

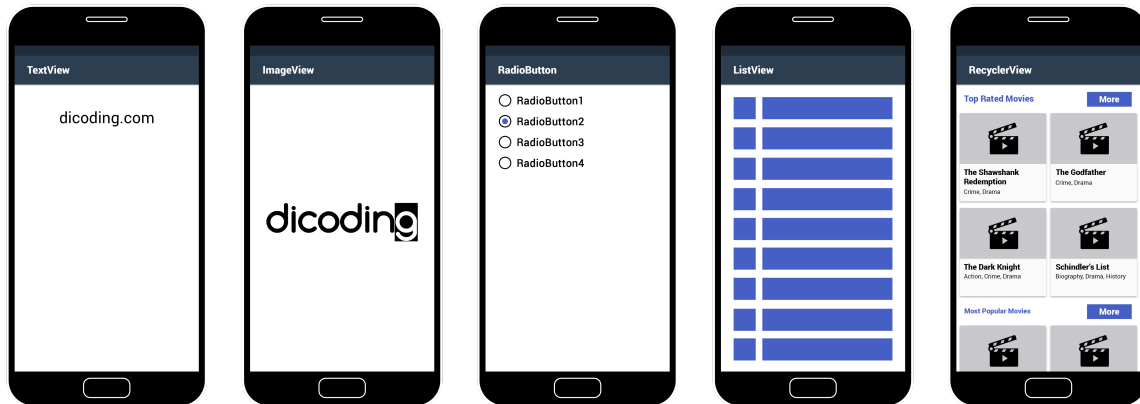
Modul 3 (Views & ViewGroup)

View

Sebuah view adalah obyek yang menggambar komponen tampilan ke layar yang mana pengguna dapat melihat dan berinteraksi langsung.

Contoh komponen turunan dari view seperti :

- **TextView,**
komponen yang berguna untuk menampilkan teks ke layar.
- **Button,**
komponen yang membuat pengguna dapat berinteraksi dengan cara ditekan untuk melakukan sesuatu.
- **ImageView,**
Komponen untuk menampilkan gambar.
- **ListView,**
komponen untuk menampilkan informasi dalam bentuk list.
- **GridView,**
komponen untuk menampilkan informasi dalam bentuk grid.
- **RadioButton,**
komponen yang memungkinkan pengguna dapat memilih satu pilihan dari berbagai pilihan yang disediakan.
- **Checkbox,**
komponen yang memungkinkan pengguna dapat memilih lebih dari satu dari pilihan yang ada.

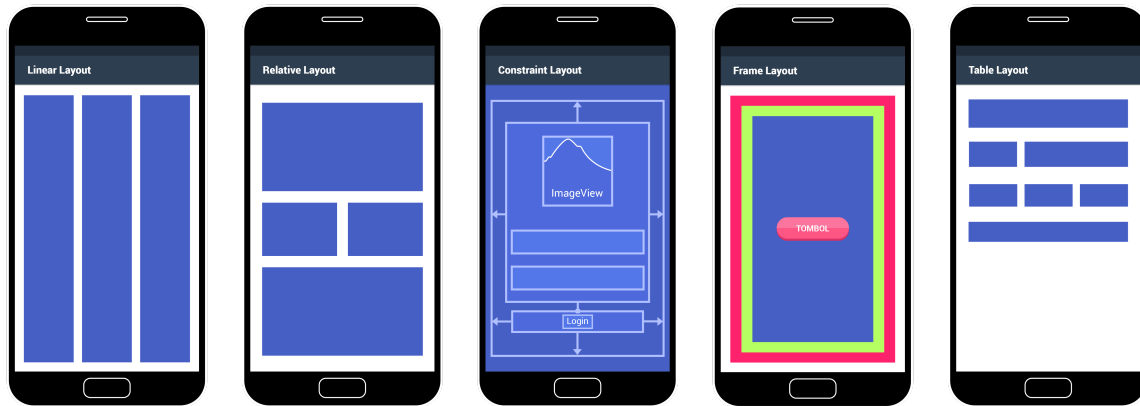


View Group

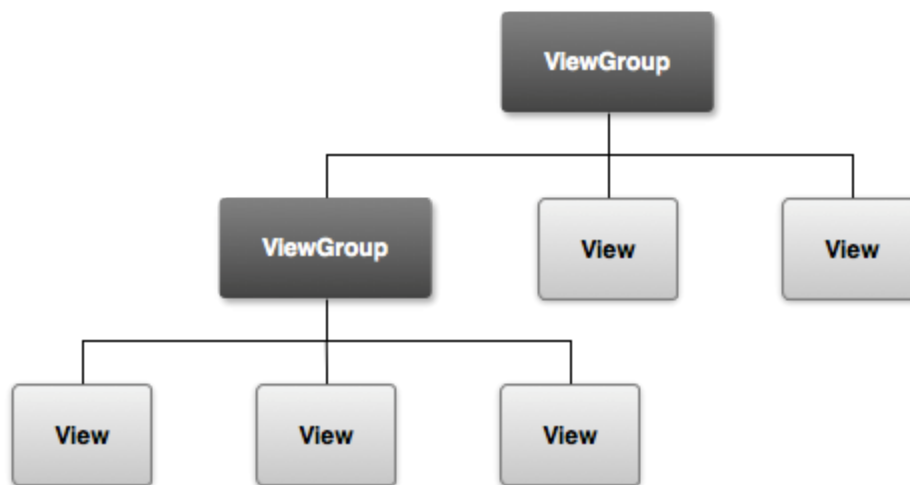
viewgroup adalah sebuah obyek yang mewadahi obyek-obyek view dan viewgroup itu sendiri sehingga membentuk satu kesatuan tampilan aplikasi yang utuh.

Contoh komponen viewgroup adalah:

- **LinearLayout**
- **RelativeLayout**
- **ConstraintLayout**
- **FrameLayout**
- **TableLayout**



Hierarki Komponen View dan Viewgroup



```

<?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="I am a TextView" />
    <Button android:id="@+id/button"

```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a Button" />
    </LinearLayout>
```

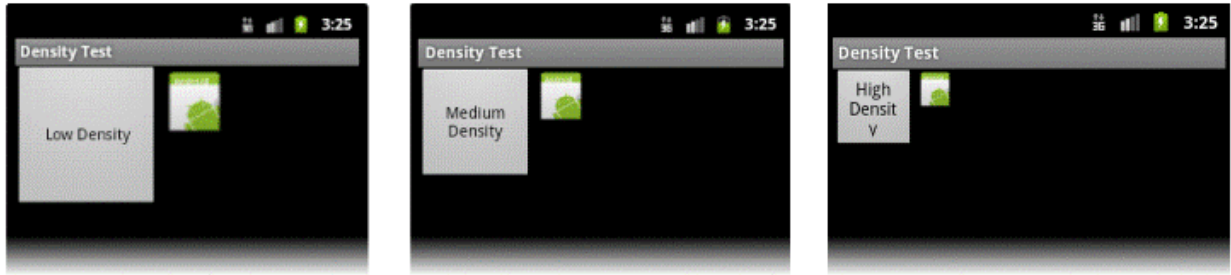
Pada contoh diatas dimana memiliki **ViewGroup** `Linear Layout` dan **View** ada `TextView` dan `Button`.

Satuan Dimensi di Android

Android sendiri memiliki satuan unit dimensi untuk ukuran tinggi dan lebar sebuah komponen view atau viewgroup. Berikut adalah esensi dari satuan dimensi unit di android.

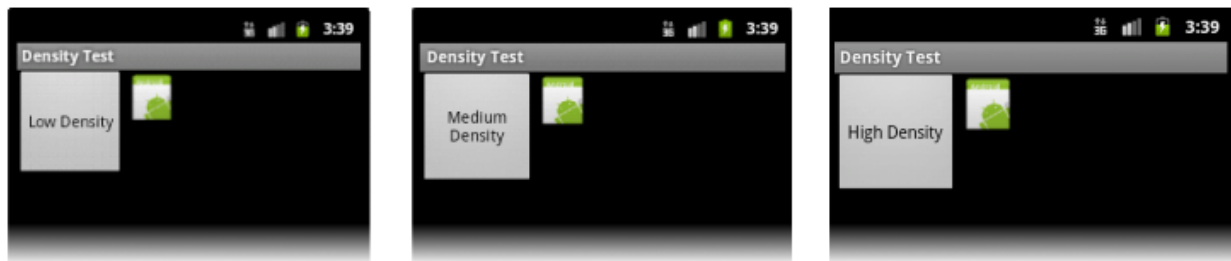
1. **Ekosistem Android dikenal dengan fragmentasi spesifikasi perangkat yang sangat bervariasi. Beragam perangkat Android memiliki perbedaan dimensi layar dan kerapatan *pixel* (*density*).**
2. **Untuk tampilan yang konsisten di perangkat Android, terdapat 2 jenis satuan, yaitu dip/dp (*density-independent pixel*) dan sp (*scale-independent pixels*).**
3. **Satuan dp/dip digunakan untuk satuan dari nilai dimensi misal width (attribut : layout_width) dan height (attribut : layout_height) dari sebuah komponen view atau viewgroup.**
4. **Satuan sp digunakan untuk ukuran teks. Perbedaannya dengan dp/dip adalah satuan sp android akan men-*scale* ukuran teks sesuai dengan setting ukuran teks di peranti (yang biasa dapat diakses melalui menu settings)**

Studi Kasus 1



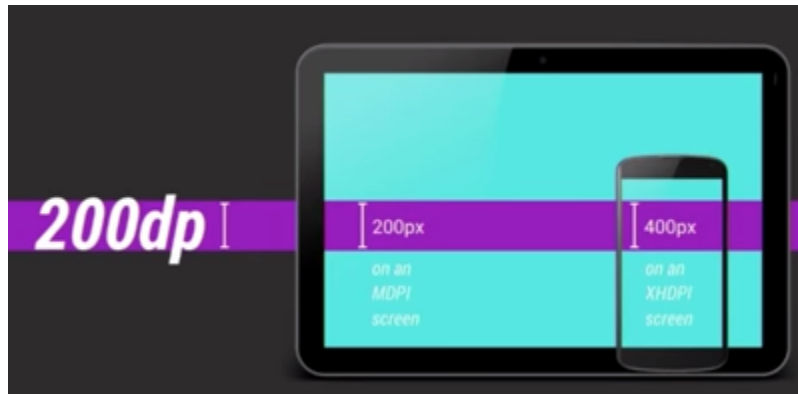
Misalkan ada dua tablet 7-inch (ukuran diagonal layar). Tablet pertama (A) memiliki resolusi layar 1200×1920px 320dpi. Tablet kedua (B) memiliki resolusi 2048×1536px 326dpi.

Sebuah tombol berukuran 300×300px akan tampak normal pada tablet A. Tapi tombol tersebut akan terlihat terlalu kecil pada tablet B.



Tapi akan berbeda jika kita tentukan ukuran button-nya dengan ukuran yang bergantung pada density layar. Bila kita menggunakan ukuran 300×300dp, maka tombol tersebut akan terlihat baik pada beragam perangkat dengan ukuran yang berbeda.

Studi kasus 2

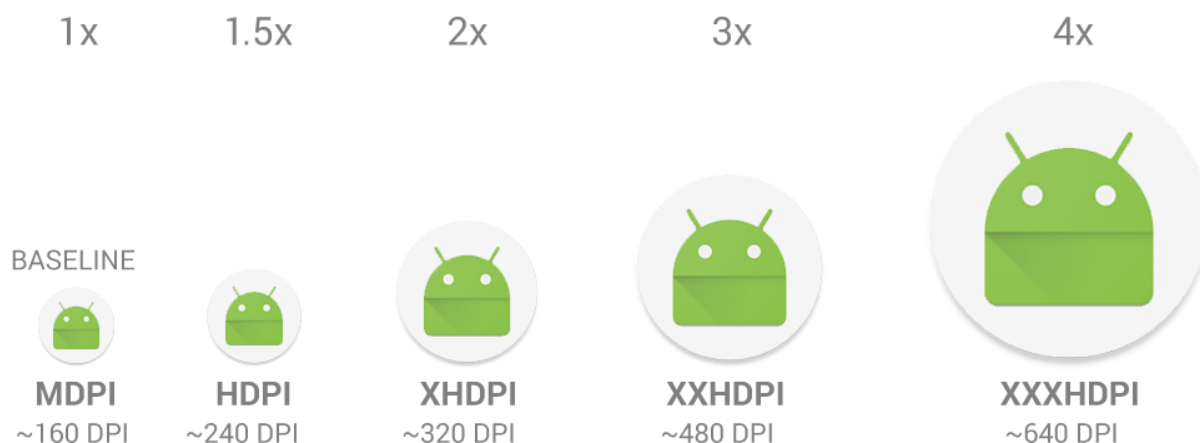


Pada gambar di atas ukuran 200dp akan dikonversi pada device mdpi (device dengan density 160dpi/dots per inch) menjadi 200px dan menjadi 400px pada device xhdpi (density 420dpi) misal pada nexus 4. Sehingga ukuran tersebut tampak sama dan konsisten secara fisik untuk beragam peranti dengan ukuran layar yang berbeda.

Gambar untuk Semua Jenis Layar

Supaya gambar pada aplikasi Anda tetap terlihat bagus pada semua jenis handphone dengan pixel yang berbeda-beda, maka Anda perlu menyiapkan gambar dengan berbagai versi untuk setiap jenis handphone, jika tidak maka gambar yang tampil akan terlihat kabur(blur).

Berikut ini adalah patokan (baseline) dalam menyiapkan gambar.



Vector Asset

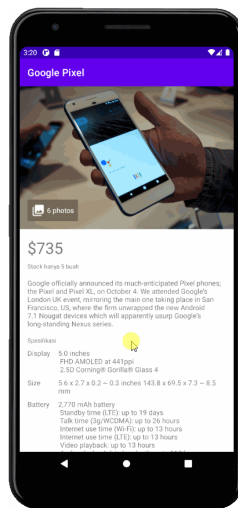
Selain menggunakan gambar dengan format JPG, PNG, atau tipe bitmap lainnya, Android Studio juga dapat menerima gambar dengan tipe vector.

Kelebihan dari penggunaan vector yaitu

- Gambar tidak akan pecah walaupun diperbesar.
- Biasa digunakan untuk icon aplikasi bukan berupa typografi

Code Lab View dan ViewGroup

Pada Code Lab kali ini kita akan mempraktekkan bagaimana penggunaan views dan Viewsgroup pada android



1. Kode Fullnya

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

        <FrameLayout
            android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content">

        <ImageView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:adjustViewBounds="true"
            android:scaleType="fitXY"
            android:src="@drawable/pixel_google"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="@style/TextAppearance.AppCompat.Small"
            android:textColor="@android:color/white"
            android:layout_gravity="bottom"
            android:layout_marginStart="16dp"
            android:layout_marginBottom="16dp"
            android:background="#4D000000"
            android:drawablePadding="4dp"
            android:gravity="center_vertical"
            android:padding="8dp"
            android:text="@string/dummy_photos"
            android:drawableStart="@drawable/ic_collections_white_18dp" />

    </FrameLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="8dp"
        android:text="@string/dummy_value"
        android:textSize="32sp" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="12sp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="16dp"
        android:text="@string/stock" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="16dp"
        android:lineSpacingMultiplier="1"
        android:text="@string/content_text" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```



```

        android:textSize="12sp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="8dp"
        android:text="@string/specification" />

<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="16dp">
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="14sp"
            android:layout_marginEnd="16dp"
            android:text="@string/display" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="14sp"
            android:layout_weight="1"
            android:text="@string/content_specs_display" />
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="16dp"
            android:textSize="14sp"
            android:text="@string/size" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:textSize="14sp"
            android:text="@string/content_specs_size" />
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="16dp"

```

```

        android:textSize="14sp"
        android:text="@string/battery" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="14sp"
        android:text="@string/content_specs_battery" />
    </TableRow>
</TableLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="16dp">
    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/profile_image"
        android:layout_width="56dp"
        android:layout_height="56dp"
        android:layout_centerVertical="true"
        android:layout_marginEnd="16dp"
        android:src="@drawable/photo_2" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_toEndOf="@+id/profile_image"
        android:text="@string/my_name" />
    </RelativeLayout>
    <Button
        android:text="@string/buy"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="16dp" />
</LinearLayout>

</ScrollView>

```

- Pada praktik kali ini kita menggunakan sebuah ViewGroup bernama ScrollView yang mana agar sebuah activity dapat discroll karena memiliki banyak informasi.
- Didalam ScrollView kita juga dapat membuat viewgroup lainnya seperti LinearLayout, Relative Layout dan lainnya seperti contoh diatas.

Glosarium

- **wrap_content**
memberi tahu tampilan Anda untuk menyesuaikan ukurannya dengan dimensi yang dibutuhkan oleh kontennya.
- **match_parent**
memberi tahu tampilan Anda sebesar yang diizinkan oleh grup tampilan induknya.
- **Komponen view dan viewgroup** memiliki dua buah atribut penting yang harus selalu diberikan nilai untuk mengatur posisi dirinya di dalam sebuah layout, yaitu: **layout_width** dan **layout_height**