tambahkan coding seperti dibawah ini. Class ini berfungsi menampilkan data user yang berhasil login.

✍ MainActivity.java

```

package com.dedykuncoro.login;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    Button btn_logout;
    TextView txt_id, txt_username;
    String id, username;
    SharedPreferences sharedPreferences;

    public static final String TAG_ID = "id";
    public static final String TAG_USERNAME = "username";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        txt_id = (TextView) findViewById(R.id.txt_id);
        txt_username = (TextView) findViewById(R.id.txt_username);
        btn_logout = (Button) findViewById(R.id.btn_logout);

        sharedPreferences =
        getSharedPreferences(Login.my_shared_preferences, Context.MODE_PRIVATE);

        id = getIntent().getStringExtra(TAG_ID);
        username = getIntent().getStringExtra(TAG_USERNAME);
    }
}

```

```

txt_id.setText("ID : " + id);
txt_username.setText("USERNAME : " + username);

btn_logout.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        // update login session ke FALSE dan mengosongkan nilai
id dan username
        SharedPreferences.Editor editor =
sharedpreferences.edit();
        editor.putBoolean(Login.session_status, false);
        editor.putString(TAG_ID, null);
        editor.putString(TAG_USERNAME, null);
        editor.commit();

        Intent intent = new Intent(MainActivity.this,
Login.class);
        finish();
        startActivity(intent);
    }
});
}
}

```

Tambahkan perijinan INTERNET, ACCESS_NETWORK_STATE, android:name=".app.AppController" pada application.

✎ AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.dedykuncoro.login">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:name=".app.AppController"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".Login" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>

            </intent-filter>
        </activity>
        <activity android:name=".MainActivity" />
        <activity android:name=".Register" />
    </application>
</manifest>

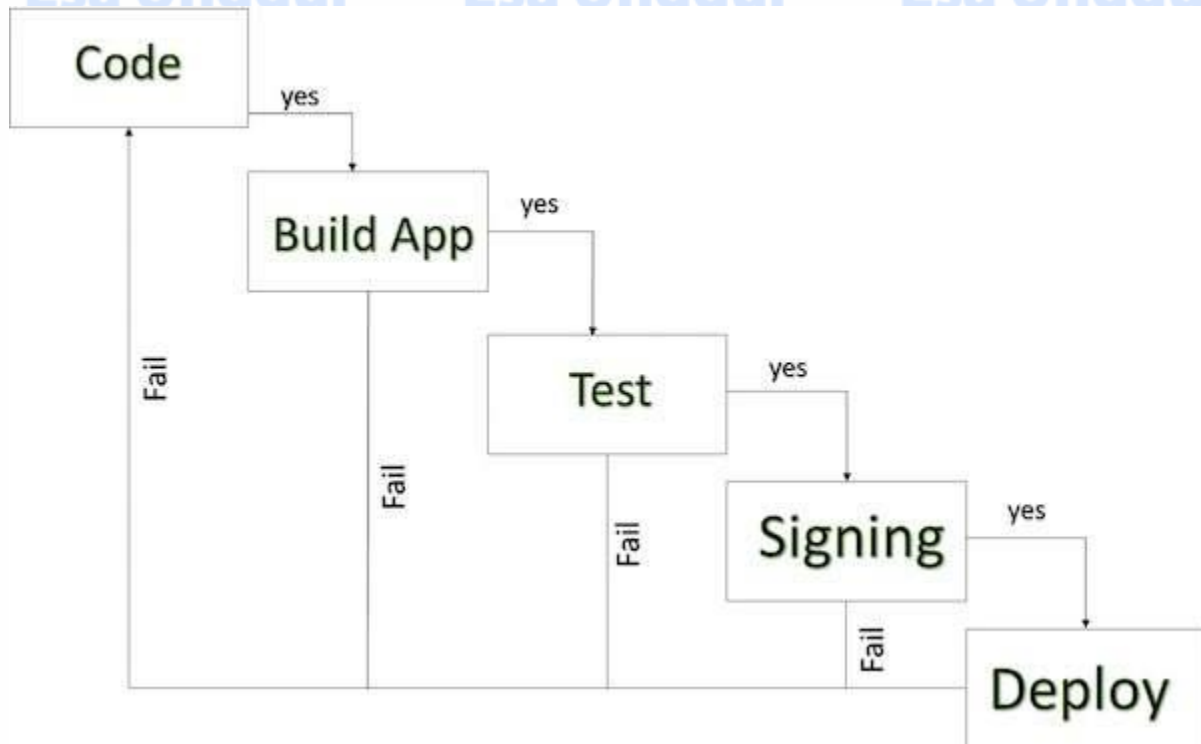
```

selengkapnya : <https://github.com/dedykuncoro/Android-Login-Register-MySQL>



Mempublikasikan Aplikasi Android

Penerbitan aplikasi Android adalah proses yang membuat aplikasi Android Anda tersedia bagi pengguna. Menginfeksi, penerbitan adalah fase terakhir dari proses pengembangan aplikasi Android.



SIKLUS HIDUP PENGEMBANGAN ANDROID

Setelah Anda mengembangkan dan sepenuhnya menguji Aplikasi Android Anda, Anda dapat mulai menjual atau mendistribusikan gratis menggunakan Google Play (A Android Marketplace yang terkenal). Anda juga dapat merilis aplikasi Anda dengan mengirimkannya langsung ke pengguna atau dengan membiarkan pengguna mengunduhnya dari situs web Anda sendiri.

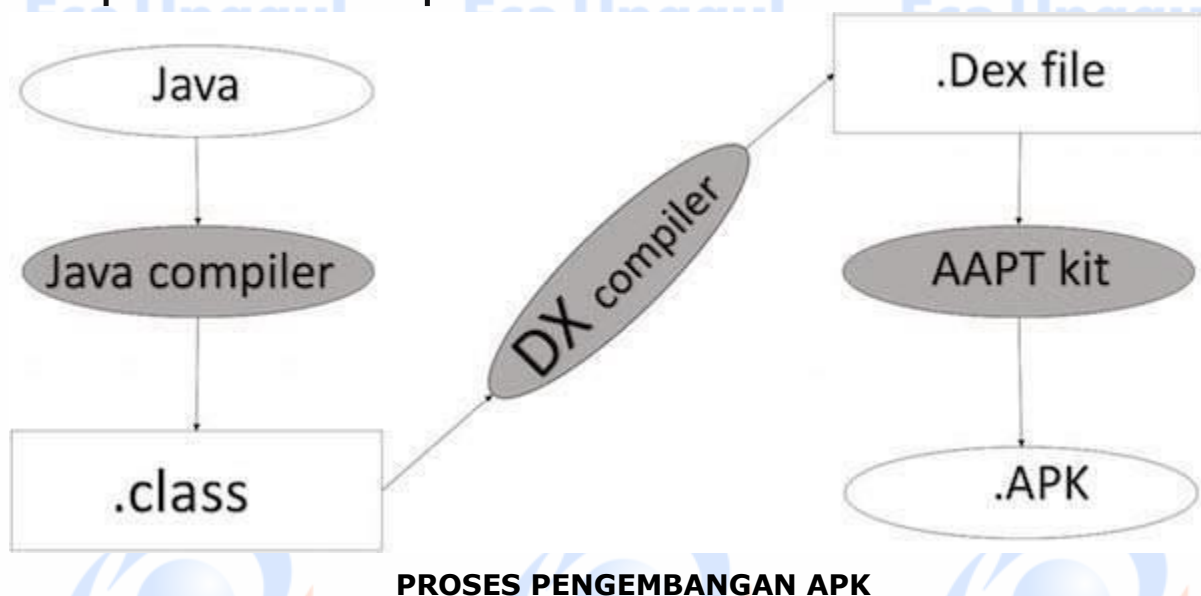
Anda dapat memeriksa proses penerbitan detail di situs web resmi Android, tetapi tutorial ini akan membawa Anda melalui langkah-langkah sederhana untuk meluncurkan aplikasi Anda di Google Play. Berikut ini daftar periksa yang disederhanakan yang akan membantu Anda dalam meluncurkan aplikasi Android Anda -

Langkah	Aktivitas
---------	-----------

1	Pengujian Regresi Sebelum mempublikasikan aplikasi Anda, Anda perlu memastikan bahwa pertemuannya memenuhi harapan kualitas dasar untuk semua aplikasi Android, di semua perangkat yang Anda targetkan. Jadi lakukan semua pengujian yang diperlukan pada perangkat yang berbeda termasuk ponsel dan tablet.
2	Peringkat Aplikasi Saat Anda akan mempublikasikan aplikasi Anda di Google Play, Anda harus menentukan peringkat konten untuk aplikasi Anda, yang memberi tahu pengguna Google tentang tingkat kedewasaannya. Peringkat saat ini tersedia adalah (a) Semua Orang (b) Kedewasaan rendah (c) Kematangan sedang (d) Kedewasaan tinggi.
3	Kawasan Tertarget Google Play memungkinkan Anda mengontrol negara dan wilayah mana aplikasi Anda akan dijual. Dengan demikian Anda harus mengatur zona waktu, lokalisasi atau persyaratan khusus lainnya sesuai dengan wilayah yang ditargetkan.
4	Ukuran Aplikasi Saat ini, ukuran maksimum untuk APK yang dipublikasikan di Google Play adalah 50 MB. Jika aplikasi Anda melebihi ukuran tersebut, atau jika Anda ingin menawarkan unduhan sekunder, Anda dapat menggunakan File Perluasan APK, yang akan dihosting Google Play secara gratis di infrastruktur servernya dan secara otomatis menangani unduhan ke perangkat.
5	SDK dan Kompatibilitas Layar Penting untuk memastikan bahwa aplikasi Anda dirancang untuk berjalan dengan baik pada versi platform Android dan ukuran layar perangkat yang ingin Anda targetkan.
6	Harga Aplikasi Menentukan apakah aplikasi Anda akan gratis atau berbayar penting karena, di Google Play, aplikasi gratis harus tetap gratis. Jika Anda ingin menjual aplikasi Anda maka Anda harus menentukan harganya dalam mata uang yang berbeda.
7	Konten Promosi Ini adalah praktik pemasaran yang baik untuk menyediakan berbagai aset grafis berkualitas tinggi untuk memamerkan aplikasi atau merek Anda. Setelah Anda menerbitkan, ini muncul di halaman detail produk Anda, di cantuman toko dan hasil pencarian, dan di tempat lain.

8	Bangun dan Unggah APK siap rilis APK siap rilis adalah apa yang akan Anda unggah ke Konsol Pengembang dan distribusikan kepada pengguna. Anda dapat memeriksa detail lengkap tentang cara membuat versi siap rilis dari aplikasi Anda: <u>Mempersiapkan Rilis</u> .
9	Finalisasi Detail Aplikasi Google Play memberi Anda berbagai cara untuk mempromosikan aplikasi Anda dan terlibat dengan pengguna di halaman detail produk Anda, mulai dari grafik warna-warni, tangkapan layar, dan video hingga deskripsi yang dilokalkan, perincian rilis, dan tautan ke aplikasi Anda yang lain. Jadi Anda dapat menghias halaman aplikasi Anda dan memberikan detail yang jelas dan jelas yang dapat Anda berikan.

Ekspor Proses Aplikasi Android

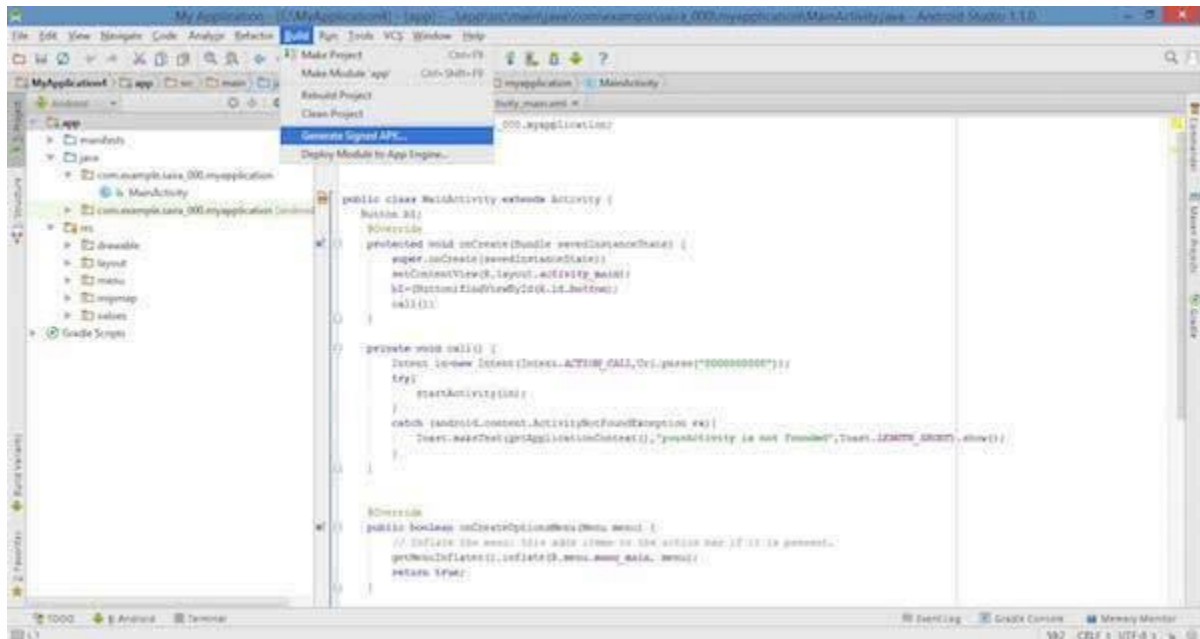


Sebelum mengekspor aplikasi, Anda harus memiliki beberapa alat

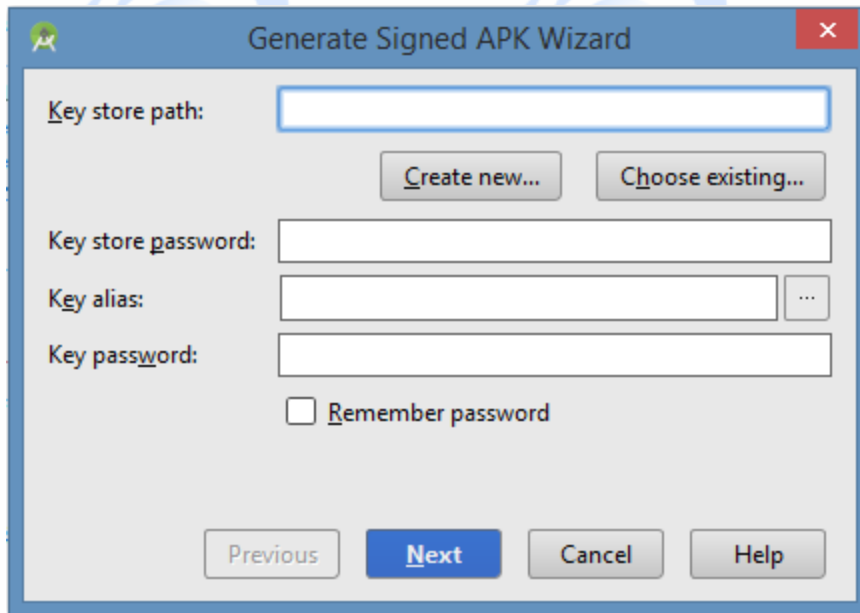
- **Dx tools** (Dalvik executable tools): Ini akan mengkonversi **file .class** ke **file .dex**. ini berguna untuk pengoptimalan memori dan mengurangi waktu kecepatan boot-up
- **AAPT** (alat pengemasan bantuan Android): berguna untuk mengonversi **file .Dex** menjadi **.Apk**
- **APK** (kit kemasan Android): Tahap akhir proses penyebaran disebut sebagai .apk.

Anda perlu mengekspor aplikasi Anda sebagai file APK (Paket Android) sebelum mengunggahnya ke Google Play marketplace.

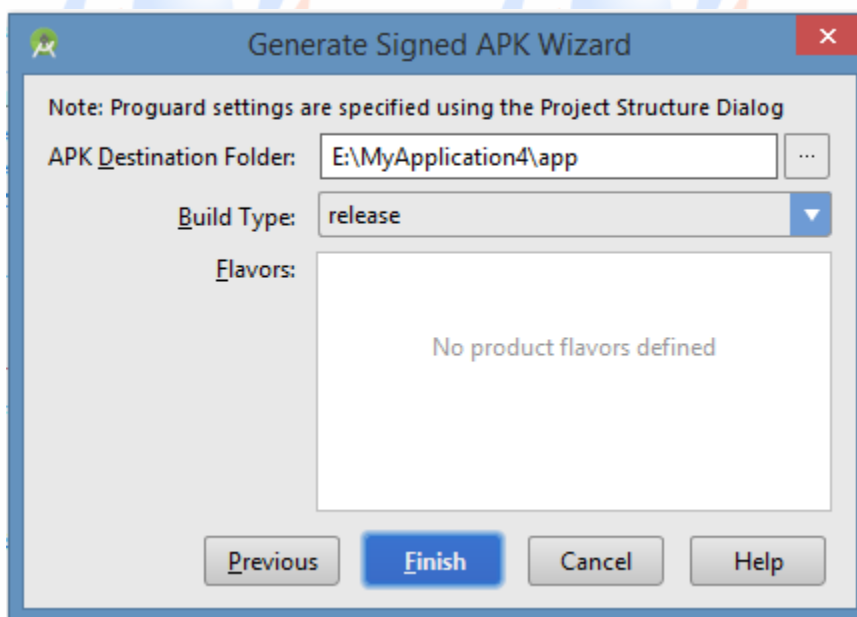
Untuk mengekspor aplikasi, buka saja proyek aplikasi itu di studio Android dan pilih **Build** → **Generate Signed APK** dari studio Android Anda dan ikuti langkah-langkah sederhana untuk mengekspor aplikasi Anda -



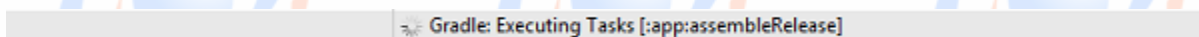
Selanjutnya, pilih opsi **Generate Signed APK** seperti yang diperlihatkan dalam cuplikan layar di atas dan kemudian klik sehingga Anda mendapatkan layar berikut di mana Anda akan memilih **Buat keystore baru** untuk menyimpan aplikasi Anda.



Masukkan jalur penyimpanan kunci Anda, kata sandi penyimpanan kunci, alias kunci dan kata sandi kunci untuk melindungi aplikasi Anda dan klik tombol **Berikutnya** sekali lagi. Ini akan menampilkan layar berikut untuk membiarkan Anda membuat aplikasi -



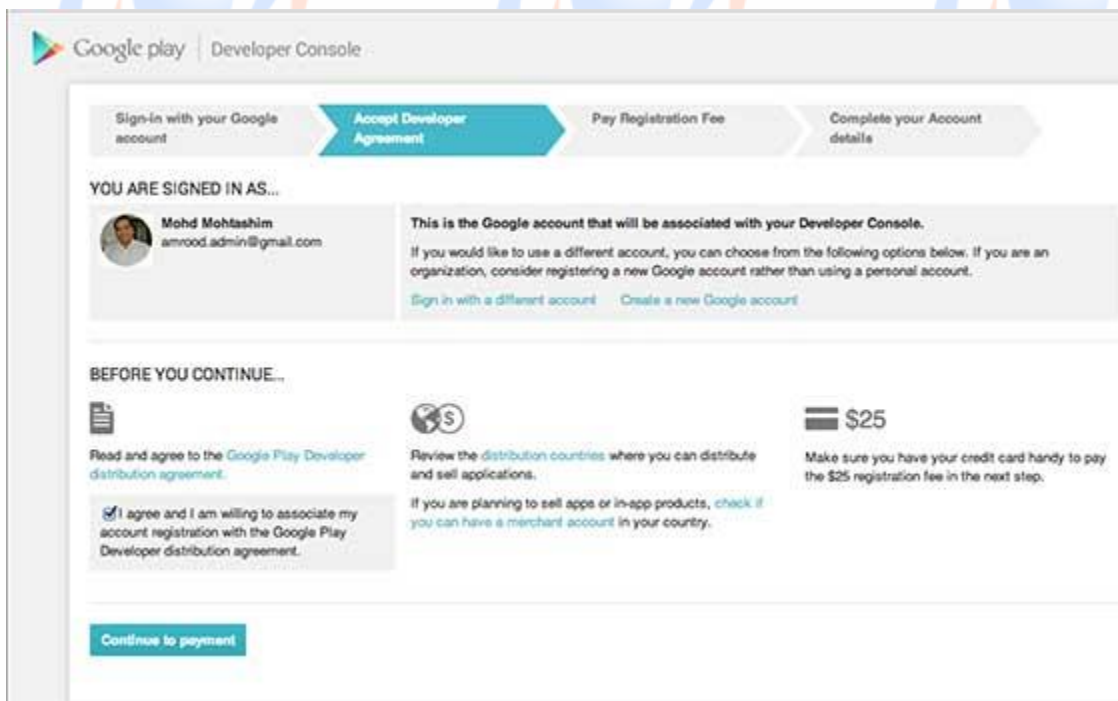
Setelah Anda mengisi semua informasi, seperti tujuan aplikasi, jenis bangunan, dan rasa, klik tombol **selesai**. Saat membuat aplikasi, aplikasi akan ditampilkan seperti di bawah ini



Akhirnya, ini akan menghasilkan Aplikasi Android Anda sebagai APK format File yang akan diunggah di Google Play marketplace.

Pendaftaran Google Play

Langkah yang paling penting adalah mendaftar dengan Google Play menggunakan Google Play Marketplace. Anda dapat menggunakan ID Google Anda yang ada jika Anda memiliki yang lain, Anda dapat membuat ID Google baru dan kemudian mendaftar ke pasar. Anda akan memiliki layar berikut untuk menerima syarat dan ketentuan.



Anda dapat menggunakan tombol **Lanjutkan ke pembayaran** untuk melanjutkan melakukan pembayaran sebesar \$ 25 sebagai biaya pendaftaran dan akhirnya melengkapi detail akun Anda.

Setelah Anda menjadi pengguna terdaftar di Google Play, Anda dapat mengunggah **APK siap rilis** untuk aplikasi Anda dan akhirnya Anda akan melengkapi detail aplikasi menggunakan halaman detail aplikasi sebagaimana disebutkan pada langkah 9 dari daftar periksa yang disebutkan di atas.

Menandatangani Aplikasi Anda Secara Manual

Anda tidak perlu Android Studio untuk menandatangani aplikasi Anda. Anda dapat menandatangani aplikasi Anda dari baris perintah menggunakan alat standar dari SDK Android dan JDK. Untuk menandatangani aplikasi dalam mode rilis dari baris perintah -

- Hasilkan kunci pribadi menggunakan keytool

```
$ keytool -genkey -v -keystore my-release-key.keystore  
-alias alias_name -keyalg RSA -keysize 2048 -validity 10000
```

- Kompilasi aplikasi Anda dalam mode rilis untuk mendapatkan APK yang tidak ditandatangani
- Masuk aplikasi Anda dengan kunci pribadi menggunakan jarsigner

```
$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1  
-keystore my-release-key.keystore my_application.apk alias_name
```

- Verifikasi bahwa APK Anda ditandatangani. Misalnya -

```
$ jarsigner -verify -verbose -certs my_application.apk
```

- Sejajarkan paket APK final menggunakan zipalign.

```
$ zipalign -v 4 your_project_name-unaligned.apk your_project_name.apk
```

References

Daring, M. P. (2018, Mei Senin). *Tutorials Point*. Retrieved from <https://www.tutorialspoint.com/android/index.htm>:

<https://www.tutorialspoint.com/android/index.htm>

Fathur, R. (2018, Mei Senin). *Situs Belajar Android*. Diambil kembali dari <https://www.okedroid.com/>: <https://www.okedroid.com/>

Google, T. P. (2018, Mei Senin). *Kursus Dasar-Dasar Developer Android — Praktik*. Retrieved from <https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-practicals/content/idn/>: <https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-practicals/content/idn/>