

Praktikum ke :15

Judul praktikum : Build APK

Alokasi waktu : 2 x 50 menit

1. Tujuan Instruksional Khusus

- a. Mahasiswa dapat memahami tentang build APK
- b. Mahasiswa dapat membuat sebuah Build APK

2. Teori

Build APK

Salah satu langkah terakhir yang perlu dilakukan setelah mengembangkan aplikasi Android adalah membuat berkas *executable* dalam format APK (Android Application Package). Berkas ini yang akan didistribusikan oleh Google Play ke pengguna. Jadi, ketika hendak mempublikasikan Aplikasi ke Google Play, berkas inilah yang harus unggah.

Jika belum memahami berkas APK, maka dapat menyamakannya dengan berkas *exe* di windows atau *ipa* di iOS.

Cara membuat file APK di Android terbilang cukup mudah. dapat menggunakan sebuah *wizard* atau melalui *command line*. Pada modul ini, kita akan fokus menggunakan *wizard*.

Membuat APK dapat dibagi menjadi 2:

1. Dengan menggunakan *default keystore*
2. Dengan menggunakan *custom keystore*

Keystore adalah sebuah berkas biner yang berisi informasi tentang satu atau lebih *private key*. *Private key* ini digunakan untuk mencegah pemalsuan aplikasi. Konsep umumnya adalah:

- Sistem Operasi Android mewajibkan semua APK *di-sign* sebelum terpasang ke dalam device.
- Proses *signing* ini membutuhkan *Public* dan *Private Key*.
- Proses *signing* ini berlangsung selama pembuatan APK dalam mode *debug* maupun *released*.
- Sebuah sertifikat *digital public key*, atau *identity certificate*, berisi informasi mengenai sertifikat itu sendiri dan *metadata* dari pemilik sertifikat tersebut. Pemilik sertifikat ini biasanya adalah developer yang mengembangkan aplikasi.
- *Public key* yang digunakan dalam proses *signing* di atas akan dilampirkan di dalam berkas APK. Proses ini dilakukan secara otomatis oleh Android Studio.
- Ketika hendak memperbarui Aplikasi pada Google Play, maka Google Play hanya akan menerimanya bila *keystore* yang digunakan sama dengan *keystore* yang pertama kali gunakan ketika mengunggah Aplikasi tersebut ke Google Play.

Kegunaan lain dari *keystore* adalah:

- Untuk integrasi ke layanan Google seperti Google Maps dengan menggunakan nilai *hash* (digest SHA1) di dalamnya.
- Untuk integrasi ke layanan API Facebook dengan menggunakan key hash base64 yang terkandung di dalam *keystore*.

Keystore merupakan sebuah berkas penting yang harus jaga, terlebih ketika aplikasi memiliki jumlah unduhan pengguna yang banyak. Sebabnya, kelalaian menjaga *keystore* ini dapat menghalangi untuk memperbarui aplikasi. Akibat terburuk adalah harus melakukannya dari awal lagi.

Berikut adalah tips yang bisa gunakan untuk mengamankan *keystore* :

1. Pilih kata kunci (password) yang sulit ditebak, kombinasikan angka, alfabet dan simbol dalam membuatnya.

2. Bedakan antara *keystore password* dan *key password* ketika membuat berkas APK dengan *custom keystore*.
3. Jangan memberikan *keystore* kepada orang yang tidak dipercaya apalagi meletakkannya didalam berkas proyek aplikasi.
4. Letakan ditempat yang kamu ingat dan pastikan aman.

Untuk memahami masalah di atas lebih jauh, dapat membaca tautan berikut:

<https://developer.android.com/studio/publish/app-signing.html>

Jika ingin menemukan default keystore , maka pengguna Mac dapat menemukannya di `~/.android/debug.keystore`. Sementara itu pengguna Windows bisa menemukannya di `C:\User\YourUser\.android\debug.keystore`.

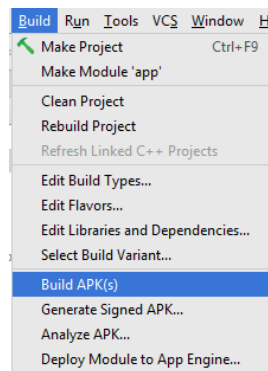
3. Alat dan Bahan

- a. BKPM
- b. Komputer
- c. LCD
- d. Alat Tulis Kantor (ATK)

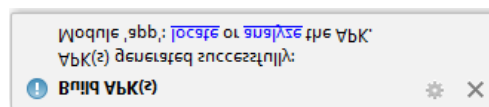
4. Pelaksanaan Praktikum

Untuk mulai melakukan proses *build* APK, dapat mengikuti langkah berikut:

1. Buka kembali project kosong yang telah kita buat sebelumnya.
2. Sekarang klik menu Build → Build APK



3. Gradle akan membuat berkas APK secara otomatis. Lama proses ini bergantung pada seberapa kompleks Aplikasi yang buat, dan jumlah *dependency* yang gunakan.
4. Ketika berhasil, dapat melihat notifikasi pada sudut kanan atas Android Studio:



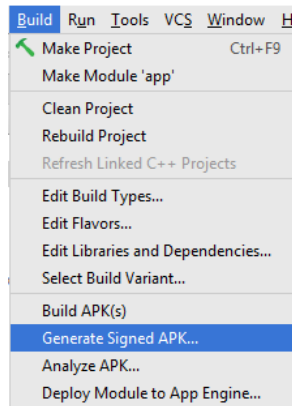
Sekarang tinggal tekan tautan yang terdapat pada notifikasi tersebut. Secara otomatis akan diarahkan ke lokasi di mana berkas APK disimpan.

Biasanya lokasinya mengikuti struktur `project-name/module-name/build/outputs/apk/`. Jika proyek bernama HelloWorld, maka lokasinya adalah **HelloWorld/app/build/outputs/apk/apk-debug.apk**

5. Sekarang sudah berhasil membuat APK dengan menggunakan default *keystore*. Ingat, APK yang baru saja buat akan ditolak oleh Google Play Store jika mencoba mengunggahnya ke Google Play Store. Agar dapat diterima, harus menjalankan proses *signing* atau generate APK tersebut dengan menggunakan *custom keystore*.
6. Sekarang coba pindahkan berkas APK yang baru dibuat ke dalam sebuah peranti (device). Buka lokasi berkas tersebut menggunakan *file explorer* pada device tersebut. Kemudian lakukan instalasi aplikasi seperti biasa.

Selamat, aplikasi Android baru sudah terpasang di peranti. Berkas APK ini bisa berikan ke pengguna lain untuk dicoba.

7. Mudah bukan? Sekarang kita lanjut membuat APK dengan *custom keystore*.
8. Kembali ke Proyek, klik Build → Generate Signed APK



9. Selanjutnya, pilih **create new**. Pada form yang tampil, lengkapi isian di dalamnya. Contoh pengisiannya adalah seperti gambar di bawah ini:

A screenshot of the 'New Key Store' dialog box. The dialog contains several fields for creating a new keystore. The 'Key store path' is set to 'C:\Users\Faisal\android\helloworld.jks'. The 'Password' and 'Confirm' fields are masked with dots. Under the 'Key' section, the 'Alias' is 'HelloWord', and the 'Validity (years)' is set to 25. The 'Certificate' section contains fields for personal and organizational information: 'First and Last Name' (Faisal Lutfi Afriansyah), 'Organizational Unit' (Jurusan Teknologi Informasi), 'Organization' (Politeknik Negeri Jember), 'City or Locality' (Jember), 'State or Province' (Jawa Timur), and 'Country Code (ISO)' (ID). 'OK' and 'Cancel' buttons are at the bottom right.

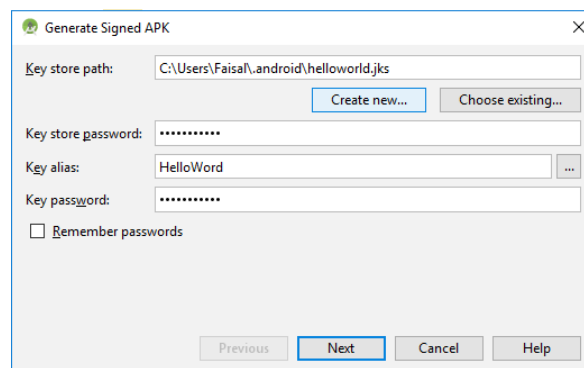
Berikut penjelasan tiap isian sebagai berikut:

- Keystore path : perlu menentukan dimana lokasi *keystore*
- Password : Isikan keystore password minimal 6 digit dan bedakan dengan keypassword dibawahnya
- Alias : Alias dari keystore

- Password : keypassword
- Validity : Berapa lama keystore akan valid (dalam hitungan tahun)
- Firstname hingga Country Code : isikan *metadata*. Penting untuk mengisi data ini dengan benar.

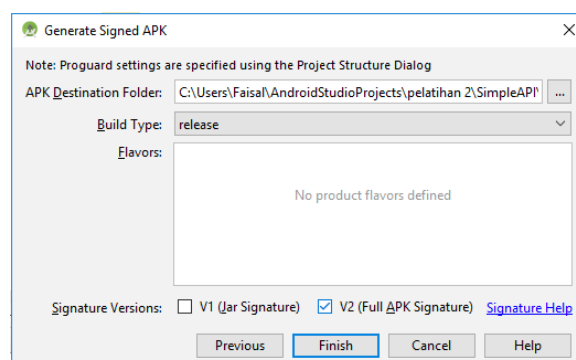
10. Setelah selesai klik **OK**.

11. Dialog yang di awal akan secara otomatis terisi ketika sudah berhasil mengisi form sebelumnya. Klik **next** untuk melanjutkan.



12. Jika ditanyakan kata kunci, masukan kata kunci yang gunakan untuk laptop atau komputer.

13. Selanjutnya tentukan di mana menyimpan APK yang dihasilkan. Di sini kami membiarkan secara *default*. Klik **finish** untuk memulai *generate signed APK*.



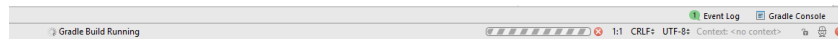
diharuskan memilih Signature V1 atau V2. Google menyarankan untuk memilih V2 karena *signature* jenis ini akan membuat instalasi APK lebih cepat. Selain itu, ia lebih aman terhadap pergantian (*alteration*) program

dengan tujuan yang tidak baik. Ikuti tautan ini untuk keterangan lebih lanjut.

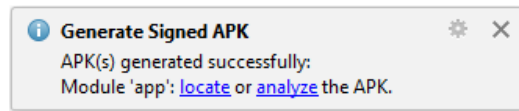
[https://developer.android.com/about/versions/nougat/android-](https://developer.android.com/about/versions/nougat/android-7.0#apk_signature_v2)

[7.0#apk_signature_v2](https://developer.android.com/about/versions/nougat/android-7.0#apk_signature_v2)

14. Perhatikan gradle process di status bar bagian bawah untuk melihat progress signed/generate APK.



15. Ketika berhasil, notifikasi seperti berikut akan tampil :



16. Selamat APK versi *released* telah berhasil dibuat. Proses ini perlu ketika hendak mempublikasikan aplikasi di Google Play Store dan memperbaruinya di kemudian waktu.

Ketika melakukan perbaruan aplikasi, jangan lupa untuk mengubah nilai yang ada di dalam build.gradle(Module:app):



5. Tugas

Build APK sample code kemudian running program pada device