**MAKALAH ESPRESSO**

****

**DISUSUN OLEH :**

**KELOMPOK 9**

**ZIKRULLAH ALIWAINDA 6706161013**

**HIDAYATULHIDAYAH 6706161103**

**SIDIK SUHENDAR 6706160073**

**D3IF-40-01**

**D3 TEKNIK INFORMATIKA**

**FAKULTAS ILMU TERAPAN**

**TELKOM UNIVERSITY**

**BANDUNG**

**2018**

**Espresso UI Testing**

1. **Pengertian**

Espresso adalah framework pengujian untuk Android yang memudahkan penguji dalam penulisan pengujian antarmuka pengguna (UI) yang andal untuk aplikasi. Framework, yang merupakan bagian dari *Android Support Repository*, menyediakan API untuk menulis hasil pengujian UI untuk menyimulasikan interaksi pengguna di dalam aplikasi, mulai dari mengklik tombol dan menavigasi tampilan sampai memilih pilihan menu.

1. **Petunjuk Pengaturan**

Saat melakukan pengujian dengan Espresso, terdapat beberapa hal yang harus diatur terlebih dahulu. Termasuk pengaturan pada *Environment* perangkat pengujian. Guna menghindari beberapa kesalahan seperti tidak terlihatnya dengan jelas hasil testing, sangat dianjurkan untuk mematikan 3 hal pada perangkat testing.

Pada perangkat pengujian pilih menu *Settings > Developer Options*, matikan atau *disable* 3 hal berikut :

* Window Animation Scale
* Transition Animation Scale
* Animator Duration Scale

Kemudian Tambahkan *Dependencies* pada projek dengan cara sebagai berikut :

* Buka projek aplikasi, pada file *build.gradle* atau app/build.gradle
* Tambahkan dependencies berikut :

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
androidTestImplementation 'com.android.support.test:runner:1.0.2'  
androidTestImplementation 'com.android.support.test:rules:1.0.2'

1. **Tujuan Espresso Testing**

Manusia cenderung melakukan kesalahan saat membangun aplikasi, itulah tujuan utama dilakukannya testing pada aplikasi. Adapun tujuan dari Espresso testing adalah

* Dapat mendeteksi error dan bugs selama proses pengembangan aplikasi
* Memastikan ketepatan dari keinginan pelanggan saat mengembangakan aplikasi
* Memastikan kualitas dari produk yang membantu untuk meraih kepercayaan dari pelanggan

1. **Espresso Test Recorder**

Salah satu jenis uji dari Espresso adalah menggunakan Test Recorder. Alat Espresso Test Recorder memungkinkan membuat pengujian UI untuk aplikasi tanpa menulis kode pengujian apa pun. Dengan merekam skenario pengujian, kita bisa merekam interaksi dengan sebuah perangkat dan menambahkan pernyataan untuk memverifikasi elemen UI dalam snapshot aplikasi tertentu. Espresso Test Recorder kemudian mengambil rekaman yang sudah disimpan dan secara otomatis menghasilkan pengujian UI yang sesuai dan bisa dijalankan untuk menguji aplikasi.

Espresso Test Recorder menulis pengujian berdasarkan [*framework Pengujian Espresso*](https://google.github.io/android-testing-support-library/docs/espresso/), sebuah API dalam [*Android Testing Support Library*](https://developer.android.com/topic/libraries/testing-support-library/index.html?hl=id). Espresso API mendorong Anda untuk membuat pengujian UI yang ringkas dan dapat diandalkan berdasarkan tindakan pengguna. Dengan menyatakan harapan, interaksi, dan pernyataan tanpa secara langsung mengakses aktivitas dan tampilan aplikasi yang mendasarinya, struktur ini mencegah gagalnya pengujian dan mengoptimalkan kecepatan pengujian.

Merekam Pengujian Espresso

Pengujian Espresso terdiri dari dua komponen utama: Interaksi UI dan pernyataan pada elemen View. Interaksi UI mencakup tindakan tap dan ketik yang mungkin digunakan pengguna untuk berinteraksi dengan aplikasi. Pernyataan memverifikasi keberadaan atau konten elemen visual pada layar. Misalnya, uji Espresso untuk [aplikasi pengujian Catatan](https://github.com/googlecodelabs/android-testing) mungkin mencakup interaksi UI untuk mengklik tombol dan menulis catatan baru tetapi menggunakan pernyataan untuk memverifikasi adanya tombol dan konten catatan.

Untuk menjalankan pengujian Espresso, gunakan jendela **Project** https://developer.android.com/studio/images/studio-icon.png?hl=id di sisi kiri Android Studio IDE:

1. Buka folder modul aplikasi yang diinginkan dan buka pengujian yang ingin jalankan. Lokasi pengujian tergantung pada lokasi akar [pengujian instrumentasi](https://developer.android.com/training/testing/unit-testing/instrumented-unit-tests.html?hl=id) dan nama paket aktivitas yang diluncurkan. Contoh berikut menunjukkan tempat penyimpanan pengujian untuk [aplikasi pengujian Catatan](https://github.com/googlecodelabs/android-testing):
   * Bila menggunakan [tampilan *Android*](https://developer.android.com/studio/projects/index.html?hl=id#ProjectFiles) di dalam jendela, buka *java > com.example.username.appname (androidTest)*.
   * Bila menggunakan tampilan *Project* di dalam jendela, buka *src > androidTest > java > com.example.username.appname* dalam folder modul.
2. Klik kanan pada pengujian dan klik **Run ‘testName.’**

Atau, bisa membuka file pengujian dan mengklik kanan metode atau kelas pengujian yang dihasilkan.

1. Pada jendela *Select Deployment Target*, pilih perangkat yang ingin Anda jalankan pengujiannya. Bila perlu, [buat](https://developer.android.com/studio/run/managing-avds.html?hl=id) *[Android Virtual Device](https://developer.android.com/studio/run/managing-avds.html?hl=id)* [baru](https://developer.android.com/studio/run/managing-avds.html?hl=id). Klik **OK**.

Monitor kemajuan pengujian di jendela **Run** di bagian bawah IDE. Android Studio menjalankan versi lengkap project dan membuka tab dengan nama pengujian di jendela **Run.** Kita juga bisa memeriksa apakah pengujian berhasil atau gagal dalam tab ini, serta seberapa lama pengujian itu berjalan. Saat pengujian selesai, tab akan mencatat log "Tests ran to completion."

Berikut terlampir link untuk mengakses video lengkap mengenai Espresso Testing :

<https://developer.android.com/studio/test/espresso-test-recorder?hl=id>