Mobile Program

Pertemuan 12

Event

- Event merupakan interaksi pengguna dengan Android
- Di Android, ada lebih dari satu cara untuk mencegat Event dari interaksi pengguna dengan aplikasi.
- Menangkap Event dari objek View spesifik yang berinteraksi dengan pengguna.

Event Listener

- Event Listener adalah antarmuka di kelas View yang berisi metode *callback*.

- Metode-metode ini akan dipanggil oleh kerangka kerja Android ketika View yang memiliki Listener dipicu oleh interaksi pengguna di UI.

CallBack Event

- **onClick ()**
- Dari View.OnClickListener. Metode akan dipicu ketika pengguna menyentuh item

- **onLongClick ()**
- Dari View.OnLongClickListener. Metode akan dipicu ketika pengguna menyentuh tahan item

Cont'd

- **onFocusChange ()**
- Dari View.OnFocusChangeListener. Metode akan dipicu ketika pengguna menavigasi ke atau jauh dari item

- **onKey ()**
- Dari View.OnKeyListener. Metode akan dipicu ketika pengguna fokus pada item dan menekan atau melepaskan kunci perangkat keras pada perangkat.

Cont'd

- **onTouch ()**
- Dari View.OnTouchListener. Metode akan dipicu ketika pengguna melakukan tindakan yang memenuhi syarat sebagai event sentuh

Implementasi Listener

- Event Registration adalah proses di mana Event Handler yang terdaftar dengan Event Listener sehingga handler akan dipanggil ketika Event Listener memicu event.
- Terdapat cara pemasangan Listener:
  - Kelas aktivitas mengimplementasikan antarmuka Listener.
  - Menggunakan file Layout activity_main.xml untuk menentukan event handler secara langsung.

Implementasi Listener #1

```
protected void onCreate(savedValues: Bundle) {
    val button: Button = findViewById(R.id.corky)


    button.setOnClickListener { view ->
        // do something when the button is clicked
    }
}
```

Implementasi Listener #2

```
class ExampleActivity : Activity(), OnClickListener {
    protected fun onCreate(savedValues: Bundle) {
        val button: Button = findViewById(R.id.corky)
        button.setOnClickListener(this)
    }
    fun onClick(v: View) {
        // do something when the button is clicked
    }
}
```

Implementasi #3

```
<Button android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button 1"
    android:onClick="HandleClick"/>
```

Metode Input

- Ketika seorang pengguna menavigasi antarmuka pengguna dengan tombol arah atau trackball, perlu untuk fokus pada item sehingga pengguna dapat memberikan input.

- Namun, jika perangkat memiliki kemampuan sentuh, dan pengguna berinteraksi dengan antarmuka dengan menyentuhnya, maka tidak perlu lagi fokus ke View tertentu

Touch Gestures

- Android menyediakan berbagai API untuk membantu membuat dan mendeteksi gerakan.
- Meskipun aplikasi tidak harus bergantung pada gerakan sentuh untuk perilaku dasar, tetapi dapat menambahkan kegunaan dan daya tariknya.

Mendeteksi Gerakan

- "Gerakan sentuh" terjadi ketika pengguna menempatkan satu atau lebih jari pada layar sentuh, dan aplikasi mengartikan pola sentuhan itu sebagai gerakan tertentu.
- Ada dua fase yang sesuai untuk deteksi gerakan:
  - Kumpulkan data tentang Event Touch.
  - Tafsirkan data untuk melihat apakah memenuhi kriteria untuk setiap gerakan yang didukung aplikasi

Mengumpulkan Data

- Ketika pengguna menempatkan satu atau lebih jari pada layar, ini memicu callback onTouchEvent () pada View yang menerima peristiwa sentuhan.

- Untuk setiap urutan peristiwa sentuh (posisi, tekanan, ukuran, penambahan jari lain, dll.) Yang akhirnya diidentifikasi sebagai gerakan, onTouchEvent () ditembakkan beberapa kali.

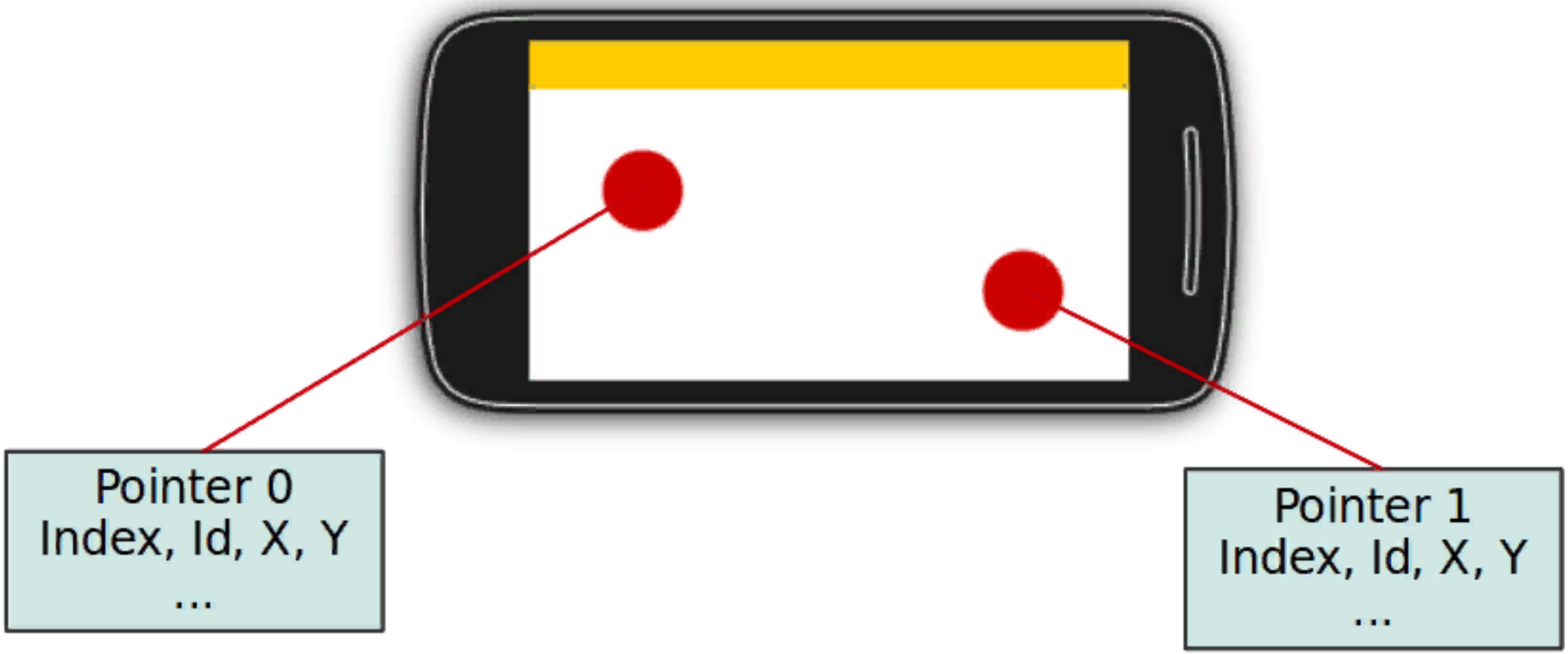
Deteksi Gerakan

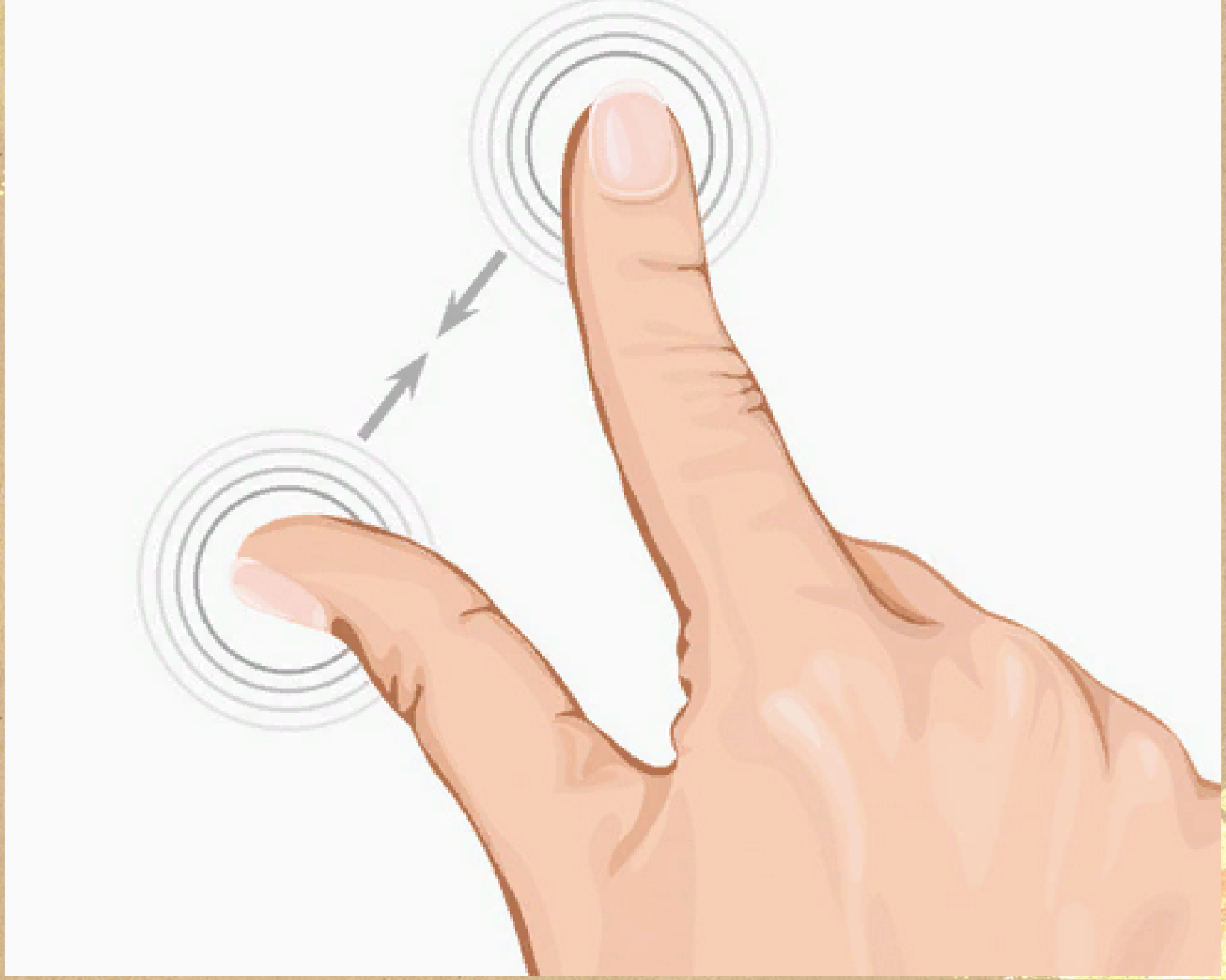
- Android menyediakan kelas GestureDetector untuk mendeteksi gerakan umum.
- Beberapa gerakan yang didukungnya termasuk onDown (), onLongPress (), onFling (), dan sebagainya.
- Pengembang dapat menggunakan GestureDetector bersamaan dengan metode onTouchEvent ()

Multi-Touch Gesture

- Gerakan multi-sentuh adalah ketika banyak pointer (jari) menyentuh layar secara bersamaan.
- Ini menjelaskan cara mendeteksi gerakan yang melibatkan banyak penunjuk (pointer).

Pointers





Style dan Tema

- Style didefinisikan dalam sumber daya XML yang terpisah dari XML yang menentukan tata letak.
- File XML ini berada di bawah direktori res / values / dari proyek

Contoh

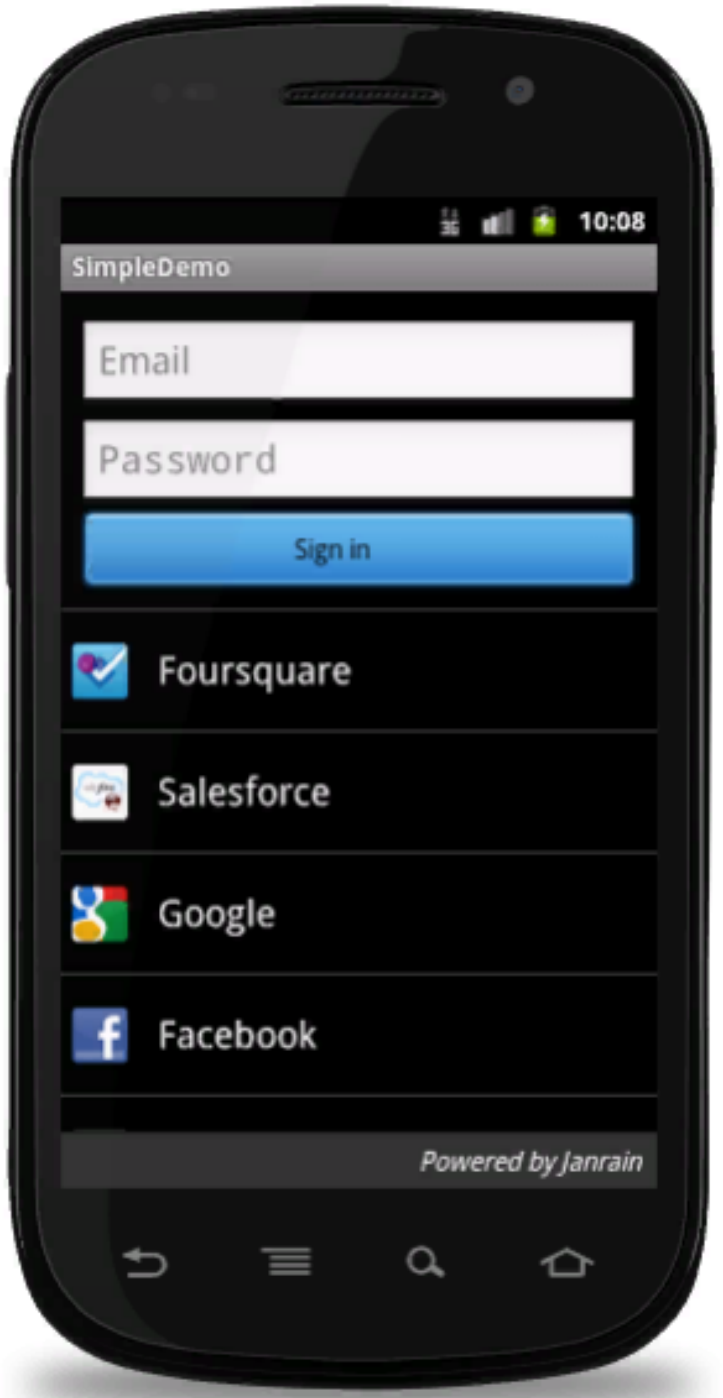
```
<resources>
  <style name="CustomFontStyle">
    <item name="android:layout_width">fill_parent</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:capitalize">characters</item>
    <item name="android:typeface">monospace</item>
    <item name="android:textSize">12pt</item>
    <item name="android:textColor">#00FF00</item>/>
  </style>
</resources>
```

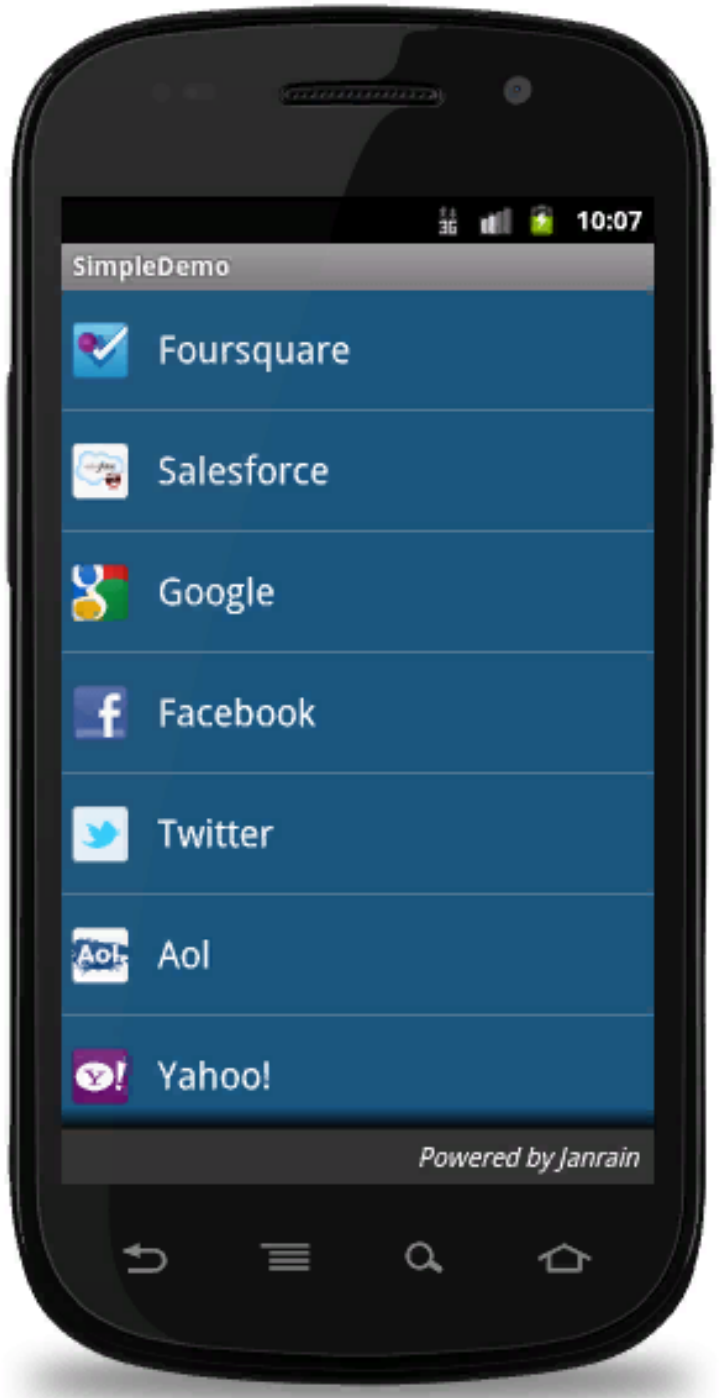
Menggunakan Style

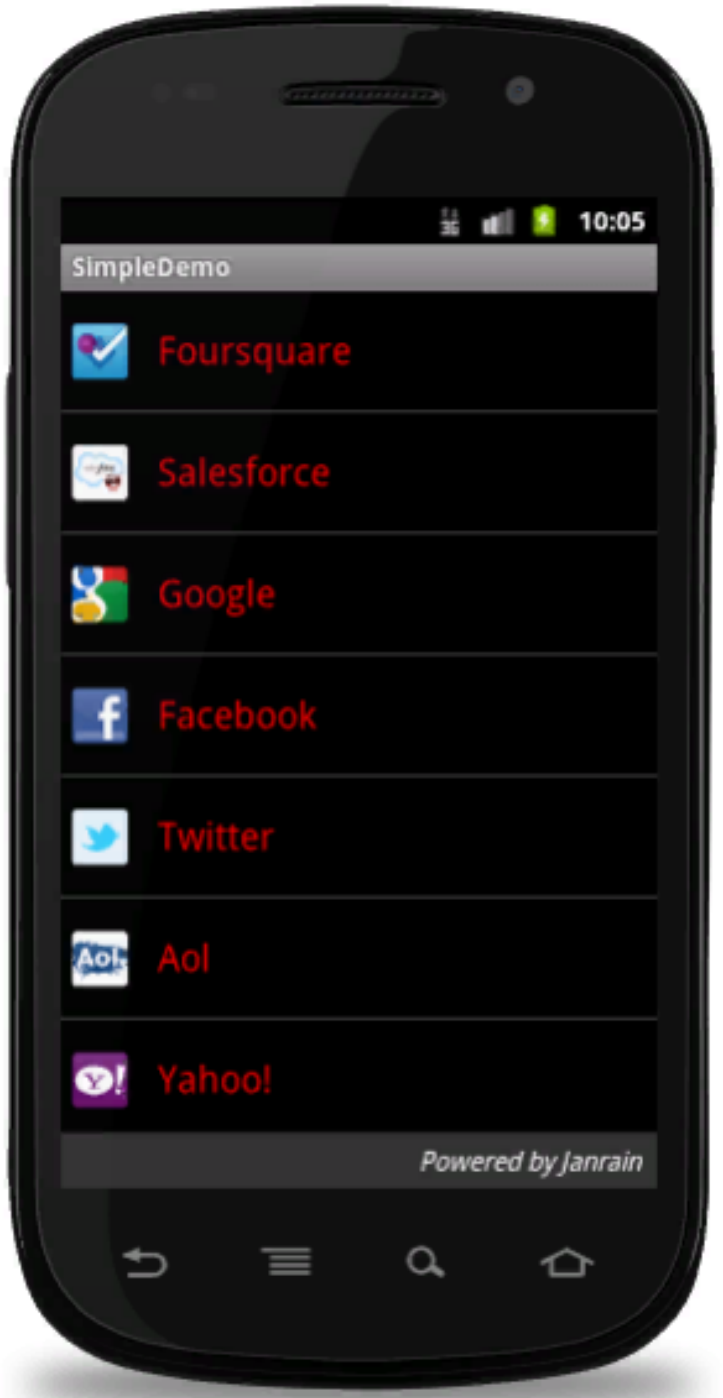
```
<TextView
    android:id="@+id/text_id"
    style="@style/CustomFontStyle"
    android:text="@string/hello_world" />
```

Menggunakan Ulang Style

```
<resources>
   ...
   <style name="MyCustomTheme"
parent="android:style/Theme">
   <item name="android:textColorPrimary">#ffff0000</item>
   </style>
   ...
</resources>
```


10:08
SimpleDemo
Email
Password
Sign in
Foursquare
Salesforce
Google
Facebook
Powered by Janrain


10:07
SimpleDemo
Foursquare
Salesforce
Google
Facebook
Twitter
Aol
Yahoo!
Powered by Janrain


10:05
SimpleDemo
Foursquare
Salesforce
Google
Facebook
Twitter
Aol
Yahoo!
Powered by Janrain

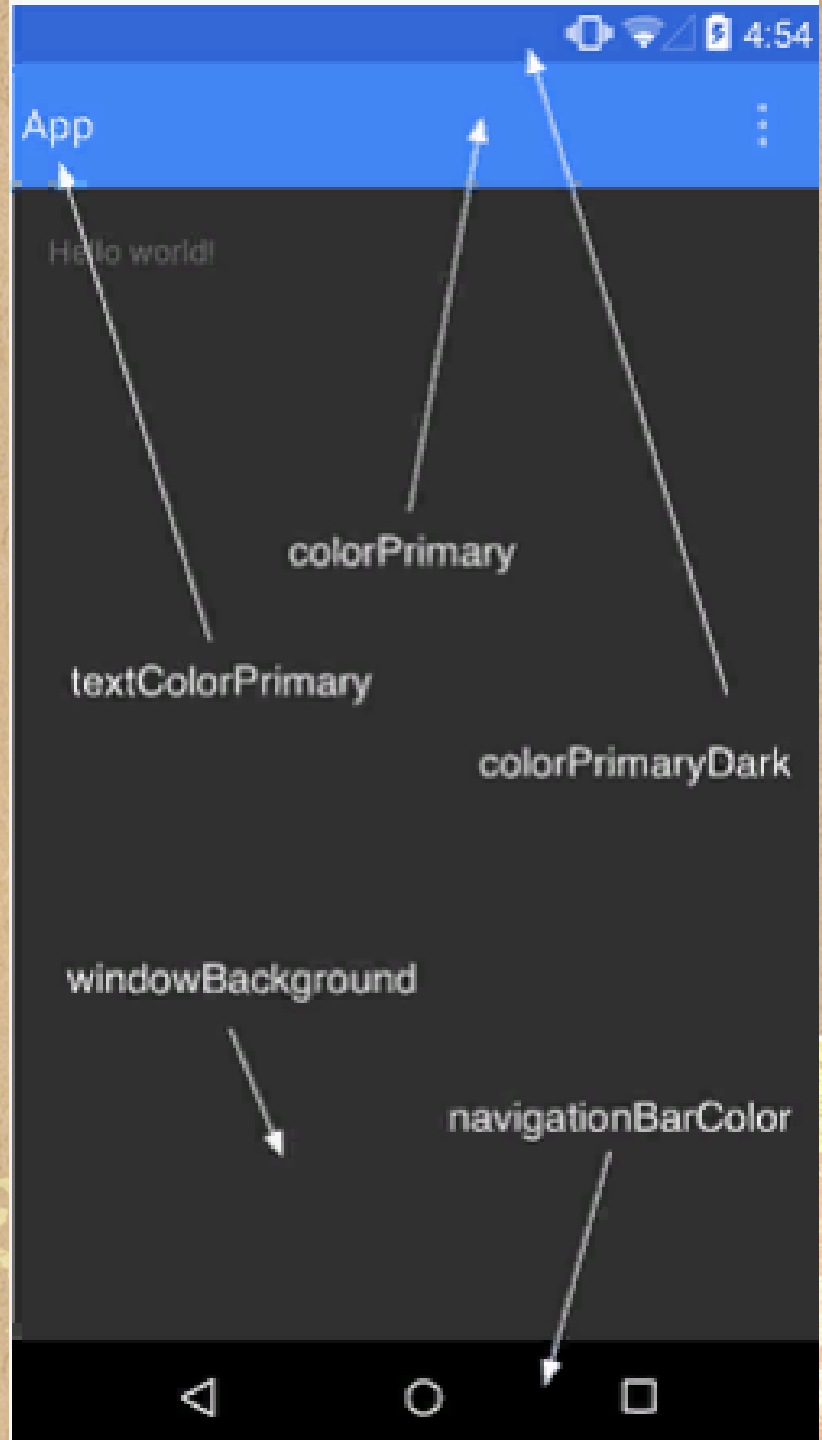
Tema Android

- Tema tidak lain adalah Style Android yang diterapkan pada seluruh Aktivitas atau aplikasi, dan bukan pada View individual.

- Dengan demikian, ketika diterapkan, setiap View dalam Aktivitas atau aplikasi akan menerapkan setiap properti Style yang didukungnya.

Palet Aktivitas

- Setiap elemen Aktivitas dapat diberikan atribut sesuai dengan keinginan



Implementasi

```
<resource>
  <style name="AppTheme" parent="android:Theme.Material">
    <item name
="android:color/primary">@color/primary</item>
    <item name
="android:color/primaryDark">@color/primary_dark</item>
    <item name
="android:colorAccent/primary">@color/accent</item>
  </style>
<resource>
```