

**LAPORAN RESMI**  
**MODUL I**  
**PENGENALAN PEMROGRAMAN BERGERAK**  
**PEMROGRAMAN BERGERAK**



|                      |                                     |
|----------------------|-------------------------------------|
| <b>NAMA</b>          | <b>: SEVIN DIAS ANDIKA</b>          |
| <b>N.R.P</b>         | <b>: 210441100105</b>               |
| <b>DOSEN</b>         | <b>: Ir. ACH. DAFID, S.T., M.T.</b> |
| <b>ASISTEN</b>       | <b>: DAVID NASRULLOH</b>            |
| <b>TGL PRAKTIKUM</b> | <b>: 17 MARET 2023</b>              |

Disetujui : .. ..... 2022  
Asisten

**DAVID NASRULLOH**  
**190441100060**



**LABORATORIUM BISNIS INTELIJEN SISTEM**  
**PRODI SISTEM INFORMASI**  
**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pemrograman bergerak adalah salah satu jenis pemrograman yang berkaitan dengan pengembangan aplikasi dan perangkat lunak yang dapat dijalankan pada perangkat mobile seperti smartphone dan tablet. Seiring dengan perkembangan teknologi yang semakin pesat, penggunaan perangkat mobile semakin meningkat drastis, sehingga memunculkan kebutuhan yang besar untuk aplikasi mobile yang lebih banyak dan bervariasi. Oleh karena itu, pemrograman bergerak menjadi salah satu bidang yang sangat penting untuk dipelajari. Di dalam bidang pemrograman bergerak, kita akan mempelajari bahasa pemrograman Kotlin yang dirancang untuk membangun aplikasi mobile pada platform Android dan iOS.

Pada materi pengenalan pemrograman bergerak, kita akan membahas tentang dasar-dasar pemrograman bergerak serta beberapa konsep dasar yang perlu dipahami untuk memulai membangun aplikasi mobile. Beberapa konsep dasar tersebut antara lain adalah aplikasi mobile, UI design, database, serta penggunaan library dan framework. Dalam membangun aplikasi mobile, penting untuk memahami konsep-konsep dasar tersebut, karena akan mempengaruhi kualitas dan performa dari aplikasi yang dibangun. Selain itu, pemahaman tentang konsep-konsep dasar tersebut akan membantu mempermudah proses pengembangan aplikasi mobile pada platform yang berbeda.

### **1.2 Tujuan**

- Memahami dan mengetahui tentang aplikasi mobile, cara penginstallan Android Studio, debug dengan virtual device dan debug dengan physical device.
- Mampu Memahami tentang Pengembangan Aplikasi Android.
- Mampu Membuat aplikasi sederhana dengan desain standard yang disediakan dan menjalankan aplikasi di emulator maupun di perangkat mobile (keluaran).

## **BAB II**

### **DASAR TEORI**

#### **2.1 Pendahuluan**

Coba perhatikan kalian kalo nulis itu kayak gini, pokoknya kayak gini Coba perhatikan kalian kalo nulis itu kayak gini, pokoknya kayak gini Coba perhatikan kalian kalo nulis itu kayak gini, pokoknya kayak gini Selamat datang di Praktikum Pemrograman Bergerak. Kita akan menggunakan Android Kotlin untuk praktikum ini. Praktikum ini menggunakan codelab yang menuntun kita membangun aplikasi Android menggunakan Kotlin. Prasyarat untuk dapat mengikuti praktikum ini dengan baik adalah memiliki pengetahuan dalam bahasa pemrograman berorientasi objek penuh seperti Java, C++. Untuk pengembangan, disarankan untuk menggunakan sumber referensi selain modul praktikum ini.

Kita juga harus terbiasa menavigasi GitHub dan terbiasa dengan konsep-konsep: Algoritma pemrograman, penanganan error, bagaimana kode dibangun, dikompilasi, dan dieksekusi, secara umum.

Kita akan menggunakan sumber referensi utama pada link android developer dan codelab berikut:

- 3 <https://kotlinlang.org/docs/reference/> untuk belajar dasar pemrograman kotlin
- 4 <https://developer.android.com/kotlin> untuk pemahaman kenapa android menggunakan kotlin.
- 5 <https://developer.android.com/courses/kotlin-android-fundamentals/toc> untuk codelab review untuk dasar pemrograman kotlin.
- 6 <https://codelabs.developers.google.com/android-kotlin-fundamentals/> untuk belajar koding project kotlin.

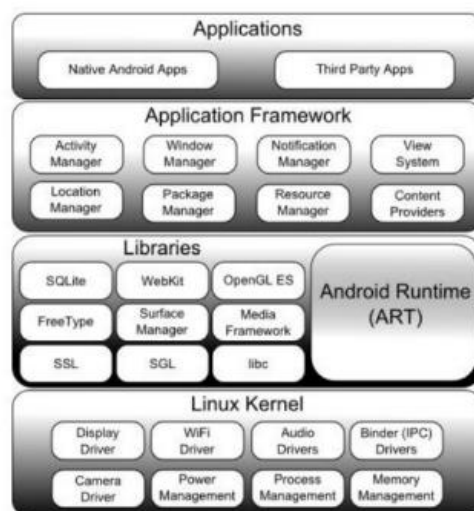
#### **2.2 Android dan Pemrograman Bergerak**

Android adalah sistem operasi mobile yang open source. Tahun 2005, Google mengakuisisi perusahaan Android Inc. untuk memulai mengembangkan platform Android. Tahun 2007, sekelompok pemimpin industri datang bersama membentuk

Open Handset Alliance (<http://www.openhandsetalliance.com>). November 2007, Android SDK dirilis pertama kali dengan “tampilan awal” (early look). September 2008, T-Mobile mengumumkan ketersediaan HTC Dream G1, smartphone pertama yang berbasis platform Android. •Beberapa hari berikutnya Google mengumumkan ketersediaan Android SDK Release Candidate 1.0. Oktober 2008, Google membuat kode program dari platform Android tersedia di bawah ”Apache’s open source license”.

Android adalah platform mobile pertama yang lengkap, terbuka dan bebas. Sistem operasi yang mendasari android dilisensikan dibawah GNU, General Public Lisensi Versi 2 (GPL), yang sering dikenal dengan istilah “copyleft” lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah terms. Android didistribusikan dibawah lisensi Apache Software (ASL/Apache2), yang memungkinkan dan seterusnya.

Android disusun dalam bentuk software stack yang terdiri dari aplikasi, sistem operasi, lingkungan run-time, middleware, layanan dan pustaka (library). Setiap lapisan dari tumpukan, dan unsur-unsur yang sesuai dalam setiap lapisan, saling terintegrasi untuk memberikan pengembangan aplikasi dan lingkungan eksekusi yang optimal untuk perangkat mobile. Arsitektur ini ditampilkan pada Gambar 1-1. ntuk distribusi kedua dan seterusnya.



Gambar Arsitektur Aplikasi Android

Dalam rangka mengembangkan aplikasi Android di Android Studio akan diperlukan untuk mengkompilasi dan menjalankan aplikasi beberapa kali. Aplikasi Android dapat diuji dengan menginstal dan menjalankannya baik pada perangkat fisik atau dalam Virtual Device (AVD) lingkungan emulator Android. Sebelum AVD dapat digunakan, terlebih dahulu harus dibuat dan dikonfigurasi untuk mencocokkan spesifikasi model perangkat tertentu. Tujuan dari bab ini, oleh karena itu, adalah untuk bekerja melalui langkah-langkah yang terlibat dalam menciptakan suatu perangkat virtual menggunakan Nexus 7 tablet sebagai contoh referensi.

AVD pada dasarnya adalah emulator yang memungkinkan aplikasi Android untuk diuji tanpa perlu menginstal aplikasi pada perangkat berbasis Android fisik. Sebuah AVD dapat dikonfigurasi untuk meniru berbagai fitur perangkat keras termasuk pilihan ukuran layar, kapasitas memori dan fitur seperti kamera, dukungan navigasi GPS atau accelerometer. Sebagai bagian dari instalasi Android Studio standar, memungkinkan emulator dipasang dan dikonfigurasi untuk berbagai perangkat yang berbeda. AVD baru dibuat dan dikelola dengan menggunakan Android Virtual Device Manager, yang dapat digunakan dalam mode baris perintah atau dengan lebih antarmuka grafis yang userfriendly.

Programmer yang menggunakan bahasa pemrograman berorientasi objek seperti Java akan terbiasa karena aplikasi Android ditulis di Kotlin, ini masih sangat banyak terjadi. Android, bagaimanapun juga, mengambil konsep yang dapat digunakan kembali komponen ke tingkat yang lebih tinggi.

Aplikasi Android diciptakan dengan menggunakan satu atau lebih komponen bersama, yang dikenal sebagai Activity. Sebuah Activity adalah satu, modul mandiri dari aplikasi yang biasanya berkorelasi langsung ke layar antarmuka pengguna. Activity dimaksudkan sebagai komponen, yang dapat digunakan kembali dan dapat dipertukarkan, dan bisa dibagi di antara aplikasi yang berbeda. Sebuah aplikasi email yang ada, misalnya, mungkin berisi Activity khusus untuk membuat dan mengirim pesan email. Seorang pengembang mungkin menulis sebuah aplikasi yang juga memiliki persyaratan untuk mengirim pesan email. Daripada mengembangkan Activity komposisi email khusus untuk aplikasi baru, pengembang hanya dapat menggunakan Activity dari aplikasi email yang ada.

Activity diciptakan sebagai subclass dari kelas Activity Android dan harus dieksekusi sehingga menjadi terpisah sepenuhnya dari Activity lain dalam aplikasi. Dengan kata lain, Activity bersama tidak bisa dipanggil langsung dalam program (karena aplikasi lain dapat menggunakan Aktivitas) dan satu Activity tidak bisa langsung memanggil metode atau mengakses data Activity lain. Sebagai gantinya, untuk mencapai tujuan ini, dengan menggunakan Intents dan Content Providers. Secara default, suatu Activity tidak dapat memberikan hasil dengan aktivitas yang ia dipanggil. Jika fungsi ini diperlukan, Activity harus secara khusus dimulai sebagai sub-aktivitas.

Aplikasi manifest file yang mengatur berbagai elemen dalam aplikasi adalah file Manifest. Berkas Manifest berbasis XML ini, menguraikan Activity, Service, Content Provider dan permission yang membentuk suatu aplikasi secara lengkap. Selain file Manifest dan file Dex yang berisi kode-kode byte, paket aplikasi Android biasanya berisi kumpulan berkas Resources (sumber daya). Berkas ini mengandung sumber daya seperti string, gambar, huruf dan warna yang muncul dalam antarmuka pengguna secara bersama-sama, dengan representasi XML layout antarmuka pengguna. Secara default, berkas ini disimpan dalam /res, sub-direktori dalam hirarki proyek aplikasi.

Bila aplikasi dikompilasi, kelas bernama R dibuat, yang berisi referensi ke sumber daya aplikasi. File manifest dan sumber daya ini digabungkan untuk membuat apa yang dikenal sebagai Konteks Aplikasi. Konteks ini, diwakili oleh kelas Context Android, dapat digunakan dalam kode aplikasi untuk mendapatkan akses ke sumber daya aplikasi pada saat runtime. Selain itu, berbagai metode dapat dipanggil pada konteks aplikasi untuk mengumpulkan informasi dan membuat perubahan pada lingkungan aplikasi pada saat runtime.

### **2.2.1 Jelajahi file activity dan layout**

Kita akan fokus pada dua file paling penting yang membentuk aplikasi kita: File MainActivity Kotlin, dan file layout activity\_main.xml.

#### **Langkah 1: Periksa MainActivity**

MainActivity adalah contoh Activity. Suatu Activity adalah kelas inti Android yang menggambar antarmuka pengguna aplikasi Android (UI) dan

menerima acara masukan. Saat aplikasi kita diluncurkan, aplikasi meluncurkan aktivitas yang ditentukan dalam file `AndroidManifest.xml`. Banyak bahasa pemrograman menentukan metode utama yang memulai program. Aplikasi Android tidak memiliki metode utama. Sebaliknya, file `AndroidManifest.xml` menunjukkan bahwa `MainActivity` harus diluncurkan ketika pengguna mengetuk ikon peluncur aplikasi. Untuk meluncurkan suatu kegiatan, OS Android menggunakan informasi dalam manifest untuk mengatur lingkungan aplikasi dan membangun `MainActivity`. Kemudian `MainActivity` melakukan beberapa pengaturan secara bergantian. Setiap aktivitas memiliki file layout terkait. Activity dan layout dihubungkan oleh proses yang dikenal sebagai layout inflasi. Saat Activity dimulai, tampilan yang didefinisikan dalam file tata letak XML diubah menjadi (atau "digelembungkan" menjadi) objek tampilan Kotlin di memori. Setelah ini terjadi, Activity dapat menarik objek-objek ini ke layar dan juga secara dinamis memodifikasinya.

## **Run Emulator**

Uji coba aplikasi wajib dilakukan seorang developer. Proses running atau debugging bisa dilakukan dengan dua cara, yaitu running dengan emulator atau peranti (device). Baik emulator maupun peranti memiliki kelebihan dan kekurangan masing-masing. Kita sebagai developer tinggal pilih mana yang sesuai keperluan.

Persiapan Running Menggunakan Emulator Sebelum menggunakan emulator, pastikan beberapa hal berikut ini:

### **1. Virtualization**

Untuk menjalankan emulator di dalam Android Studio, pastikan aspek virtualization. Sistem kita harus memenuhi persyaratannya, yakni ketentuan prosesor dan sistem operasi dari laptop / PC yang kita gunakan.

### **2. Processor**

- **Prosesor Intel:** Jika laptop/pc kita menggunakan prosesor Intel, maka pastikan ia mendukung Intel VT-x, Intel EM64T (Intel 64), dan Execute Disable (XD) Bit functionality.

- Prosesor AMD: Jika laptop/pc kita menggunakan AMD, maka pastikan bahwa ia support dengan AMD Virtualization (AMD-V) dan Supplemental Streaming SIMD Extensions 3 (SSSE3).

### 3. Sistem Operasi

- Intel : Jika menggunakan processor Intel maka kita dapat menjalankannya di sistem operasi Windows, Linux, maupun Mac.
- AMD : Untuk prosesor AMD maka hanya bisa menjalankannya di sistem operasi Linux.

### Menginstal Hardware Accelerated Execution Manager (HAXM)

Setelah memenuhi persyaratan di atas, langkah selanjutnya adalah menginstal HAXM. HAXM adalah hardware-assisted virtualization engine yang menggunakan teknologi VT dari Intel untuk mempercepat aplikasi Android yang diemulasi di mesin host. HAXM diperlukan untuk menjalankan emulator di Android Studio.

HAXM diperlukan jika sistem operasi yang kita gunakan adalah Windows atau Mac. Untuk menginstalnya, ikuti petunjuk berikut ini.

1. Buka SDK Manager.
2. Pilih SDK Update Sites, kemudian hidupkan Intel HAXM.
3. Tekan OK.
4. Cari berkas installer-nya di directory folder sdk kaomputer Anda, `~sdk\extras\intel\Hardware_Accelerated_Execution_Manager\intelhaxman droid.exe`.
5. Jalankan installer dan ikuti petunjuknya sampai selesai.



## BAB III

### TUGAS PENDAHULUAN

#### 3.1 Soal

1. Jelaskan apa yang anda ketahui tentang SDK?
2. Apa yang dimaksud Manifest dan digunakan untuk apa?
3. Apa kegunaan Gradle? Jelaskan!
4. Buat Program di new Activity, dari tulisan Hello world rubahlah dengan tulisan "Selamat datang di Universitas Trunodoyo Madura" serta rubah background dengan warna #2775A3

#### 3.2 Jawaban

1. SDK adalah kumpulan perangkat lunak yang digunakan untuk mengembangkan aplikasi untuk perangkat atau sistem operasi tertentu.
2. Manifest adalah dokumen sarana pengangkut yang berupa daftar muatan barang-barang yang diangkut. Manifest akan digunakan oleh orang-orang atau lembaga yang memiliki kepentingan dan digunakan untuk memastikan bahwa penumpang dan kargo yang terdaftar telah berada di dalam transportasi dan siap diangkut menuju tempat tujuan.
3. Gradle adalah build tools yang digunakan pada Android Studio, untuk compile - menjalankan Project Aplikasi yang sedang dikembangkan, berdasarkan settingan atau konfigurasi yang berada di bagian Gradle Script.

```
4. <?xml version="1.0" encoding="utf-8"?>
  <androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#2775A3"
    tools:context=".MainActivity">

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="“Selamat datang di Universitas Trunojoyo Madura”"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

  </androidx.constraintlayout.widget.ConstraintLayout>
```

## BAB IV

### IMPLEMENTASI

#### 4.1 Source Code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
<ImageView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:scaleType="centerCrop"
    android:src="@drawable/background"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
```

```
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="572dp"
    android:gravity="center_horizontal"
    android:text="Happy Birthday, My Friend !!!"
    android:textColor="@color/black"
    android:textSize="22sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="488dp"
    android:gravity="center_horizontal"
    android:text="20th"
```

```
android:textColor="@color/black"  
android:textSize="50sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.0"  
app:layout_constraintStart_toStartOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## 4.2 Hasil



## **BAB V**

### **PENUTUP**

#### **5.1 Analisa**

Dari hasil praktikum, praktikan menganalisa bahwa Android adalah sistem operasi mobile yang open source. Tahun 2005, Google mengakusisi perusahaan Android Inc. untuk memulai mengembangkan platform Android. Tahun 2007, sekelompok pemimpin industri datang bersama membentuk Open Handset Alliance (<http://www.openhandsetalliance.com>). November 2007, Android SDK dirilis pertama kali dengan “tampilan awal” (early look). September 2008, T-Mobile mengumumkan ketersediaan HTC Dream G1, smartphone pertama yang berbasis platform Android. •Beberapa hari berikutnya Google mengumumkan ketersediaan Android SDK Release Candidate 1.0. Oktober 2008, Google membuat kode program dari platform Android tersedia di bawah “Apache’s open source license”.

Dalam mengembangkan aplikasi mobile, terdapat banyak tantangan yang akan dihadapi. Namun, dengan pemahaman yang baik tentang dasar-dasar pemrograman bergerak dan kemampuan yang terus dikembangkan, kita dapat mengatasi tantangan tersebut dan membangun aplikasi mobile yang berkualitas tinggi.

#### **5.2 Kesimpulan**

1. Pemrograman bergerak adalah bidang yang sangat penting dan berkembang pesat dalam era digital yang semakin maju. Kita perlu mempelajari dasar-dasar pemrograman bergerak serta konsep-konsep dasar yang perlu dipahami untuk memulai membangun aplikasi mobile.
2. AVD adalah emulator yang memungkinkan aplikasi Android untuk diuji tanpa perlu menginstal aplikasi pada perangkat berbasis Android fisik.
3. Android adalah platform mobile pertama yang lengkap, terbuka dan bebas.