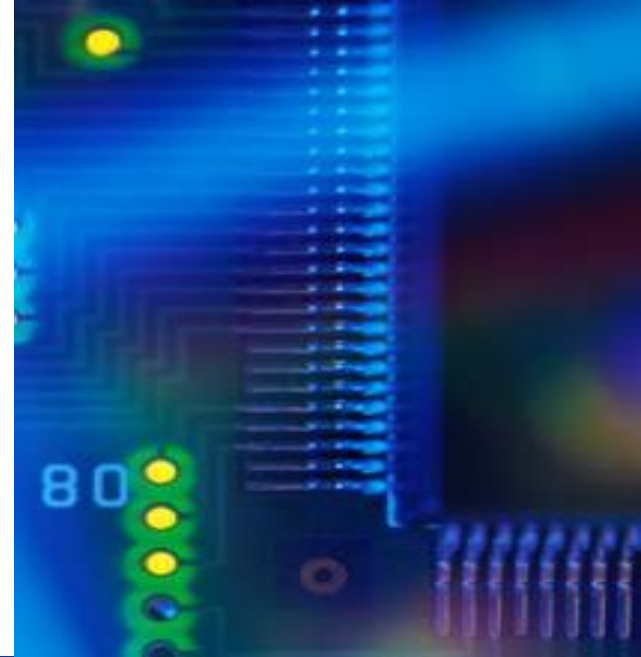




KEMENTERIAN KOMUNIKASI DAN INFORMATIKA
REPUBLIK INDONESIA

Menuju Masyarakat Informasi Indonesia



JUNIOR MOBILE PROGRAMMER

Menunjukkan platform operating system dan bahasa pemrograman di dalam perangkat lunak

Deskripsi Singkat

Deskripsi Singkat mengenai Topik

Topik ini menjelaskan bahasa pemrograman berbasis mobile (Dasar Java untuk mobile programming)

Tujuan Pelatihan

1. Peserta mampu menentukan mobile pemrograman berbasis mobile jenis bahasa pemrogramannya.
2. Peserta mampu membandingkan perbedaannya bahasa pemrograman berbasis mobile.
3. Peserta mampu Mengkonfigurasi Perangkat lunak terkait penggunaan bahasa pemrograman berbasis mobile sesuai dengan spesifikasinya.
4. Peserta mampu Menghasilkan Alur program untuk pembuatan aplikasi berbasis mobile.
5. Menentukan tipe-data variabel dan konstanta dalam salah satu bahasa pemrograman berbasis mobile
6. Menentukan konsep struktur kondisi dan perulangan dalam salah satu bahasa pemrograman berbasis mobile.
7. Menjelaskan Konsep layout dan objek dalam salah satu bahasa pemrograman berbasis mobile.
8. Membangun aplikasi mobile sederhana dengan bahasa pemrograman mobile.

Materi Yang akan disampaikan:

9. Kategori dan Jenis Bahasa Pemrograman
10. Pengaturan Ruang Kerja
11. Dasar Alur Pembuatan Software
12. Konsep Variabel dan Konstanta
13. Struktur Kondisi dan Perulangan
14. Konsep Layout
15. Membuat Sebuah Aplikasi Sederhana

Kategori Aplikasi dan Jenis Bahasa Pemrograman

Kategori Aplikasi

Pelatihan

Native mobile application

dikembangkan dalam bahasa pemrograman yang berasal dari perangkat dan sistem operasi, dan perlu membuat sebuah aplikasi untuk platform tertentu.

Hybrid mobile application

adalah aplikasi lintas platform namun membuat antarmuka pengguna menggunakan browser web, memanfaatkan HTML, CSS, dan Javascript.

Cross-platform mobile application

dikembangkan menggunakan Bahasa seperti Javascript, dan bukan native terhadap sistem operasi smartphone.

Kategori dan Jenis Bahasa Pemrograman

Pelatihan

Native mobile apps

- iOS (Objective-c or Swift)
- Android (Java)
- Windows Phone (C#/Visual Basic & XAML).

Cross-platform apps

- Unity
- Xamarin
- Appcelerator
- React Native
- NativeScript

Hybrid apps

- Cordova
- Trigger.IO
- Ionic Framework
- Framework7
- Telerik Platform

Pengaturan environment

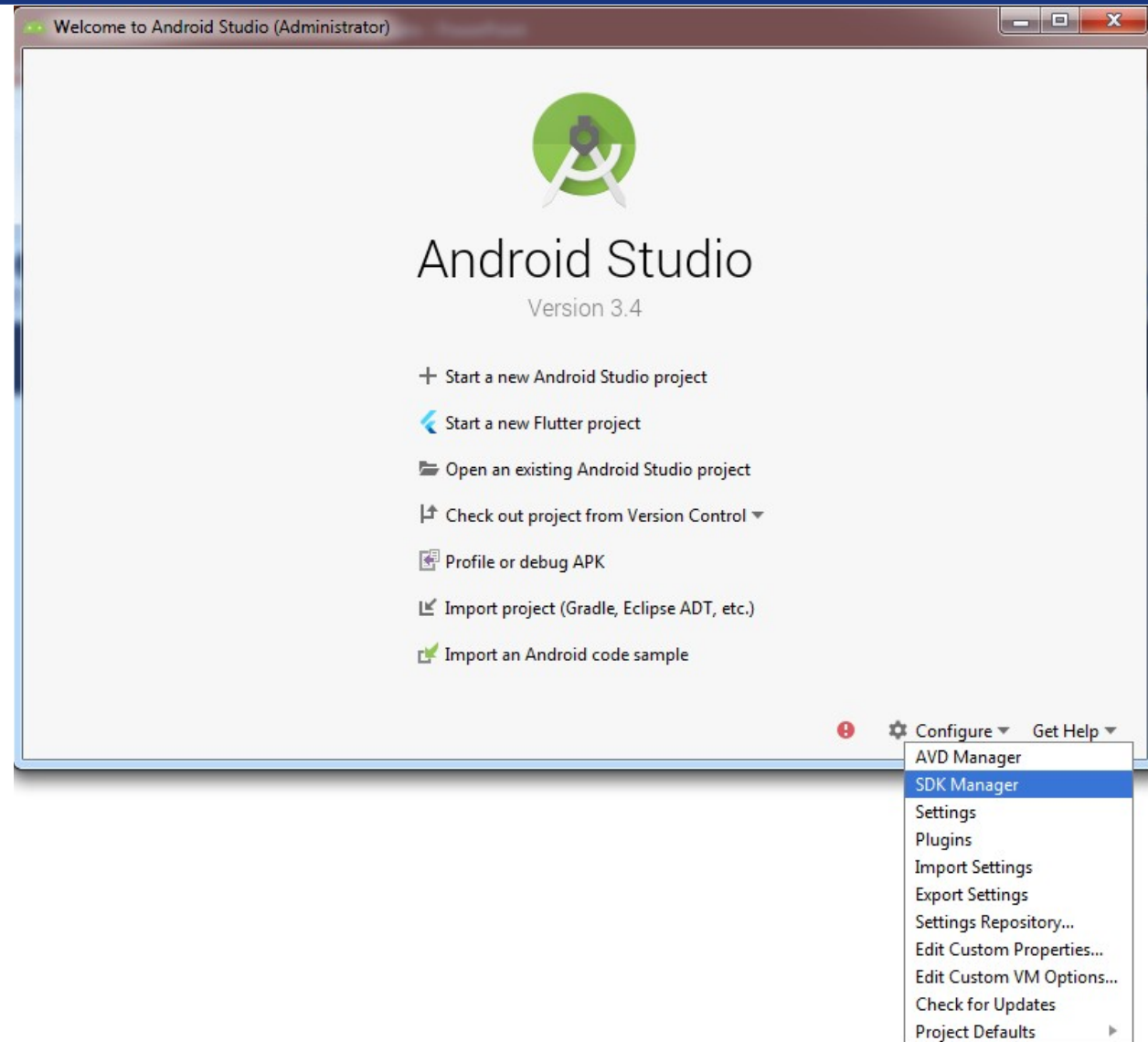
Pelatihan

1. Java Development Kit (JDK)
2. Android Studio
3. Software Development Kit (SDK)

Instalasi Software Development Kit

Pelatihan

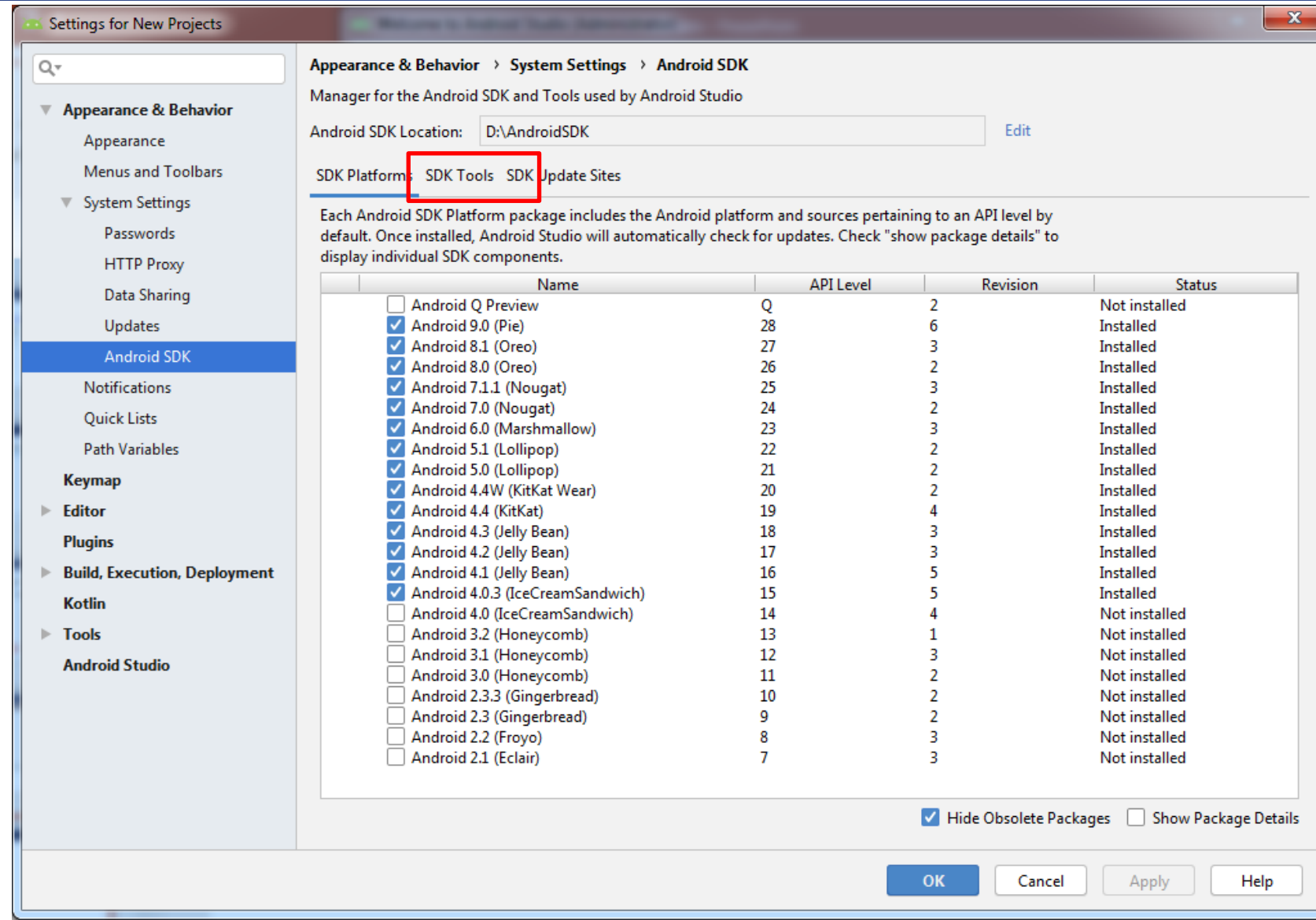
1. Buka android Studio
2. Pilih SDK Manager



Instalasi Software Development Kit

Pelatihan

3. Pilih SDK Tools

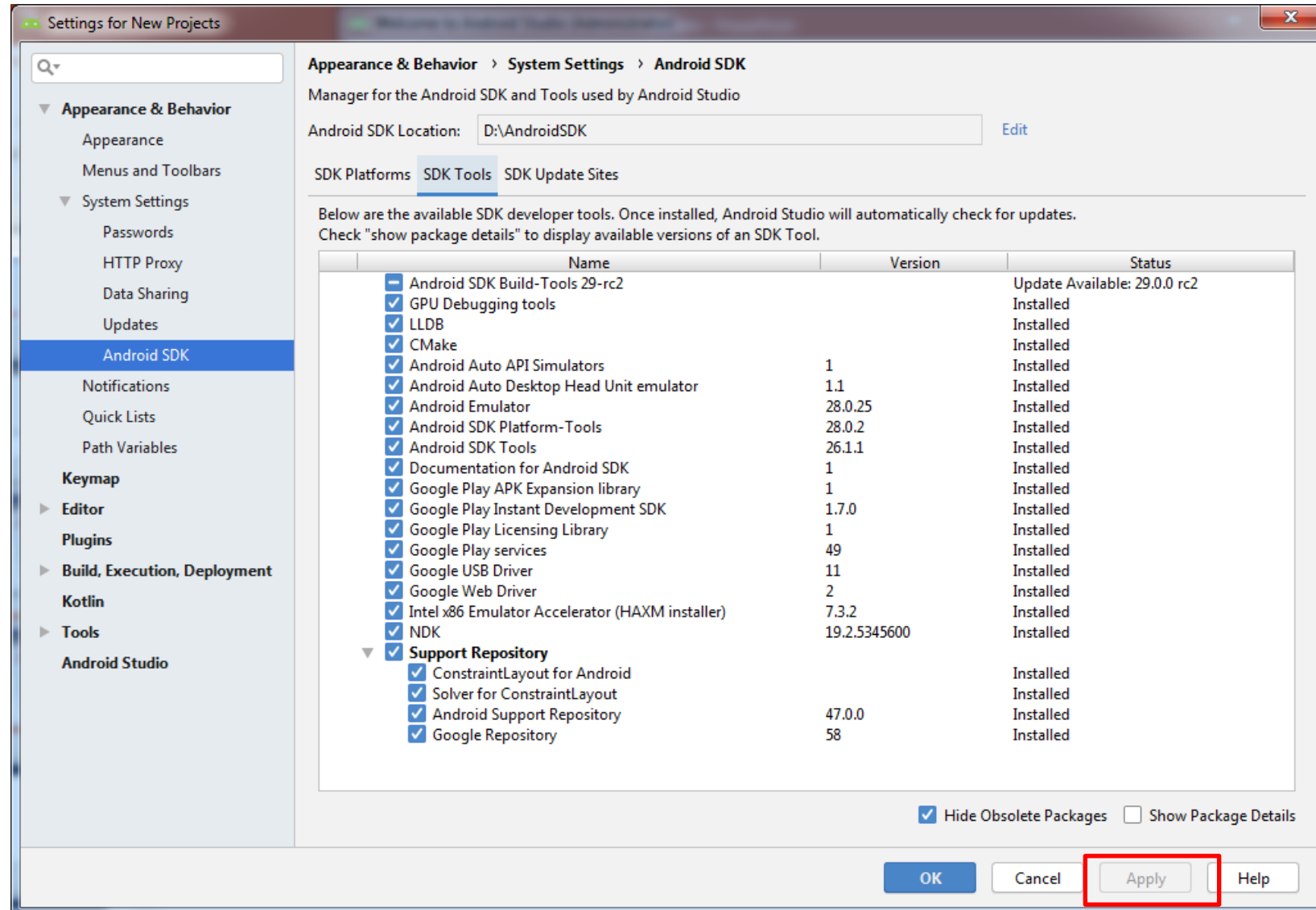


Instalasi Software Development Kit

Pelatihan

4. Centang pada bagian yang dibutuhkan. Kemudian tekan tombol apply untuk melakukan instalasi.

SDK manager akan mendownload dan menginstall kebutuhan tersebut.



Alur pengembangan aplikasi

Dasar Alur Kerja

Pelatihan

Alur kerja untuk mengembangkan aplikasi untuk Android secara konseptual sama dengan platform aplikasi lainnya. Namun, untuk secara efisien membangun aplikasi yang dirancang dengan baik untuk Android, Anda memerlukan beberapa alat khusus..

1. Siapkan ruang kerja anda

Instalasi Android Studio, dan membuat project baru



Dasar Alur Kerja

Pelatihan

2. Menulis code pada aplikasi anda dan menambahkan sumber daya

- ☐ Mulailah mengerjakan aplikasi.
- ☐ Android Studio memiliki berbagai alat yang dapat membantu Anda bekerja lebih cepat, menulis kode kualitas, merancang UI, dan membuat sumber daya untuk berbagai jenis perangkat.

Write

Write code

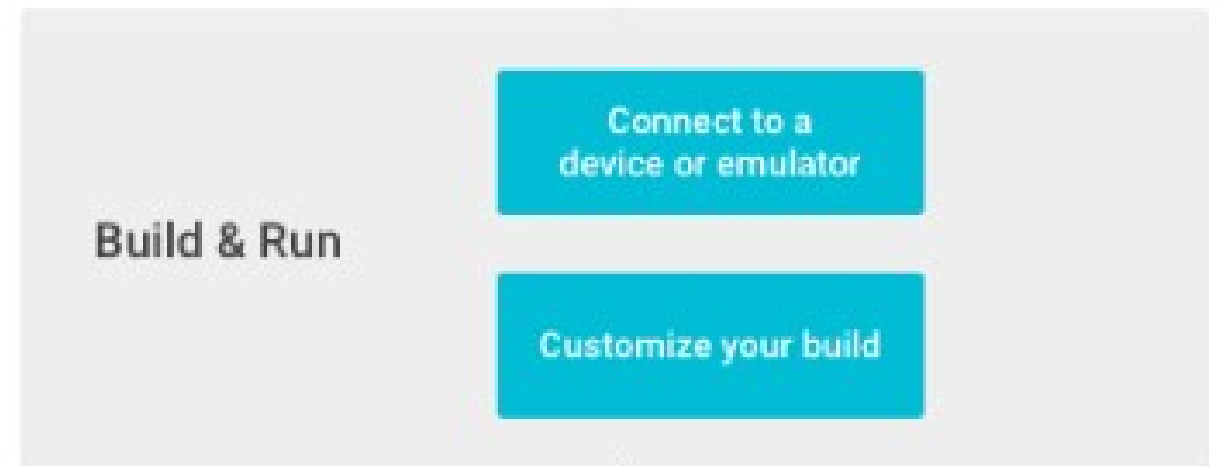
Add assets

Dasar Alur Kerja

Pelatihan

3. Membangun dan menjalankan aplikasi anda

- ☐ Jalankan aplikasi pada smarphone atau emulator.
- ☐ Customize build untuk membentuk ukuran aplikasi yang lebih kecil.

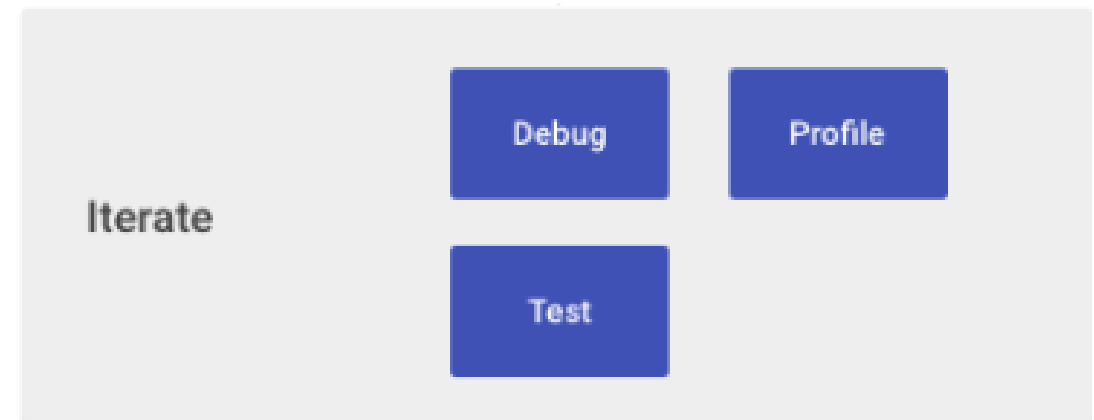


Dasar Alur Kerja

Pelatihan

4. Step perulangan

- ❑ Temukan bug dan tingkatkan kinerja aplikasi.
- ❑ Profile berguna untuk menganalisis kinerja seperti penggunaan memori, lalu lintas jaringan, dampak CPU, dan lainnya.
- ❑ Testing aplikasi anda



Dasar Alur Kerja

Pelatihan

5. Pasarkan aplikasi anda

- ☐ Atur versi aplikasi anda.
- ☐ Buat key dan tanda tangani aplikasi.
- ☐ Ketika update aplikasi, pastikan versi diatur lebih tinggi dan gunakan key yang sama.

Lebih detail pada link berikut ini [https://](https://developer.android.com/studio/publish/index.html)

developer.android.com/studio/publish/index.html

Publish

Version

Sign

Dasar

Dasar

- ❑ Aplikasi android dapat dibuat dengan menggunakan Bahasa java, kotlin dan c++.
- ❑ Android SDK mengkompilasi kode, data dan file sumber daya apa pun ke dalam APK (Android application package), yang merupakan file arsip dengan akhiran .apk.
- ❑ Satu file apk memiliki semua konten dari aplikasi android dan dapat digunakan untuk menginstal aplikasi.

Dasar (Lanjutan)

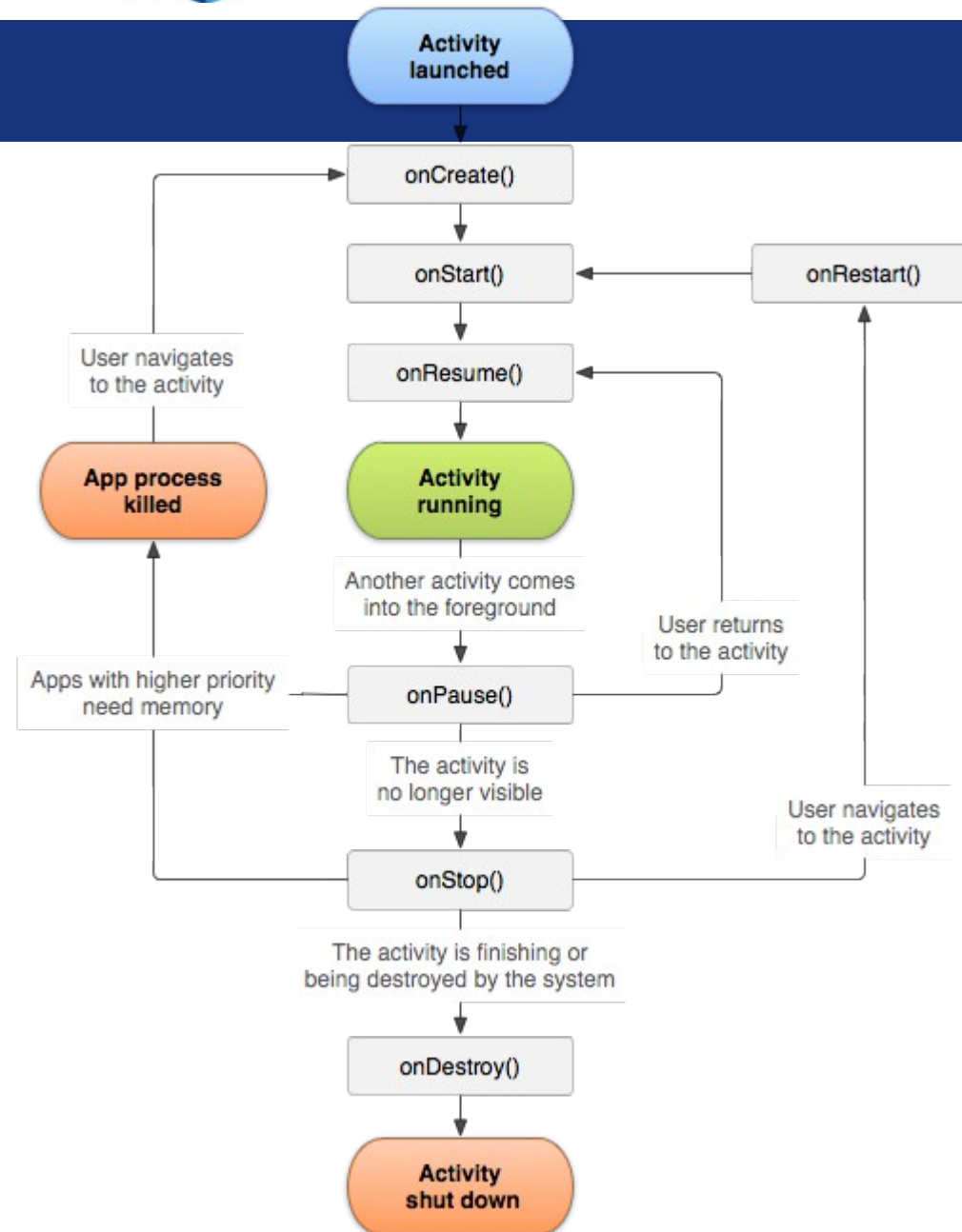
Komponen

Terdapat 4 buah tipe komponen

- ☐ Activities
- ☐ Services
- ☐ Broadcast receivers
- ☐ Content providers

Activities

- ❑ Sebuah activity akan menyajikan user interface (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi.
- ❑ Satu activity biasanya dipakai untuk menampilkan aplikasi atau bertindak sebagai user interface (UI) saat aplikasi diperlihatkan kepada user.
- ❑ Untuk pindah dari satu activity ke activity lain, dapat dilakukan dengan satu even, misalnya klik tombol, memilih opsi atau menggunakan triggers tertentu.



Service

- ❑ Service merupakan sebuah komponen yang digunakan untuk membuat aplikasi tetap menjalankan fungsi pada background.
- ❑ Service tidak memiliki user interface.
- ❑ Contoh dari penggunaan service adalah ketika memainkan music.

Contoh penerapan

- ☐ Transaksi jaringan
- ☐ Memutar musik
- ☐ Melakukan file I/O
- ☐ Berinteraksi dengan penyedia materi

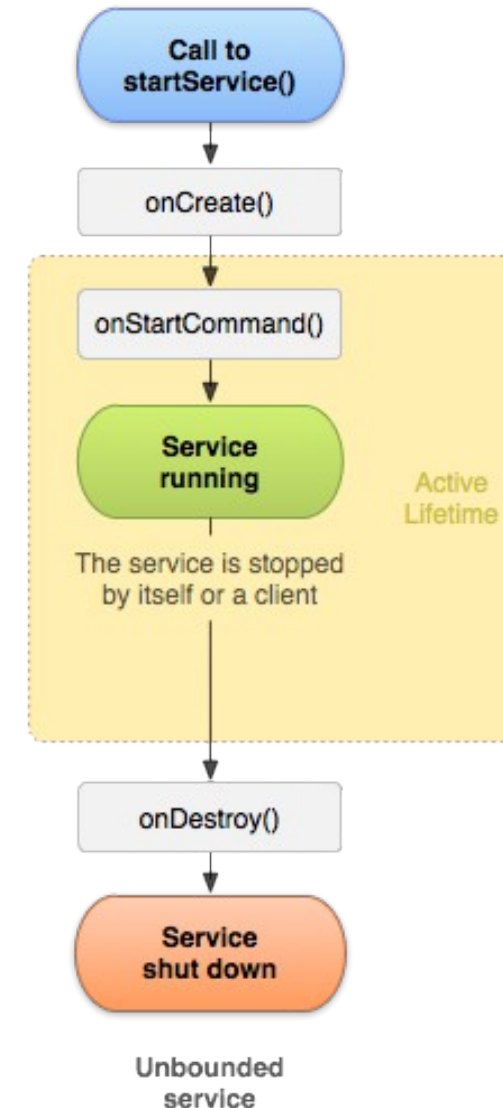
Karakteristik layanan

- ☐ Dimulai dengan Intent
- ☐ Bisa tetap berjalan ketika pengguna beralih aplikasi
- ☐ Daur hidup—yang harus Anda kelola
- ☐ Aplikasi lain bisa menggunakan layanan—mengelola izin
- ☐ Berjalan di thread utama dari proses hosting-nya

Service: tidak terikat (unbound service)

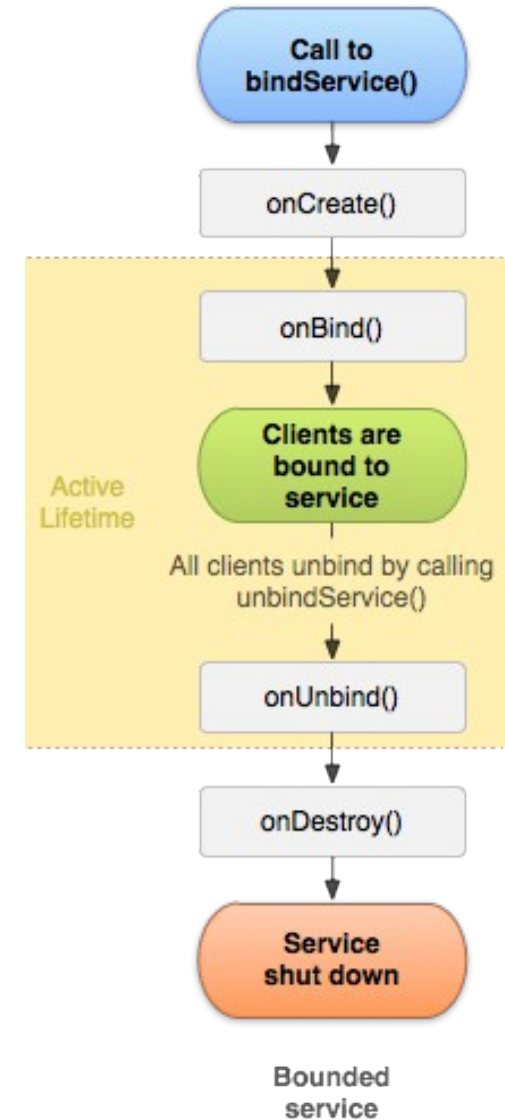
- ☐ Dimulai dengan Intent
- ☐ Bisa tetap berjalan ketika pengguna beralih aplikasi
- ☐ Berjalan di thread utama dari proses hosting-nya
- ☐ Daur hidup yang harus Anda kelola, jika tidak dihentikan, akan terus berjalan.

- Dimulai dengan startService()
- Berjalan tanpa batas waktu sampai berhenti sendiri
- Biasanya tidak memperbarui UI



Service: terikat (bound service)

- ❑ Menawarkan antarmuka klien-server yang memungkinkan komponen untuk berinteraksi dengan service
- ❑ Klien mengirimkan permintaan dan mendapatkan hasil
- ❑ Dimulai dengan `bindService()`
- ❑ Berakhir ketika semua klien melepaskan kaitan

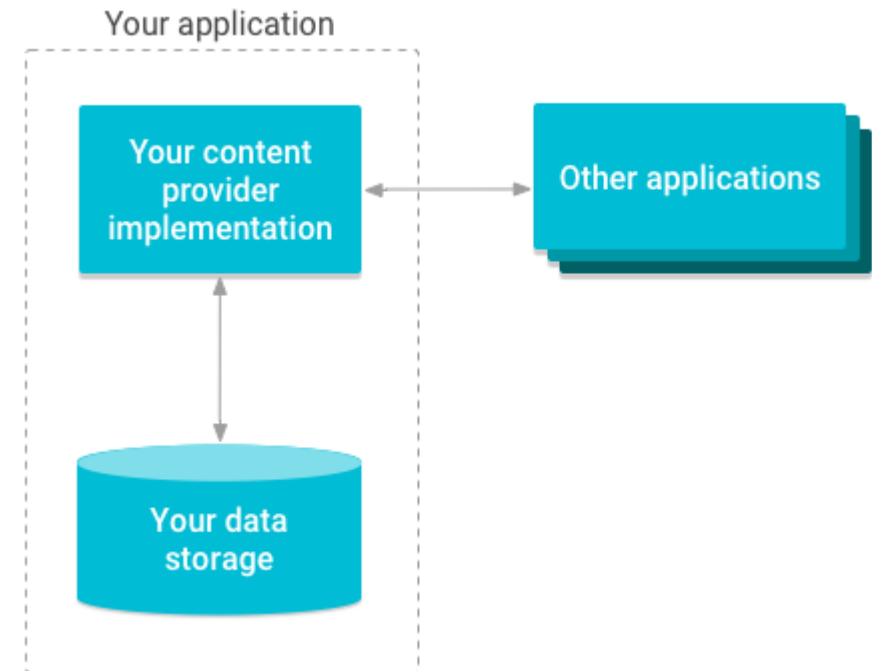


Broadcast receivers

- ☐ Broadcast receiver berfungsi menerima dan berekasi untuk menyampaikan notifikasi.
- ☐ Contohnya adalah menampilkan notifikasi zona waktu berubah, baterai lemah, atau
- ☐ pengubahan referensi bahasa yang digunakan.

Content providers

- ❑ Komponen pengambil data yang diminta aplikasi dari repositori
- ❑ Aplikasi ini tidak perlu mengetahui di mana atau bagaimana data disimpan, diformat, atau diakses



JAVA (Konsep variabel, Kondisi dan Perulangan)

Dasar Java Android

Pelatihan

Apa itu Java?

Java adalah bahasa pemrograman yang terkenal, dibuat pada tahun 1995. Dimiliki oleh Oracle.

Digunakan untuk :

1. Aplikasi Mobile (spesial Android apps)
2. Aplikasi Desktop
3. Aplikasi Web
4. Web Server dan Application server
5. Game
6. Koneksi Basisdata
7. Dan lain lain

Dasar Java Android

Pelatihan

Mengapa Menggunakan Java?

1. Java berfungsi pada berbagai platform (Windows, Mac, Linux, Raspberry Pi, dll.)
2. Salah satu bahasa pemrograman paling populer di dunia
3. Mudah dipelajari dan mudah digunakan
4. Open-source dan gratis
5. Aman, cepat dan kuat
6. Memiliki dukungan komunitas yang sangat besar (puluhan juta pengembang)

Sintaks Java

Pelatihan

Sintaks Java

kode berikut untuk mencetak "Hello World" ke layar:

```
public class MyClass {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```


Java Comment

Pelatihan

- **Java Comments**

1. Komentar dapat digunakan untuk menjelaskan kode Java, dan membuatnya lebih mudah dibaca. Juga dapat digunakan untuk mencegah eksekusi ketika menguji kode alternatif.
2. Komentar baris tunggal dimulai dengan dua garis miring (//).
3. Teks apa pun antara // dan akhir baris diabaikan oleh Java (tidak akan dieksekusi).

- **Java Multi-line Comments**

1. Komentar banyak baris dimulai dengan / * dan diakhiri dengan * /.
2. Teks apa pun antara / * dan * / akan diabaikan oleh Java.

```
// This is a comment
```

```
/* The code below will print the words Hello World  
to the screen, and it is amazing */  
System.out.println("Hello World");
```

Konsep Variabel

Pelatihan

- **Variabel Java**

Variabel adalah wadah untuk menyimpan nilai data.

Ada berbagai jenis variabel, misalnya:

1. String - menyimpan teks, seperti "Halo". Nilai string ditandai oleh tanda kutip ganda
2. int - Integer (bilangan bulat), tanpa desimal, seperti 123 atau -123
3. float - menyimpan angka floating point, dengan desimal, seperti 19,99 atau -19,99
4. char - menyimpan karakter tunggal, seperti 'a' atau 'B'. Nilai-nilai Char ditandai oleh tanda kutip tunggal
5. boolean - menyimpan nilai dengan dua status: benar atau salah

Konsep Variabel

Pelatihan

- **Mendeklarasikan (Membuat) Variabel**

1. Untuk membuat variabel, Anda harus menentukan tipe dan memberinya nilai:

Sintaksis: *type variable = value;*

2. Untuk membuat variabel yang harus menyimpan teks, lihat contoh berikut:
Contoh Buat variabel bernama nama tipe String dan berikan nilai "John":

```
String name = "John";  
System.out.println(name);
```

Contoh Lain :

```
int myNum = 5;  
float myFloatNum = 5.99f;  
char myLetter = 'D';  
boolean myBool = true;  
String myText = "Hello"
```

Konsep Variabel

Pelatihan

- **Menampilkan Variabel**

1. Metode `println ()` sering digunakan untuk menampilkan variabel.
2. Untuk menggabungkan teks dan variabel, gunakan karakter `+`:

```
String name = "John";  
System.out.println("Hello " + name);
```

Konsep Variabel

Pelatihan

- Java Identifiers
 1. Semua variabel Java harus diidentifikasi dengan nama unik.
 2. Nama-nama unik ini disebut **identifiers**.
 3. **identifiers** dapat berupa nama pendek (seperti x dan y) atau nama yang lebih deskriptif (usia, jumlah, total Volume).

Aturan umum untuk membuat nama untuk variabel (unique identifiers) adalah:

4. Nama dapat berisi huruf, angka, garis bawah, dan tanda dolar
5. Nama harus dimulai dengan huruf
6. Nama juga dapat dimulai dengan \$ dan _ (tetapi tidak dalam tutorial ini)
7. Case-sensitive ("myVar" dan "myvar" adalah variabel yang berbeda)
8. Nama harus dimulai dengan huruf kecil dan tidak boleh mengandung spasi
9. Kata-kata yang dicadangkan (seperti kata kunci Java, seperti int atau String) tidak dapat digunakan sebagai nama

Konsep Variabel

Pelatihan

- Tipe Data Java

Jenis Tipe data dibagi menjadi dua kelompok:

1. Tipe data primitif - termasuk byte, short, int, long, float, double, boolean dan char
2. Tipe data non-primitif - seperti String, Array, dan Class

- Jenis Tipe data Primitif

1. Tipe data primitif menentukan ukuran dan jenis nilai variabel, tidak memiliki metode tambahan.
2. Ada delapan tipe data primitif di Jawa:

Konsep Variabel

Pelatihan

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers from $3.4e-038$ to $3.4e+038$. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers from $1.7e-308$ to $1.7e+038$. Sufficient for storing 15 decimal digits
boolean	1 byte	Stores true or false values
char	2 bytes	Stores a single character/letter

Konsep Variabel

Pelatihan

Contoh :

```
byte myNum = 100;  
short myNum = 5000;  
int myNum = 100000;  
long myNum = 1500000000000L;  
float myNum = 5.75f;  
double myNum = 19.99d;  
boolean isJavaFun = true;  
boolean isFishTasty = false;  
char myGrade = 'B';  
String greeting = "Hello World";
```


Konsep Variabel

Pelatihan

- Tipe Data Non-Primitif

Tipe data non-primitif disebut tipe referensi karena merujuk pada objek.

Perbedaan utama antara tipe data primitif dan non-primitif adalah:

1. Tipe primitif sudah ditentukan sebelumnya (sudah ditentukan) di Java. Tipe non-primitif dibuat oleh programmer dan tidak didefinisikan oleh Java (kecuali untuk String).
2. Tipe non-primitif dapat digunakan untuk memanggil metode untuk melakukan operasi tertentu, sedangkan tipe primitif tidak bisa.
3. Tipe primitif selalu memiliki nilai, sedangkan tipe non-primitif bisa menjadi nol.
4. Tipe primitif dimulai dengan huruf kecil, sedangkan tipe non-primitif dimulai dengan huruf besar.
5. Ukuran tipe primitif tergantung pada tipe data, sedangkan tipe non-primitif memiliki semua ukuran yang sama.
6. Contoh tipe non-primitif adalah String, Array, Class, Interface, dll.

Konsep Variabel

Pelatihan

- Keyword final

Pengubah non-akses yang digunakan untuk kelas, atribut, dan metode, yang membuatnya tidak dapat diubah (tidak mungkin diwariskan atau ditimpa)

Final pada variable

```
class Lingkaran {  
    final double PI = 3.14;  
}
```

Final Pada Method

```
class Kampus {  
    final void tampil() {  
        System.out.println("I LOVE JAVA")  
    }  
}
```

Final pada class

```
class Kampus {  
    final void tampil() {  
        System.out.println("I LOVE JAVA")  
    }  
}
```

Konsep Variabel

Pelatihan

- Operator Java

Operator Aritmatika digunakan untuk melakukan operasi matematika umum.

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value from another	x / y
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$++x$
--	Decrement	Decreases the value of a variable by 1	$--x$

Kondisi

Pelatihan

- Kondisi Java dan Pernyataan `if`

Java mendukung kondisi logis yang biasa dari matematika:

1. Kurang dari: `a < b`
2. Kurang dari atau sama dengan: `a <= b`
3. Lebih besar dari: `a > b`
4. Lebih besar atau sama dengan: `a >= b`
5. Sama dengan `a == b`
6. Tidak sama dengan: `a != B`

Anda dapat menggunakan kondisi ini untuk melakukan tindakan berbeda untuk keputusan yang berbeda. Java memiliki pernyataan kondisional berikut:

7. Gunakan `if` untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan benar
8. Gunakan `else` untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama salah
9. Gunakan `else if` menentukan kondisi baru untuk diuji, jika kondisi pertama salah
10. Gunakan `switch` menentukan banyak blok kode alternatif yang akan dieksekusi

Kondisi

Pelatihan

- Pernyataan if

```
int x = 20;  
int y = 18;  
if (x > y) {  
    System.out.println("x is greater than y");  
}
```

- Pernyataan else

```
int time = 20;  
if (time < 18) {  
    System.out.println("Good day.");  
} else {  
    System.out.println("Good evening.");  
}
```

Kondisi

Pelatihan

- Pernyataan else if

```
int time = 22;  
if (time < 10) {  
    System.out.println("Good morning.");  
} else if (time < 20) {  
    System.out.println("Good day.");  
} else {  
    System.out.println("Good evening.");  
}
```

Kondisi

Pelatihan

- Pernyataan Switch Java

Gunakan pernyataan `switch` untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

1. Ekspresi `switch` dievaluasi sekali.
2. Nilai ekspresi dibandingkan dengan nilai setiap kasus.
3. Jika ada kecocokan, blok kode terkait dijalankan.
4. `break` dan kata kunci `default` adalah opsional.

Kata kunci `break`

5. Ketika Java mencapai kata kunci `break`, maka keluar dari blok `switch`.
6. Ini akan menghentikan pelaksanaan lebih banyak kode dan pengujian kasus di dalam blok.
7. Ketika kecocokan ditemukan, dan pekerjaan selesai, saatnya `break`(dihentikan). Tidak perlu pengujian lebih lanjut.
8. `Break` dapat menghemat banyak waktu eksekusi karena "mengabaikan" eksekusi semua kode lainnya di blok `switch`.

- Kata Kunci `default`

Kata kunci `default` untuk dijalankan jika tidak ada kecocokan .

Kondisi

Pelatihan

Contoh :

```
int day = 4;
switch (day) {
    case 6:
        System.out.println("Today is Saturday");
        break;
    case 7:
        System.out.println("Today is Sunday");
        break;
    default:
        System.out.println("Looking forward to the
Weekend");
}
```


Perulangan

Pelatihan

- Perulangan `while`

Perulangan `while` dikerjakan selama kondisi yang ditentukan benar.

```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

- Perulangan `Do/While`

1. Do / while loop adalah varian dari perulangan `while`.
2. Perulangan ini akan mengeksekusi blok kode sekali, sebelum memeriksa apakah kondisinya benar, akan mengulang loop selama kondisinya benar.

```
int i = 0;
do {
    System.out.println(i);
    i++;
}
while (i < 5);
```

Perulangan

Pelatihan

- Perulangan For

```
for (int i = 0; i <= 10; i = i + 2) {  
    System.out.println(i);  
}
```

1. Pernyataan 1 dieksekusi (satu kali) sebelum eksekusi blok kode. Menetapkan variabel sebelum perulangan dimulai (`int i = 0`).
2. Pernyataan 2 mendefinisikan kondisi untuk mengeksekusi blok kode. Mendefinisikan kondisi untuk menjalankan perulangan (harus kurang dari sama dengan 10). Jika kondisinya benar, perulangan akan memulai lagi, jika itu salah, perulangan akan berakhir.
3. Pernyataan 3 dieksekusi (setiap kali) setelah blok kode dieksekusi. Meningkatkan nilai (`i ++`) setiap kali blok kode dalam perulangan telah dieksekusi.

Perulangan

Pelatihan

- Perulangan For-Each

Perulangan **for-each**, digunakan secara eksklusif untuk mengulang elemen-elemen dalam array

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
    System.out.println(i);
}
```

- Java break pada perulangan for

```
for (int i = 0; i < 10; i++) {
    if (i == 4) {
        break;
    }
    System.out.println(i);
}
```

Konsep Layout

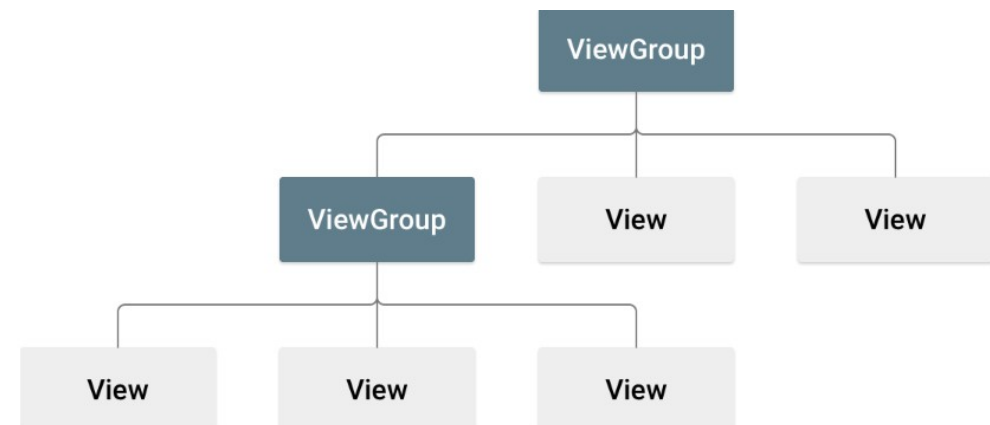
Dasar Layout

Pelatihan

- Layout

Layout mendefinisikan struktur visual untuk antarmuka pengguna, seperti UI sebuah aktivitas atau widget aplikasi. Anda dapat mendeklarasikan layout dengan dua cara:

- 1. Deklarasikan elemen UI dalam XML.** Android menyediakan sebuah kosakata XML sederhana yang sesuai dengan kelas dan subkelas View, seperti halnya untuk widget dan layout.
- 2. Buat instance elemen layout saat runtime.** Dalam suatu aplikasi bisa membuat objek View dan ViewGroup (dan memanipulasi propertinya) lewat program.



Deklarasikan elemen UI dalam XML

Pelatihan

Keuntungan mendeklarasikan UI dalam XML

1. Memungkinkan Anda memisahkan penampilan aplikasi dari kode yang mengontrol perilakunya dengan lebih baik.
2. Keterangan UI Anda bersifat eksternal bagi kode aplikasi Anda, yang berarti bahwa Anda bisa memodifikasi atau menyesuaikan tanpa harus memodifikasi dan mengompilasi ulang kode sumber.
3. Misalnya, Anda bisa membuat layout XML untuk berbagai orientasi layar, berbagai ukuran layar perangkat, dan berbagai bahasa

Layout - XML

Pelatihan

File --> New --> Project...

Choose Android --> Android Application Project

Edit res/layout/activity_main.xml, and replace everything with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/
  android"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical" >
</LinearLayout>
```

Layout - XML

Pelatihan

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```


Layout - XML

Pelatihan

- **Compile aplikasi**
 1. Masing-masing file layout XML akan dikompilasi dalam sebuah sumber daya View.
 2. Harus memuat sumber daya layout dari kode aplikasi, dalam implementasi callback Activity.onCreate().
 3. Lakukan dengan memanggil setContentView(), dengan meneruskan acuan ke sumber daya layout berupa:
`R.layout.layout_file_name`.Misalnya, jika XML layout Anda disimpan sebagai `main_layout.xml`, Anda akan memuatnya untuk Activity seperti ini:

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main_layout);  
}
```

Layout - XML

Pelatihan

- Attributes

1. Setiap objek View dan ViewGroup mendukung variasi atribut XML-nya sendiri.
2. Sebagian atribut bersifat spesifik untuk objek View (misalnya, TextView mendukung atribut **textSize**).
3. Sebagian atribut bersifat umum untuk semua objek View, karena diwarisi dari kelas Root View (seperti atribut **id**).
4. Atribut lain dianggap sebagai "parameter layout" yaitu atribut yang menjelaskan orientasi layout tertentu dari objek View, seperti yang didefinisikan oleh objek ViewGroup induk dari objek itu.

Layout - XML

Pelatihan

- ID

1. Objek View apa saja dapat memiliki ID integer yang dikaitkan dengannya, untuk mengidentifikasi secara unik `View` dalam pohon.
2. Bila aplikasi dikompilasi, ID ini akan diacu sebagai integer, namun ID biasanya ditetapkan dalam file XML layout sebagai string, dalam atribut `id`. Ini atribut XML yang umum untuk semua objek `View` (yang didefinisikan oleh kelas `View`) dan Anda akan sering sekali menggunakannya

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

Layout - XML

Pelatihan

- **Parameter Layout**

Atribut layout XML bernama *layout_something* mendefinisikan parameter layout View yang cocok untuk ViewGroup tempatnya berada.

Semua grup tampilan berisi lebar dan tinggi (*layout_width* dan *layout_height*), dan masing-masing tampilan harus mendefinisikannya. Banyak *LayoutParams* yang juga menyertakan margin dan border opsional.

Salah satu konstanta ini untuk mengatur lebar atau tinggi:

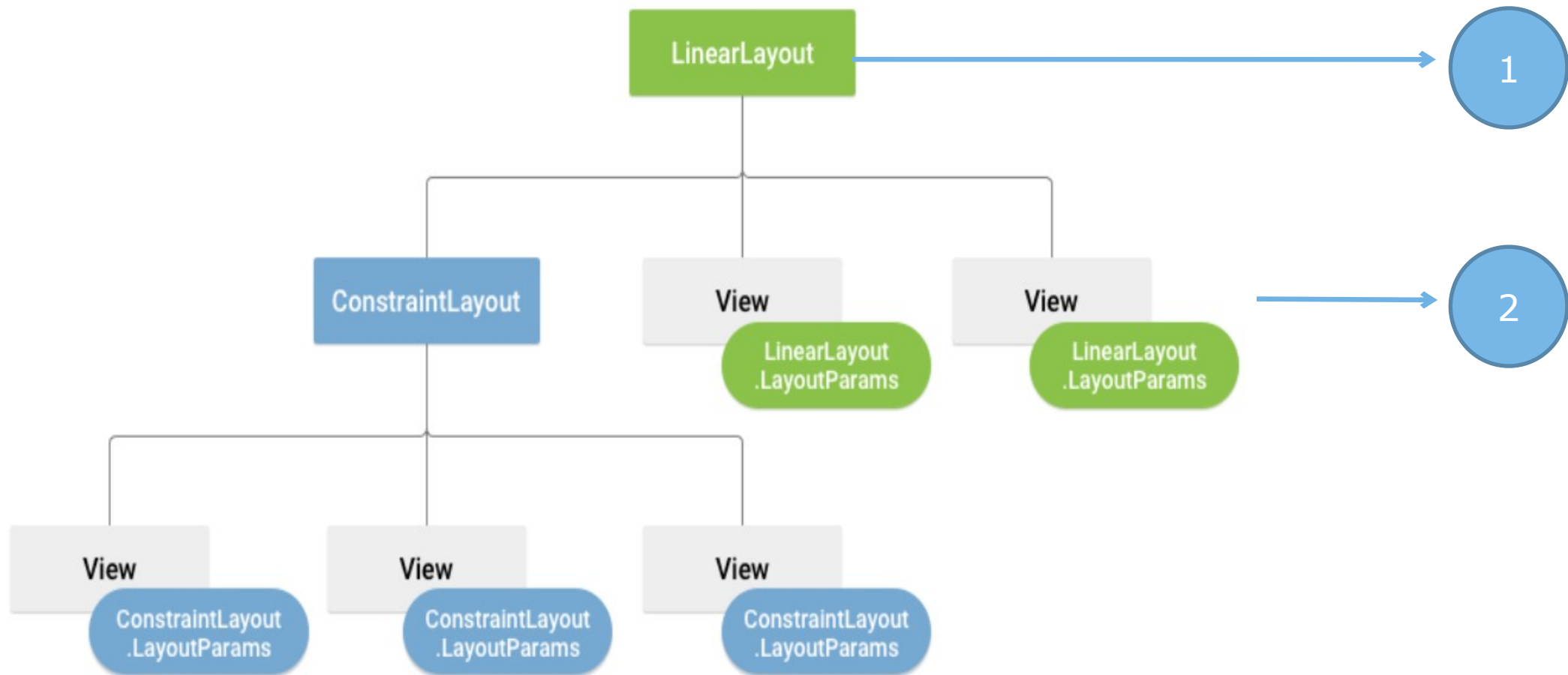
1. ***wrap_content*** memberi tahu tampilan agar menyesuaikan sendiri ukurannya dengan dimensi yang dibutuhkan oleh materinya.
2. ***match_parent*** memberi tahu tampilan agar menjadi sebesar yang akan diperbolehkan oleh kelompok tampilan induknya.

Layout - Struktur

Pelatihan

- **Parameter Layout**

1. Grup tampilan *root*.
2. Rangkaian tampilan anak dan grup tampilan pertama yang induknya

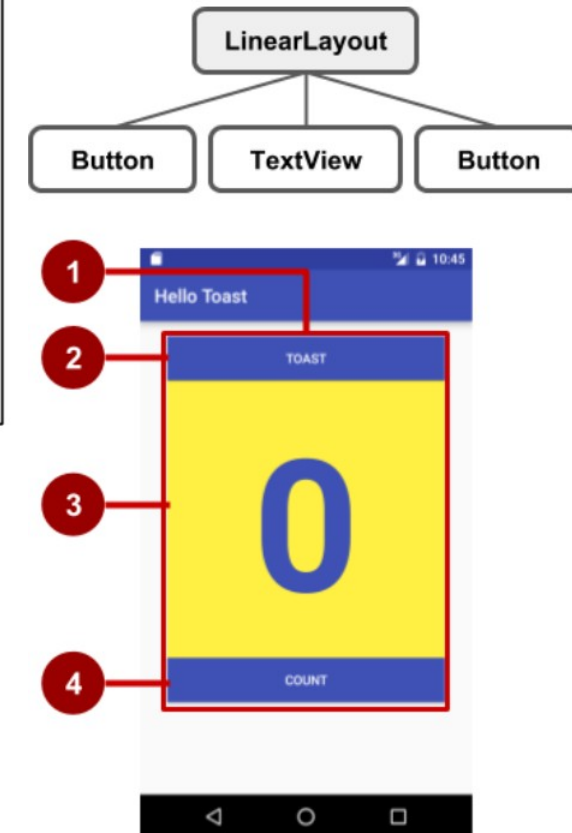
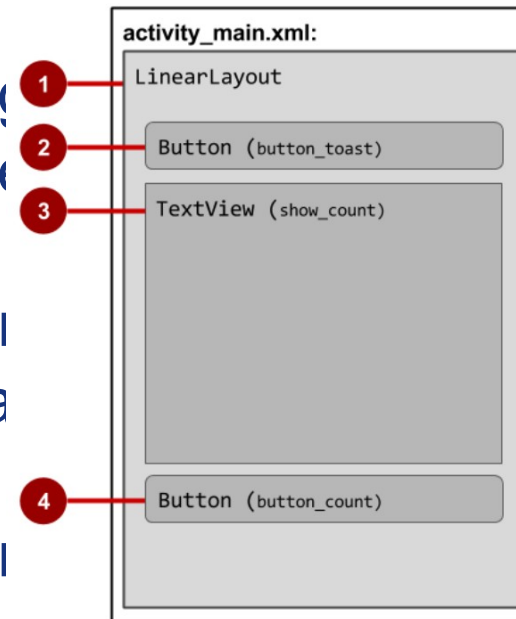


LinearLayout

Pelatihan

Linear Layout

1. Layout akar LinearLayout, yang berisi semua tampilan anak, disetel ke orientasi vertikal.
2. Button (button_toast) tampilan anak. Sebagai tampilan anak pertama muncul di bagian atas di layout linear.
3. TextView (show_count) tampilan anak. Sebagai tampilan anak kedua, muncul di bawah tampilan anak pertama di layout linear.
4. Button (button_count) tampilan anak. Sebagai tampilan anak ketiga, muncul di bawah tampilan anak kedua di layout linear.



Linearlayout

Pelatihan

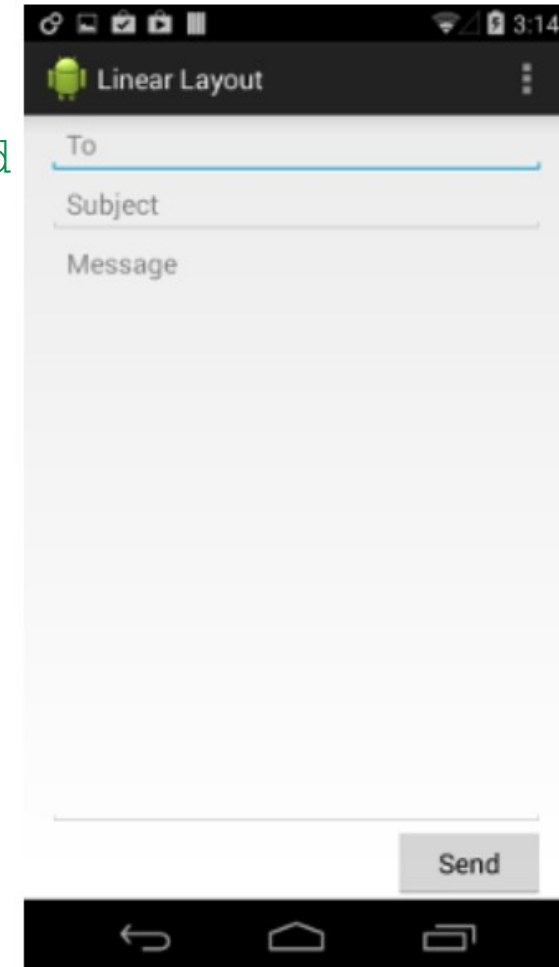
- Linear Layout
 1. LinearLayout adalah sekelompok tampilan yang menyejajarkan semua anak dalam satu arah, secara vertikal atau horizontal.
 2. Menetapkan arah layout dengan atribut `android:orientation`.



Layout-Linier Layout

Pelatihan

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
```



Layout-Linier Layout

Pelatihan

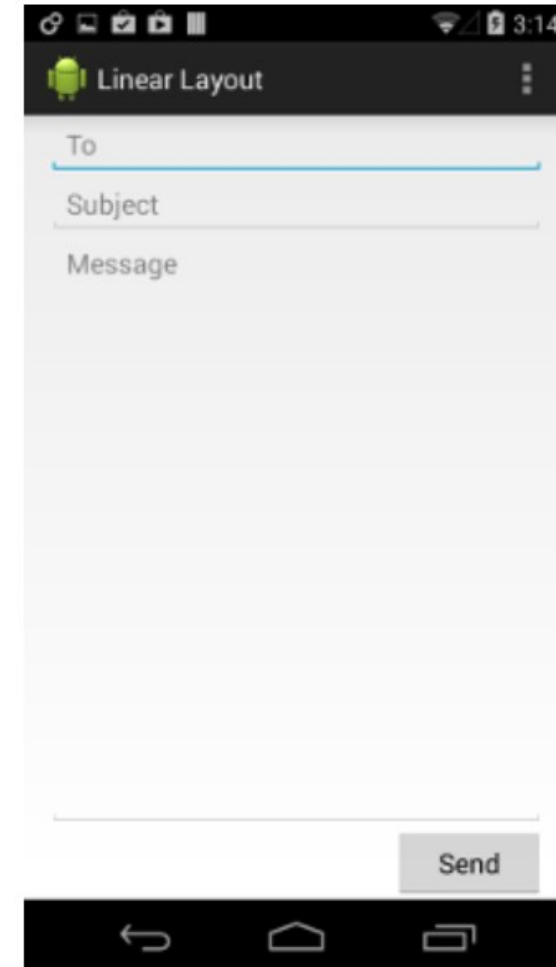
<EditText

```
android:layout_width="match_parent"  
android:layout_height="0dp"  
android:layout_weight="1"  
android:gravity="top"  
android:hint="@string/message" />
```

<Button

```
android:layout_width="100dp"  
android:layout_height="wrap_content"  
android:layout_gravity="right"  
android:text="@string/send" />
```

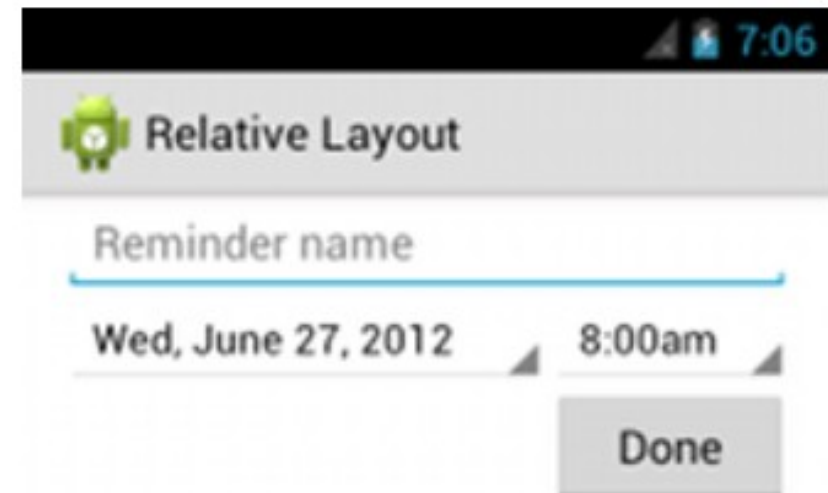
</LinearLayout>



Relatif Layout

Pelatihan

- Relatif Layout
 1. Grup tampilan anak yang setiap tampilannya diposisikan dan disejajarkan relatif terhadap tampilan dalam grup tampilan.
 2. Dengan kata lain, posisi tampilan anak bisa dijelaskan dalam hubungan satu sama lain atau dengan grup tampilan induk.



Relatif Layout

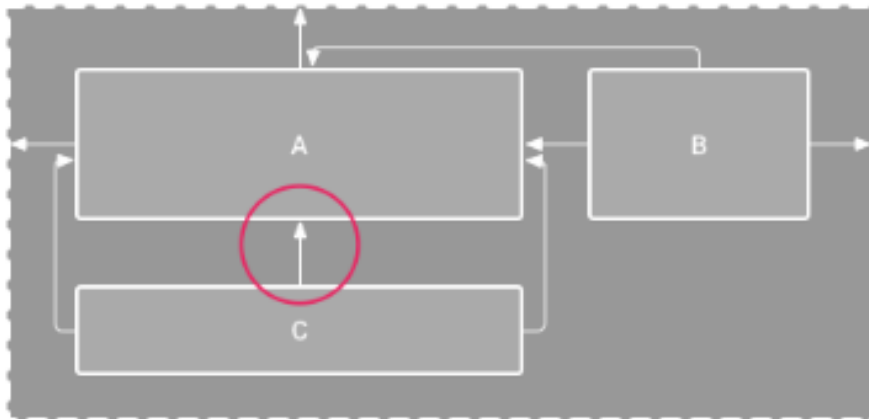
Pelatihan

```
<?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:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" /
    >
    <Spinner
        android:id="@id/times"
        android:layout_width="96dp"
```

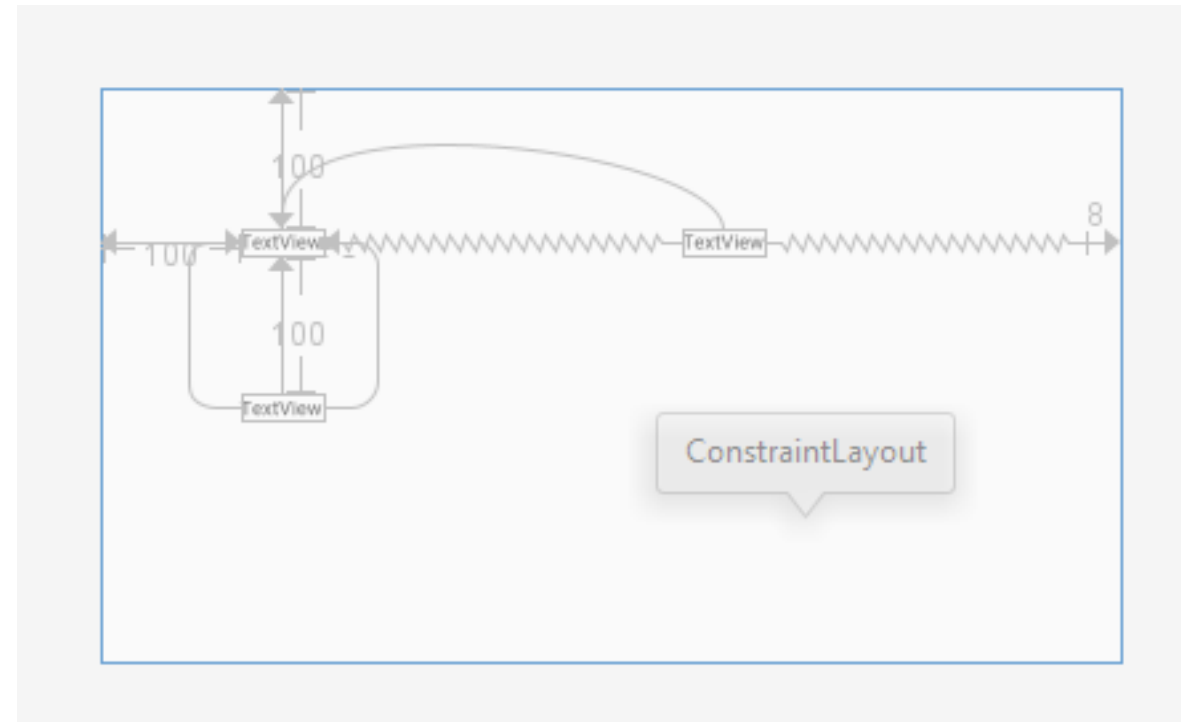
Constraint Layout

Pelatihan

1. Bangun tampilan yang responsive.
2. Mirip seperti layout relatif, dimana tampilan bergantung dengan view lain atau parent layoutnya.
3. Didukung oleh android studio, gunakan drag and drop daripada mengedit kode xml.



Contoh constraint layout



Constraint Layout

Pelatihan

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
    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">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="100dp"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="100dp"
        android:text="TextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="TextView"
```

KESIMPULAN

1. Peserta mengetahui Kategori dan Jenis Bahasa Pemrograman
2. Peserta mengetahui cara mengatur Ruang Kerja
3. Peserta mengetahui Dasar Alur Pembuatan Software
4. Peserta mengetahui Konsep Variabel dan Konstanta
5. Peserta mengetahui Struktur Kondisi dan Perulangan
6. Peserta mengetahui Konsep Layout
7. Peserta bisa membuat sebuah aplikasi sederhana

Referensi

1. THE WORLD'S LARGEST WEB DEVELOPER SITE, Java Tutorial, di akses 27/04/2019 <https://www.w3schools.com/java/default.asp>
2. Developer Google-Android developer guides (Doc), User Interface & Navigation, di akses 27/04/2019 <https://developer.android.com/guide/topics/ui>,
3. Tim Pelatihan Developer Google, Kursus Dasar-Dasar Developer Android-Konsep, Creative Commons Attribution-NonCommercial 4.0 International License, Desember 2016, di akses 27/04/2019, https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/idn/Unit%201/12_c_layouts,_views_and_resources.html

Tim Penyusun:

- Alif Akbar Fitrawan, S.Pd, M. Kom (Politeknik Negeri Banyuwangi);
- Anwar, S.Si, MCs. (Politeknik Negeri Lhokseumawe);
- Eddo Fajar Nugroho (BPPTIK Cikarang);
- Eddy Tungadi, S.T., M.T. (Politeknik Negeri Ujung Pandang);
- Fitri Wibowo (Politeknik Negeri Pontianak);
- Ghifari Munawar (Politeknik Negeri Bandung);
- Hetty Meileni, S.Kom., M.T. (Politeknik Negeri Sriwijaya) ;
- I Wayan Candra Winetra, S.Kom., M.Kom (Politeknik Negeri Bali) ;
- Irkham Huda (Vokasi UGM) ;
- Josseano Amakora Koli Parera, S.Kom., M.T. (Politeknik Negeri Ambon) ;
- I Komang Sugiarta, S.Kom., MMSI (Universitas Gunadarma) ;
- Lucia Sri Istiyowati, M.Kom (Institut Perbanas) ;
- Maksy Sendiang, ST, MIT (Politeknik Negeri Manado) ;
- Medi Noviana (Universitas Gunadarma) ;
- Muhammad Nashrullah (Politeknik Negeri Batam) ;
- Nat. I Made Wiryana, S.Si., S.Kom., M.Sc. (Universitas Gunadarma) ;
- Rika Idmayanti, ST, M.Kom (Politeknik Negeri Padang) ;
- Rizky Yuniar Hakkun (Politeknik Elektronik Negeri Surabaya) ;
- Robinson A.Wadu, ST., MT (Politeknik Negeri Kupang) ;
- Roslina. M.IT (Politeknik Negeri Medan) ;
- Sukamto, SKom., MT. (Politeknik Negeri Semarang) ;
- Syamsi Dwi Cahya, M.Kom. (Politeknik Negeri Jakarta) ;
- Syamsul Arifin, S.Kom, M.Cs (Politeknik Negeri Jember) ;
- Usmanudin (Universitas Gunadarma) ;
- Wandy Alifha Saputra (Politeknik Negeri Banjarmasin) ;