

# Mobile Programming

Pertemuan 09

# Android Studio

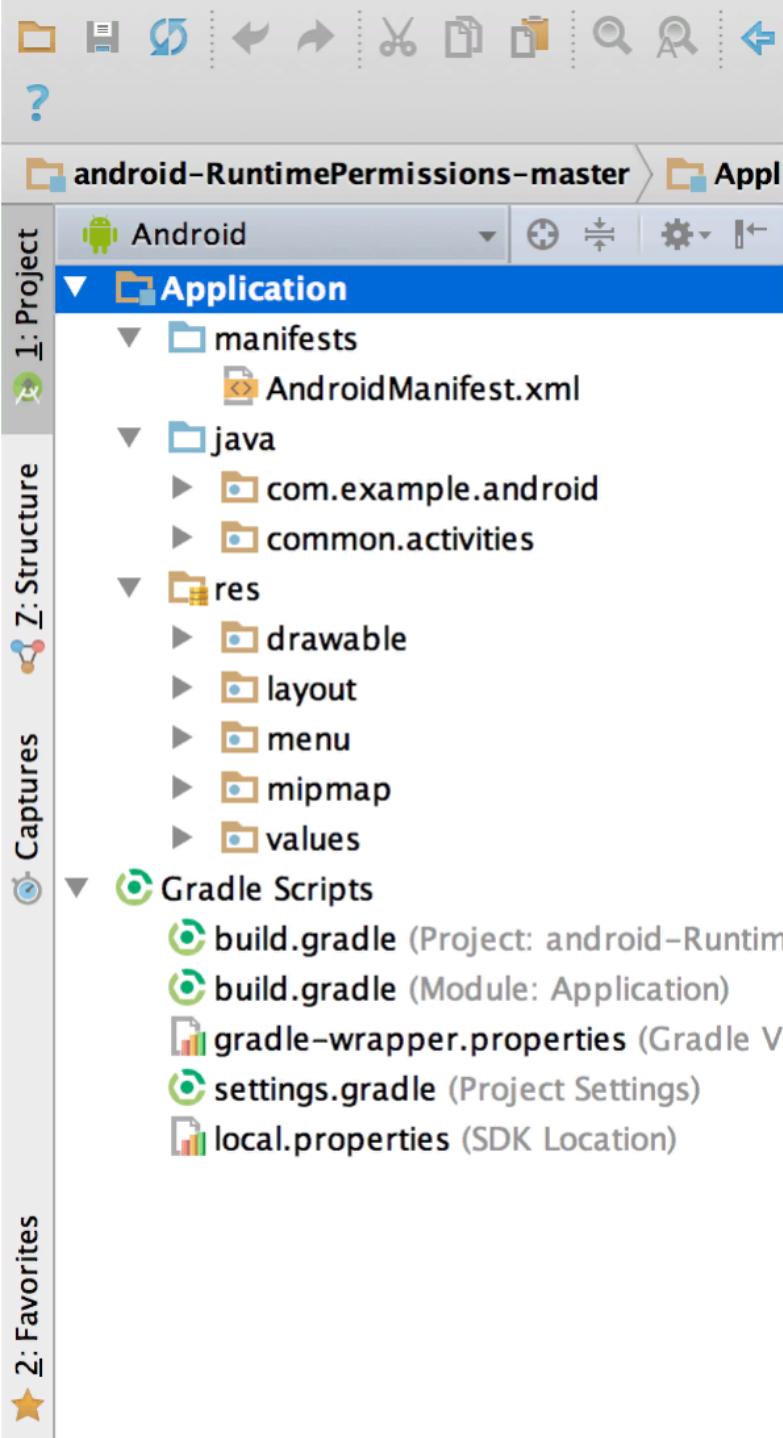
- Android Studio adalah Lingkungan Pengembangan Terpadu (IDE) resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA.
- Selain editor kode dan alat pengembang IntelliJ yang kuat, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas saat membangun aplikasi Android,

# Struktur Projek

- Setiap proyek di Android Studio berisi satu atau lebih modul dengan file kode sumber dan file sumber daya. Jenis modul meliputi:
  - Android app modules
  - Library modules
  - Google App Engine modules

# Cont'd

- Secara default, Android Studio menampilkan file proyek untuk memberikan akses cepat ke file sumber utama proyek.
- Semua file yang diperlukan untuk build terlihat di bawah folder Script Gradle



# Cont'd

- manifes: Berisi file AndroidManifest.xml.
- java: Berisi file kode sumber Java, termasuk kode uji JUnit.
- res: Berisi semua sumber daya non-kode, seperti layout XML, string UI, dan gambar bitmap.

# Sistem Gradle

- Android Studio menggunakan Gradle sebagai dasar dari sistem build, dengan lebih banyak kemampuan spesifik Android yang disediakan oleh plugin Android untuk Gradle.
- Sistem build ini berjalan sebagai alat terintegrasi dari menu Android Studio, dan secara independen dari baris perintah.



Android



Captures

1: Project

2: Structure



- ▼ app
  - manifests
  - java
  - res
- ▼ Gradle Scripts
  - build.gradle (Project: PDD)
  - build.gradle (Module: app)
  - proguard-rules.pro (ProGuard Rules for app)
  - gradle.properties (Project Properties) Selected
  - settings.gradle (Project Settings)
  - local.properties (SDK Location)

# Groovy Syntax

The screenshot shows a code editor interface with three tabs at the top: 'gradle-wrapper.properties', 'Birthday\_card', and 'Birthday\_card.iml'. The 'Birthday\_card' tab is active, displaying a Groovy build script. The script includes configuration for build scripts, repositories, dependencies, and tasks. A code completion tooltip is visible over the word 'individual' in the note about application dependencies.

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {
    ext.kotlin_version = '1.2.30'
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.1.3'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
    }
}

// NOTE: Do not place your application dependencies here; they belong
// in the individual module build.gradle files

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

# Manfaat Gradle

- Kustomisasi, konfigurasi, dan perpanjang proses pembuatan.
- Buat beberapa APK untuk aplikasi Anda, dengan berbagai fitur menggunakan proyek dan modul yang sama.
- Gunakan kembali kode dan sumber daya di seluruh sumber.

# Cont'd

- Dengan menggunakan fleksibilitas Gradle, pengembang dapat membuat aplikasi tanpa mengubah file sumber inti aplikasi.
- File build Android Studio diberi nama build.gradle.
- File teks biasa yang menggunakan sintaks Groovy untuk mengonfigurasi build dengan elemen yang disediakan oleh plugin Android untuk Gradle.

# Manfaat Gradle

- Varian Build
- Sistem build dapat membantu membuat versi berbeda dari aplikasi yang sama dari satu proyek.
- Ini berguna ketika aplikasi memiliki versi gratis dan versi berbayar, atau jika ingin mendistribusikan beberapa APK untuk berbagai konfigurasi perangkat di Google Play.

# Cont'd

- Dukungan beberapa APK
- Memungkinkan membuat banyak APK secara efisien berdasarkan kepadatan layar atau ABI (Application Binary Interface)
- Penyusutan sumber daya
- Android Studio secara otomatis menghapus sumber daya yang tidak digunakan dari dependensi aplikasi dan pustaka Anda.

# Cont'd

- Mengelola dependensi
- Ketergantungan untuk proyek ditentukan oleh file build.gradle. Gradle dengan hati-hati menemukan dependensi dan membuatnya tersedia.

# Debug dan Profiling

- Android Studio membantu dalam debugging dan meningkatkan kinerja kode Anda, termasuk debugging inline dan alat analisis kinerja.

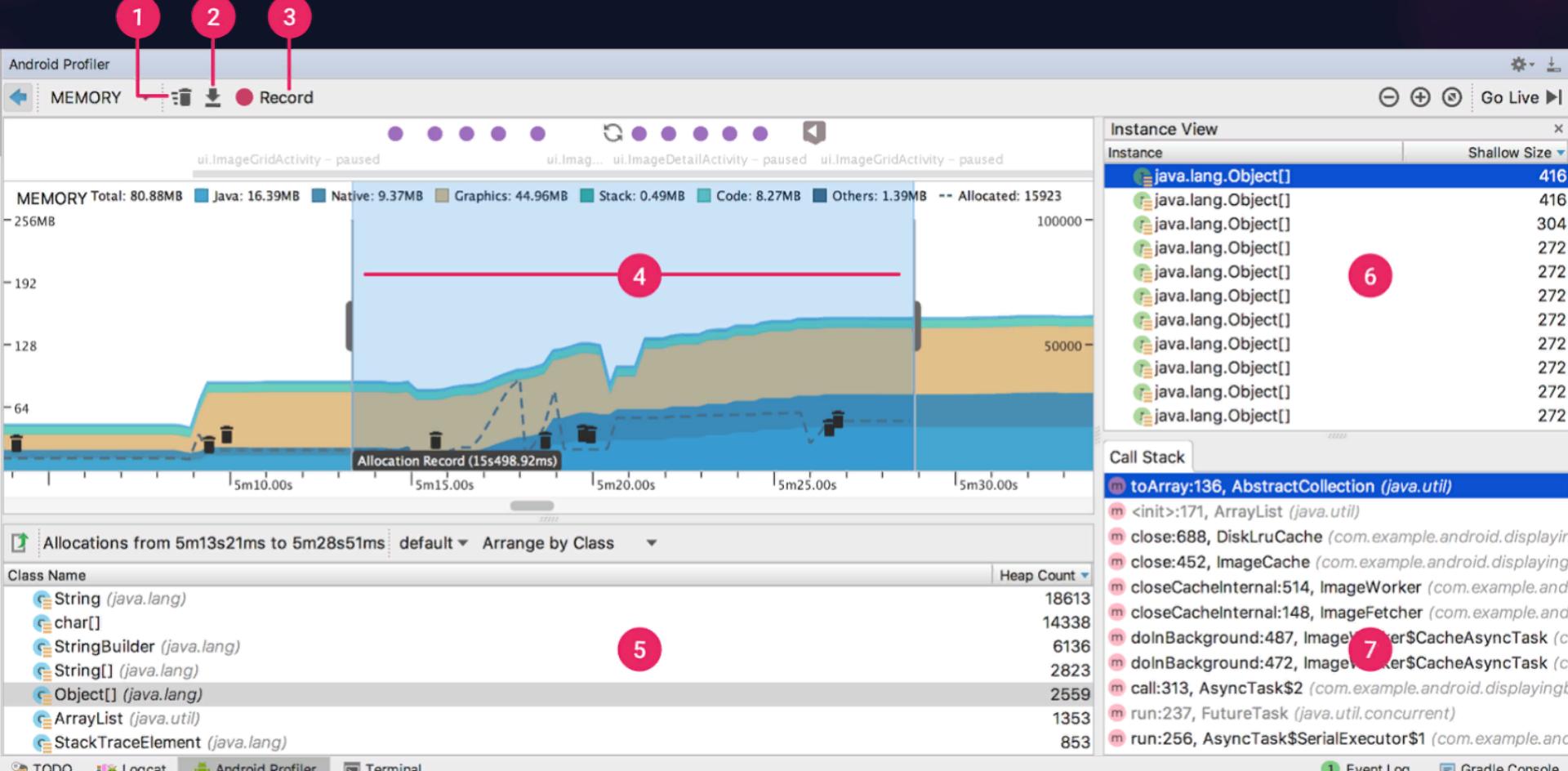
# Inline Debugging

- Gunakan debugging inline untuk penelusuran kode. Tampilan debugger terdapat verifikasi inline, ekspresi, dan nilai variabel. Informasi debug inline meliputi:
  - Nilai variabel sebaris
  - Objek yang mereferensikan objek yang dipilih
  - Metode mengembalikan nilai
  - Lambda dan ekspresi operator
  - Nilai tooltip

# Android Profiler

- Android Studio menyediakan profiler kinerja sehingga dapat lebih mudah melacak memori aplikasi dan penggunaan CPU, menemukan objek yang tidak dialokasikan, menemukan kebocoran memori, mengoptimalkan kinerja grafis, dan menganalisis permintaan jaringan.

# Contoh

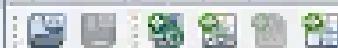


# Heap Dump

- Saat membuat profil penggunaan memori di Android Studio, Pengembang dapat secara bersamaan memulai pengumpulan sampah dan membuang tumpukan Java ke snapshot tumpukan dalam file format biner HPROF spesifik Android.

# Cont'd

- Snapshot berisi informasi tentang objek dan kelas Java di heap saat snapshot dipicu.
- Biasanya, pengumpulan sampah lengkap dipicu sebelum tumpukan timbunan ditulis, sehingga kumpulan berisi informasi tentang objek yang tersisa di tumpukan.



[heapdump] 8:42:23 PM



## [heapdump] 8:42:23 PM

## Heap Dump



Summary



Instances

[Compare with another heap dump](#)

Class Name	Instances (%)	Instances	Size
byte[]		15,284 (26%)	845,573 (28.6%)
char[]		12,948 (22%)	638,938 (21.6%)
java.lang.String		12,901 (21.9%)	258,020 (8.7%)
short[]		4,557 (7.7%)	238,088 (8%)
int[]		3,214 (5.5%)	685,268 (23.1%)
java.lang.Object[]		748 (1.3%)	23,880 (0.8%)
java.util.HashMap\$Entry		570 (1%)	13,680 (0.5%)
java.lang.reflect.Method		452 (0.8%)	32,996 (1.1%)
java.lang.Class[]		381 (0.6%)	4,196 (0.1%)
java.lang.ref.SoftReference		363 (0.6%)	11,616 (0.4%)
java.util.HashMap		312 (0.5%)	13,728 (0.5%)
java.lang.String[]		309 (0.5%)	6,360 (0.2%)
java.lang.Long		306 (0.5%)	4,896 (0.2%)
java.lang.Integer		281 (0.5%)	3,372 (0.1%)
java.util.Hashtable\$Entry		262 (0.4%)	6,288 (0.2%)
javax.management.Immut...		189 (0.3%)	3,780 (0.1%)

Class Name Filter (Contains)

# Memory Profiler

- Memory Profiler
- Pengembang dapat menggunakan Memory Profiler untuk melacak alokasi memori dan menonton di mana objek dialokasikan

# Data File Access

- Alat Android SDK, seperti Systrace, dan logcat, menghasilkan data kinerja dan debugging untuk analisis aplikasi terperinci.

1 Logcat:

2 02-02 18:52:57.132: VERBOSE/ProtocolEngine(24): DownloadRate 104166 bytes per sec. Downl  
3 08-03 13:31:16.196: DEBUG/dalvikvm(2227): HeapWorker thread shutting down  
4 08-03 13:31:16.756: INFO/dalvikvm(2234): Debugger is active  
5 08-03 16:26:45.965: WARN/ActivityManager(564): Launch timeout has expired, giving up wake  
6 08-04 16:19:11.166: ERROR/AndroidRuntime(4687): Uncaught handler: thread main exiting due  
7 08-04 16:24:11.166: ASSERT/Assertion(4687): Expected true but was false

1 Logcat:

2 02-02 18:52:57.132: VERBOSE/ProtocolEngine(24): DownloadRate 104166 bytes per sec. Downl  
3 08-03 13:31:16.196: DEBUG/dalvikvm(2227): HeapWorker thread shutting down  
4 08-03 13:31:16.756: INFO/dalvikvm(2234): Debugger is active  
5 08-03 16:26:45.965: WARN/ActivityManager(564): Launch timeout has expired, giving up wake  
6 08-04 16:19:11.166: ERROR/AndroidRuntime(4687): Uncaught handler: thread main exiting due  
7 08-04 16:24:11.166: ASSERT/Assertion(4687): Expected true but was false

1 Logcat:

2 02-02 18:52:57.132: VERBOSE/ProtocolEngine(24): DownloadRate 104166 bytes per sec. Downl  
3 08-03 13:31:16.196: DEBUG/dalvikvm(2227): HeapWorker thread shutting down  
4 08-03 13:31:16.756: INFO/dalvikvm(2234): Debugger is active  
5 08-03 16:26:45.965: WARN/ActivityManager(564): Launch timeout has expired, giving up wak  
6 08-04 16:19:11.166: ERROR/AndroidRuntime(4687): Uncaught handler: thread main exiting du  
7 08-04 16:24:11.166: ASSERT/Assertion(4687): Expected true but was false

# Code Inspection

- Setiap kali Pengembang mengkompilasi program, Android Studio secara otomatis menjalankan Lint yang telah dikonfigurasi dan inspeksi IDE lainnya untuk membantu mengidentifikasi dan memperbaiki masalah dengan kualitas struktural kode

# Lint

Lint Warnings

3 errors, 18 warnings

Description

- ▶ "hello" is not translated in de [\(2 items\)](#)
- ▶ Incorrect formatting string delete\_deck\_success; missing conversion character
- ▶ Duplicate id @+id/button1, defined or included multiple times in layout/foo
- ▶ Use fraction character ¼ (&#190;) instead of 3/4 ?
- ▶ This text field does not specify an inputType or a hint
- ▼ [I18N] Hardcoded string "Button", should use @string resource [\(5 items\)](#)
  - ▶ [I18N] Hardcoded string "Button", should use @string resource
  - ▶ [I18N] Hardcoded string "Button", should use @string resource
  - ▶ [I18N] Hardcoded string "Button", should use @string resource
  - ▶ [I18N] Hardcoded string "Button", should use @string resource
  - ▶ [I18N] Hardcoded string "Button", should use @string resource

# Anotasi

- Android Studio mendukung anotasi untuk variabel, parameter, dan mengembalikan nilai untuk membantu Pengembang menangkap bug, seperti pengecualian null pointer dan konflik tipe sumber daya.

```
public class Processor extends AbstractProcessor {  
    public Processor() {  
        System.out.println("XXX: constructor");  
    }  
  
    @Override  
    public synchronized void init(ProcessingEnvironment processingEnvironment) {  
        super.init(processingEnvironment);  
        System.out.println("XXX: init");  
    }  
  
    @Override  
    public boolean process(Set<? extends TypeElement> set, RoundEnvironment roundEnvironment) {  
        for (Element e : set) {  
            System.out.println("XXX: " + e.getSimpleName());  
        }  
        return false;  
    }  
  
    @Override  
    public Set<String> getSupportedAnnotationTypes() {  
        HashSet<String> set = new HashSet<>();  
        set.add(Getter.class.getName());  
        return set;  
    }  
  
    @Override  
    public SourceVersion getSupportedSourceVersion() {  
        return SourceVersion.latestSupported();  
    }  
}
```

# Pesan Log

- Saat membangun dan menjalankan aplikasi dengan Android Studio, Pengembang dapat melihat pesan adb dan pesan log perangkat di jendela Logcat.

