

MODUL 3

Layout dengan Linear Layout dan Constrain Layout



CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu membuat Layout dengan Linear Layout dan Constraint Layout



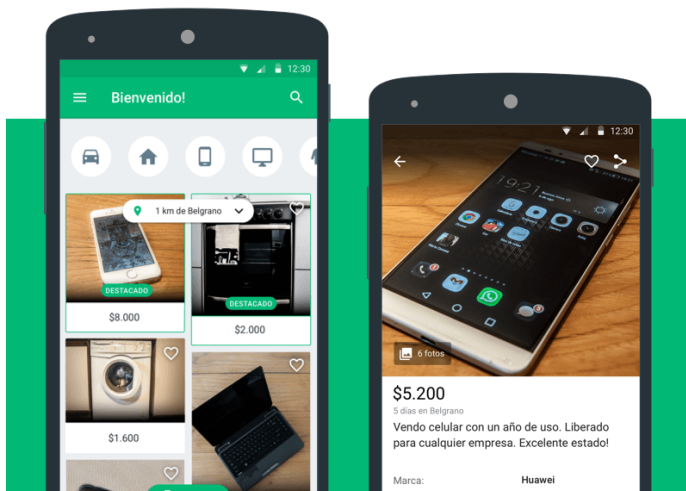
KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Android Studio 3.4.
2. Handphone Android versi 7.0 (Nougat)
3. Kabel data USB.
4. Driver ADB.



DASAR TEORI

Pada modul ini, kita akan mempelajari komponen View dan ViewGroup. Kedua komponen ini dapat berkolaborasi sehingga membentuk antar muka dengan contoh seperti pada gambar di bawah ini:



Pada dasarnya semua elemen antar pengguna di aplikasi Android dibangun menggunakan dua buah komponen inti, yaitu view dan viewgroup.

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

Contoh komponen turunan dari view seperti :

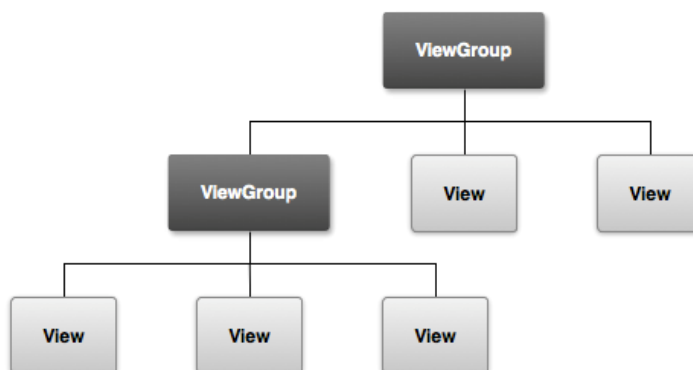
- **TextView**, komponen yang berguna untuk menampilkan teks ke layar.
- **Button**, komponen yang membuat pengguna dapat berinteraksi dengan cara ditekan untuk melakukan sesuatu.
- **ImageView**, Komponen untuk menampilkan gambar.
- **ListView**, komponen untuk menampilkan informasi dalam bentuk list.
- **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.

Sedangkan 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**
- **FrameLayout**
- **RelativeLayout**
- **TableLayout**

Hierarki komponen view dan viewgroup dapat digambarkan dengan diagram berikut:



Jika diterjemahkan di dalam sebuah viewgroup akan ditampung dua buah komponen view dan satu komponen viewgroup yang terdiri dari 3 buah komponen view. Salah satu contoh dari tampilan dalam file layout xml untuk merepresentasikan kolaborasi view dan viewgroup seperti ini :

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:orientation="vertical" >
6.     <TextView android:id="@+id/text"
7.         android:layout_width="wrap_content"
```

```

8.         android:layout_height="wrap_content"
9.         android:text="I am a TextView" />
10.    <Button android:id="@+id/button"
11.        android:layout_width="wrap_content"
12.        android:layout_height="wrap_content"
13.        android:text="I am a Button" />
14. </LinearLayout>

```

Obyek turunan viewgroup **LinearLayout** menjadi kontainer untuk obyek turunan view, button, dan textview. Beberapa komponen viewgroup seperti linearlayout, relativelayout, framelayout, dan tablelayout merupakan komponen yang paling banyak digunakan untuk menjadi *parent/root* dari komponen-komponen view.

Berikut adalah definisi singkat dan inti dari komponen-komponen di atas terhadap penempatan komponen view (*child*) di dalamnya. Kita akan membahas Linear Layout dan Constrain Layout.

LinearLayout

Layout ini akan menempatkan komponen-komponen di dalamnya secara horizontal atau vertikal. Linearlayout memiliki atribut weight untuk masing-masing *child* view yang berguna untuk menentukan porsi ukuran view dalam sebuah ruang (*space*) yang tersedia.



android:orientation="vertical"



android:orientation="horizontal"

Constrain Layout. Apa itu ConstraintLayout? (<https://blog.dicoding.com/kenal-lebih-dekat-dengan-constraintlayout/>)

ConstraintLayout merupakan salah satu komponen **ViewGroup** yang dapat kita gunakan untuk menyusun tampilan aplikasi yang kompleks tanpa adanya nested layout. ConstraintLayout tersedia dengan dukungan kompatibilitas mulai dari Android 2.3 (API Level 9) sampai dengan yang terbaru.

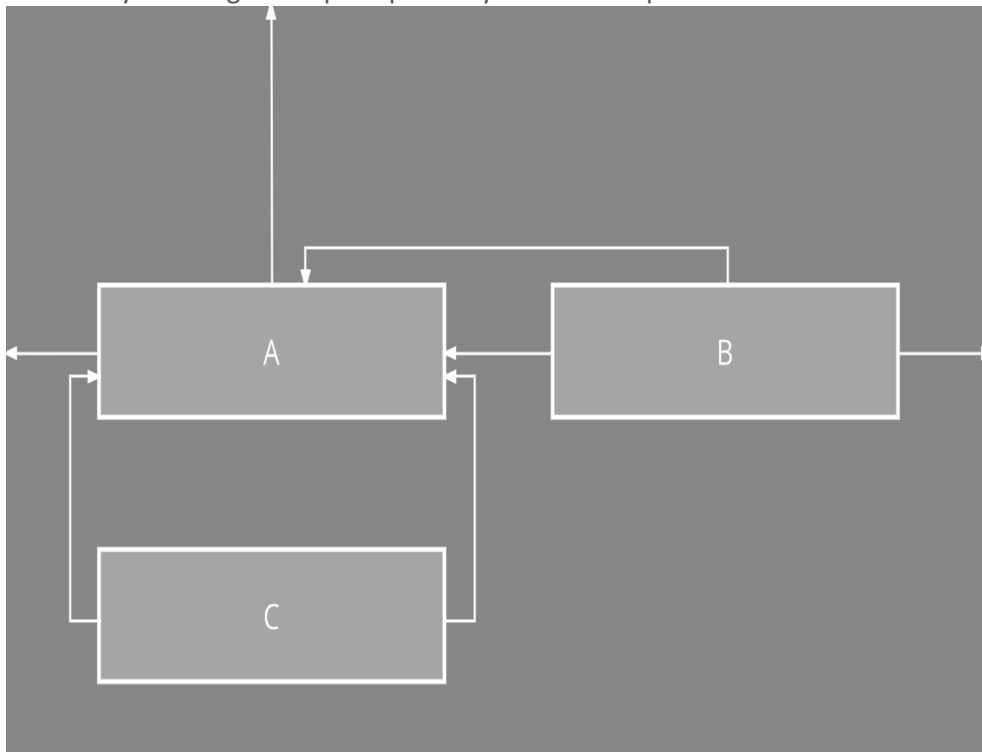
ConstraintLayout memiliki kesamaan dengan RelativeLayout. Dalam penggunaan semua view yang berada di dalamnya disusun berhubungan antara parent dan view lainnya. Tapi **ConstraintLayout** lebih fleksibel dari **RelativeLayout** dan mudah digunakan dengan dukungan Layout Editor pada Android Studio.

Let's say kita menambah view baru ke dalam **ConstraintLayout**. Kita gunakan *drag and drop* di Layout Editor yang berada pada tab **Design** atau dengan menambahnya secara manual melalui tab **Text**. Kita perlu menentukan posisi dari view atau bagaimana agar view tersebut terhubung dengan parent layout atau view lainnya.

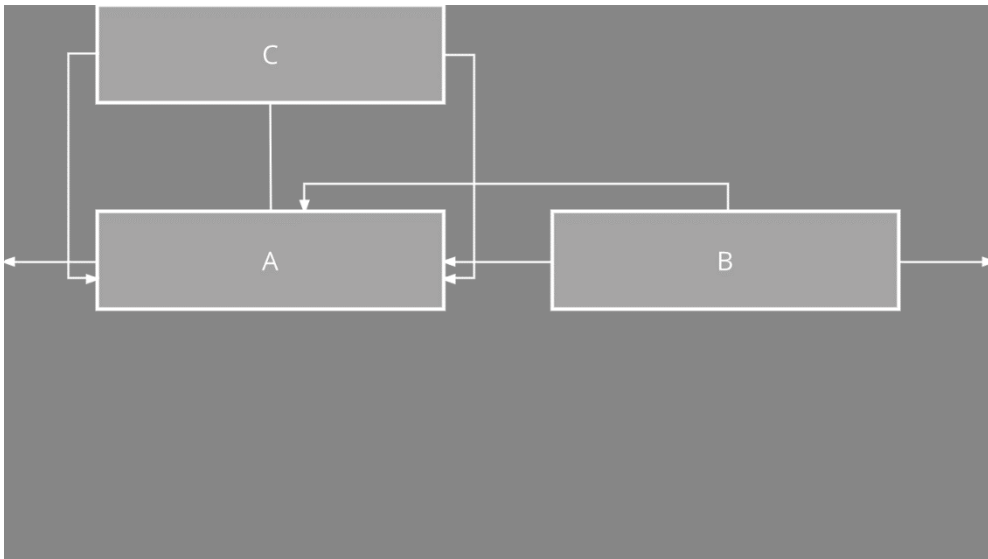
Kenapa gerangan? Karena setelah ditambahkan, view tersebut tidak memiliki **constraint** yang menghubungkannya dengan parent layout atau view lainnya. Sehingga ketika dijalankan, posisi dari view tersebut akan berada di bagian atas sebelah kiri.

Berbeda ceritanya dengan RelativeLayout. Saat kita ingin menentukan posisi atau menghubungkan dua buah view, kita bisa menggunakan attribute seperti **layout_below** atau **layout_above**. Nah untuk **ConstraintLayout** kita akan menggunakan **constraint** sebagai dasar dalam menentukan posisi agar sebuah view dapat terhubung dengan view lainnya sesuai harapan kita.

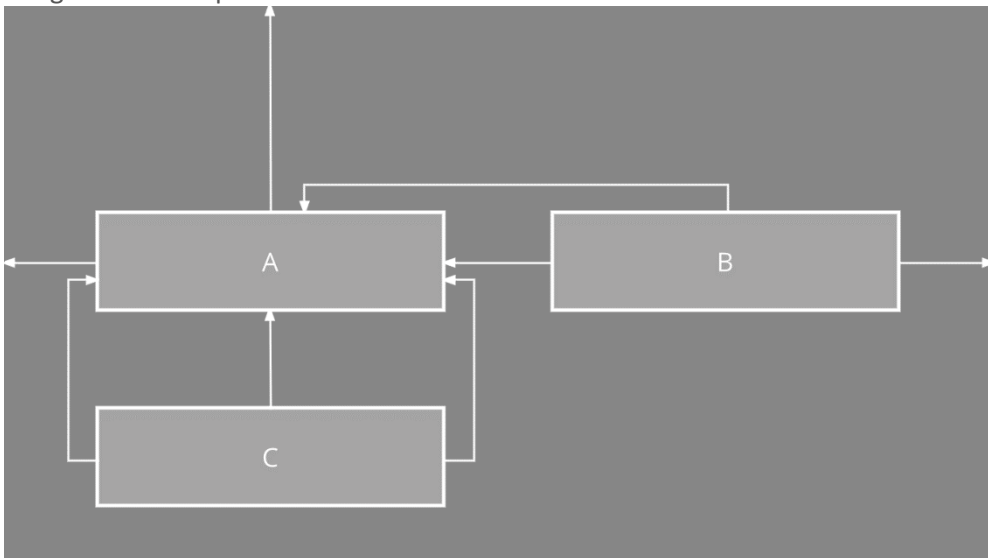
Setiap view setidaknya memiliki satu *vertikal* dan *horizontal constraint*. Misal kita memiliki sebuah layout dengan tampilan pada Layout Editor seperti berikut:



Susunan tampilan di atas akan terlihat normal. Tidak ada yang salah di Layout Editor. Tapi jika kita perhatikan seksama, **view C** diatas hanya memiliki *horizontal constraint* yang diatur sejajar dengan **view A**. Sehingga ketika jika kita coba menjalankannya, sama seperti yang disebutkan diatas, maka posisi dari **view C** akan berada di posisi atas seperti berikut:



Berbeda jika kita menambahkan vertikal constraint pada **view C** yang diatur terikat dengan **view A** seperti berikut:

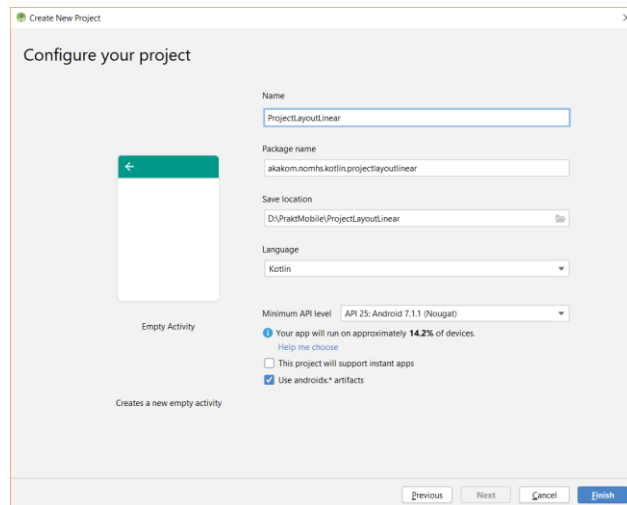


Ketika dijalankan, apa yang terjadi? Yang tampil akan sesuai dengan apa yang terlihat di Layout Editor.



PRAKTIK

1. Buatlah project dengan nama ProjectLayoutLinear dengan cara **klik** menu **File** → **New** → **New Project ...**
2. Kemudian pilih **Empty Activity**, lalu **klik** button **Next**
3. Beri project anda yang baru dengan nama ProjectLayoutLinear, kemudian **klik** button **Finish**.



4. Langkah selanjutnya, buka file activity_main.xml yang terdapat pada app → res → layout.
5. Secara default pada saat membuat project baru dengan nama ProjectLayoutLinear maka akan muncul teks Hello World.
6. Buka file activity_main.xml kemudian pilih tab text, akan terlihat koding sebagai berikut.

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

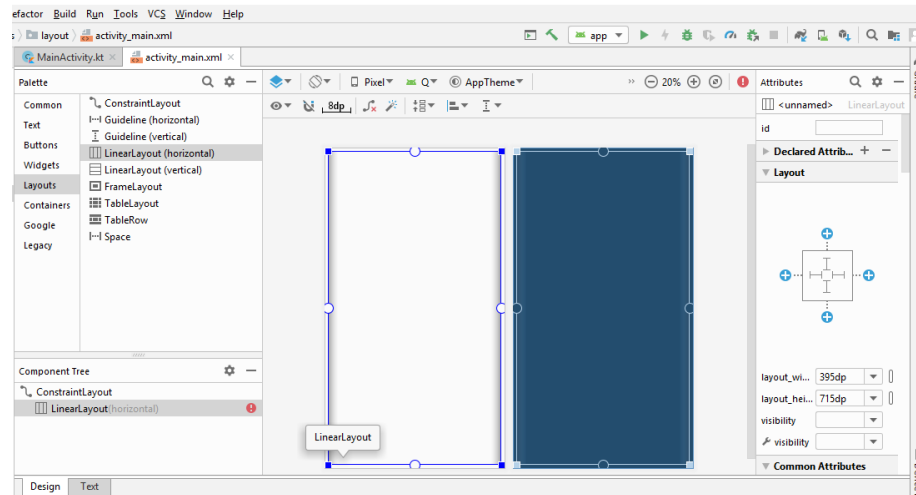
</androidx.constraintlayout.widget.ConstraintLayout>
```

7. Standar layout pertama adalah Constrain Layout. Ubahlah menjadi Linear Layout dengan koding berikut. Perhatikan atribut **orientation**.

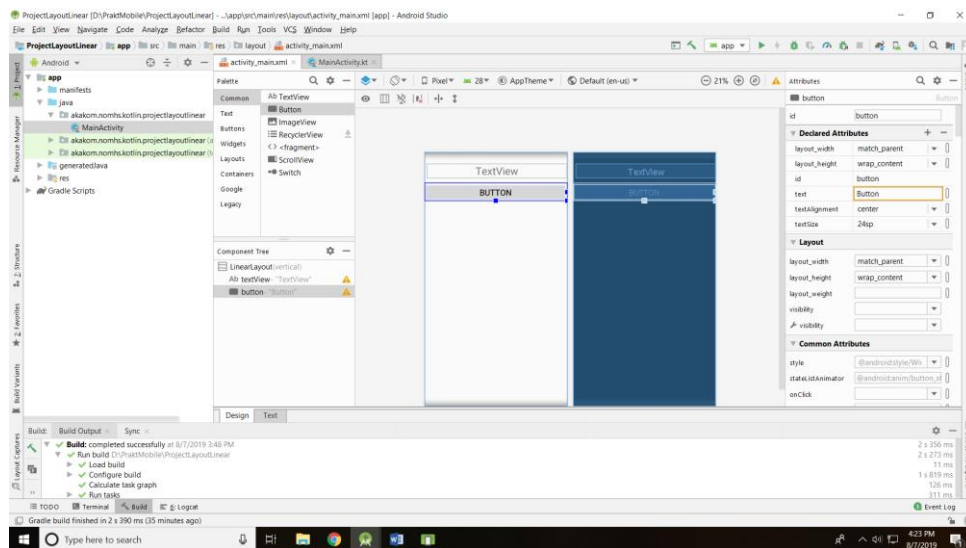
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

</LinearLayout>
```

8. Maka hasilnya akan terlihat sebuah linear layout berhasil ditempatkan di bagian area atau canvas.




9. Tambahkan komponen TextView dan Button, sehingga menjadi sebagai berikut.

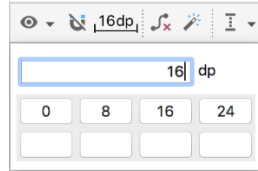


10. Jalankan dan amati hasilnya.
11. Ubah atribut-atribut yang ada dan jalankan lagi, amati perubahannya.
12. Kita akan mencoba menggunakan Constrain Layout. Buat project baru. Beri nama ProjectLayoutConstrain.
13. Buat Project baru dengan nama ProjectLayoutConstrain.
14. Buka file activity_main.xml dan klik tab Desain.
15. Anda akan menambahkan constrain secara manual, maka koneksi otomatis kita matikan. Di toolbar, temukan **Turn Off/On Autoconnect** toggle button, yang ditunjukkan di bawah ini. (Jika Anda tidak dapat melihat toolbar, klik di dalam area editor desain dari Layout Editor.) Pastikan autoconnect tidak aktif.



 Autoconnect is on.

 Autoconnect is off

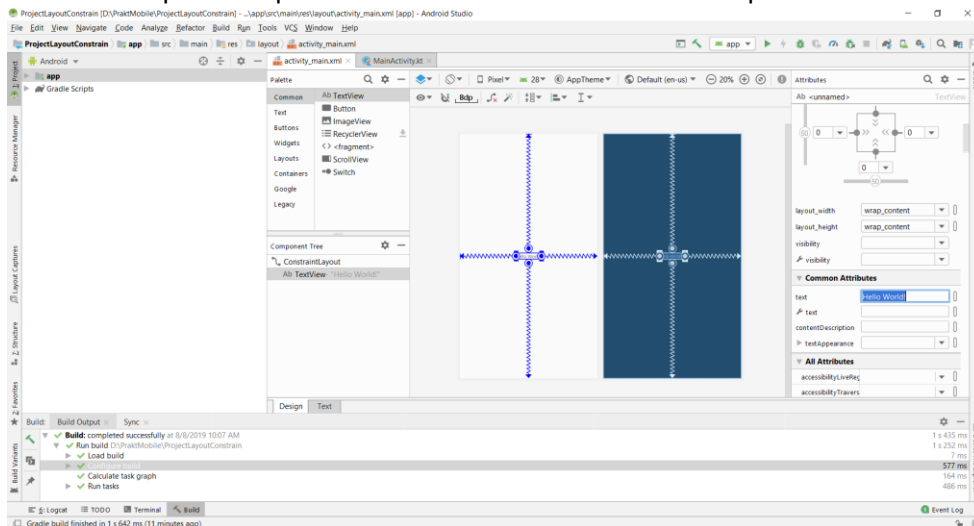
16. Gunakan toolbar untuk mengatur default margins ke 16dp. (Defaultnya adalah 8dp.)



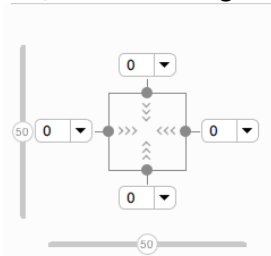
17. Ketika Anda mengatur margin default ke 16dp, constrain baru dibuat dengan margin ini, jadi Anda tidak perlu menambahkan margin setiap kali Anda menambahkan constrain.

18. Perbesar menggunakan ikon +  33%  di sebelah kanan toolbar, hingga teks Hello World terlihat di dalam tampilan teksnya.

19. Klik dua kali pada tampilan teks Hello World untuk membuka panel Attributes.

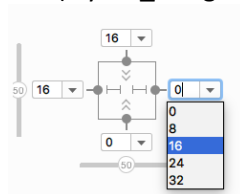


20. View Inspector, yang ditunjukkan pada gambar di bawah, adalah bagian dari panel Atribut. View Inspector mencakup kontrol untuk atribut layout seperti constrain, constraint types, constraint bias, and view margins.



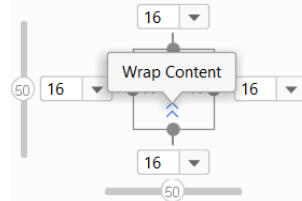
21. Constraint bias menempatkan elemen tampilan di sepanjang sumbu horizontal dan vertikal. Secara default, tampilan dipusatkan di antara dua constrain dengan bias 50%. Untuk menyesuaikan bias, Anda dapat menarik slider bias di view inspector. Menarik slider bias mengubah posisi tampilan sepanjang sumbu.



22. Tambahkan margin untuk TextView Hello World. Perhatikan bahwa dalam view inspector, margin kiri, kanan, atas, dan bawah untuk tampilan teks adalah 0. Margin default tidak ditambahkan secara otomatis, karena tampilan ini dibuat sebelum Anda mengubah margin default. Untuk margin kiri, kanan, dan atas, pilih 16dp dari menu drop-down di inspektur tampilan. Misalnya, dalam tangkapan layar berikut ini Anda menambahkan layout_marginEnd (layout_marginRight).

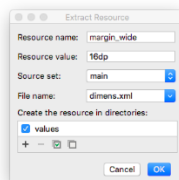


23. Sesuaikan batasan dan margin untuk TextView. Di view inspector, panah >>> di dalam kotak mewakili tipe constrain:

- a. >>> wrap content: Tampilan hanya selebar kontennya.



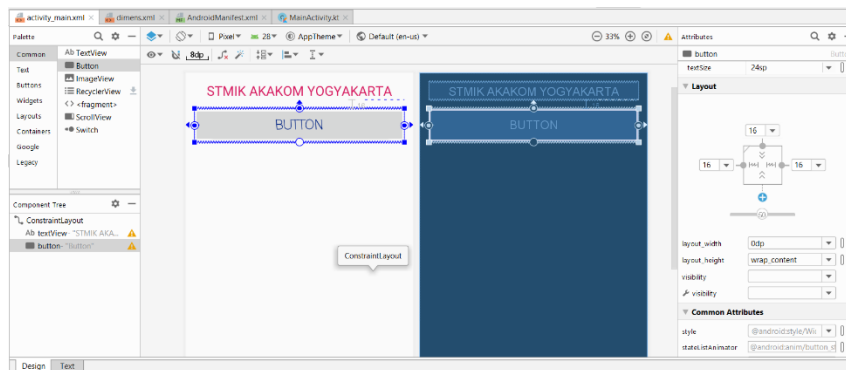
- b.  fixed: Anda dapat menentukan dimensi sebagai margin tampilan di kotak teks di sebelah panah constrain tetap.
- c.  match constrain: Tampilan melebar sebanyak mungkin untuk memenuhi constrain di setiap sisi, setelah memperhitungkan margin tampilan sendiri. Constrain ini sangat fleksibel, karena memungkinkan layout untuk beradaptasi dengan berbagai ukuran dan orientasi layar. Dengan membiarkan tampilan sesuai dengan constrain, Anda membutuhkan layout yang lebih sedikit untuk aplikasi yang Anda buat.
24. Di view inspector, ubah constrain kiri dan kanan ke Match Constraints. (Klik simbol panah untuk beralih di antara jenis constrain.)
25. Pindah ke tab Text, Ekstrak resource dimensi untuk layout_marginStart, dan atur nama Resource ke margin_wide. (blok pada bagian isian dari layout_marginStart, pilih extract dimensions resource).



26. Kerjakan untuk layout_marginEnd dan layout_marginTop. Hasilnya adalah sebagai berikut

```
android:layout_marginTop="@dimen/margin_wide"
android:layout_marginStart="@dimen/margin_wide"
android:layout_marginEnd="@dimen/margin_wide"
```

27. Kemudian, tambahkan satu button, atur constrainnya. Hasilnya sebagai berikut.



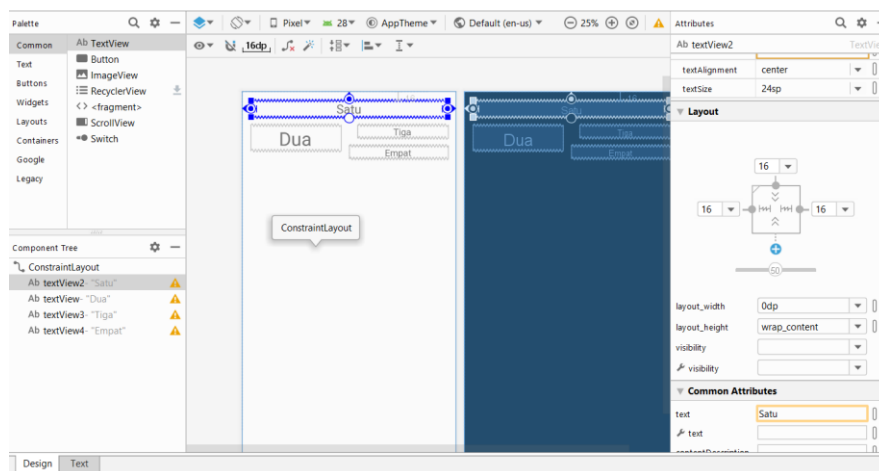
28. Jalankan dan amati hasilnya.

29. Ubah atribut-atribut yang ada dan jalankan lagi, amati perubahannya.



LATIHAN

1. Buat project baru dengan desain sebagai berikut.



2. Buat project baru dengan menggunakan Linear Layout dengan minimal 3 komponen (TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.

3. Buat project baru dengan menggunakan Constrains Layout dengan minimal 3 komponen (TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.



TUGAS

1. Buat Project pada perangkat komputer anda untuk mengimplementasikan layout dengan berbagai bentuk tampilan.



REFERENSI

1. <https://kotlinlang.org/docs/reference/>
2. <https://developer.android.com/kotlin>
3. <https://developer.android.com/courses/kotlin-android-fundamentals/toc>
4. <https://codelabs.developers.google.com/android-kotlin-fundamentals/>
5. <https://developer.android.com/kotlin/learn>
6. <https://developer.android.com/kotlin/resources>