

Dokumentasi Singkat Layout di XML

XML

- **android:layout_width** = Digunakan untuk menentukan lebar dari suatu item di XML (baik View ataupun ViewGroup)
- **android:layout_height** = Digunakan untuk menentukan tinggi dari suatu item di XML (baik View ataupun ViewGroup)
- **match_parent** = Lebar dari item XML akan mengikuti lebar dari parent view (ViewGroup)
- **wrap_content** = Lebar dari item XML akan mengikuti lebar dari konten yang terdapat di dalam item tersebut
- **match_constraint (0dp)** = Lebar dari item XML akan mengikuti lebar dari parent constraint (hanya digunakan di ViewGroup yang support dengan metode constraint seperti ConstraintLayout dan RelativeLayout).
- **android:id** = Digunakan untuk menentukan id dari suatu item di XML (id item harus berbeda antara 1 item dengan item yang lain).
- **android:background** = Digunakan untuk mengganti background dari sebuah item XML (biasanya digunakan di dalam ViewGroup).
- **android:src** = Digunakan untuk mengganti resources file di dalam item XML (biasanya digunakan di dalam view yang berkaitan dengan gambar seperti misalnya ImageView, ImageButton, dll).
- **android:text** = Digunakan untuk mengganti text di dalam item XML.
- **android:fontFamily** = Digunakan untuk mengganti font yang digunakan di dalam item XML.
- **android:textColor** = Digunakan untuk mengganti warna teks yang digunakan di dalam item XML.



- **android:layout_margin** = Digunakan untuk memberikan margin ke seluruh arah (kiri kanan atas bawah).
- **android:layout_marginTop** = Digunakan untuk memberikan margin ke atas dari item XML tersebut.

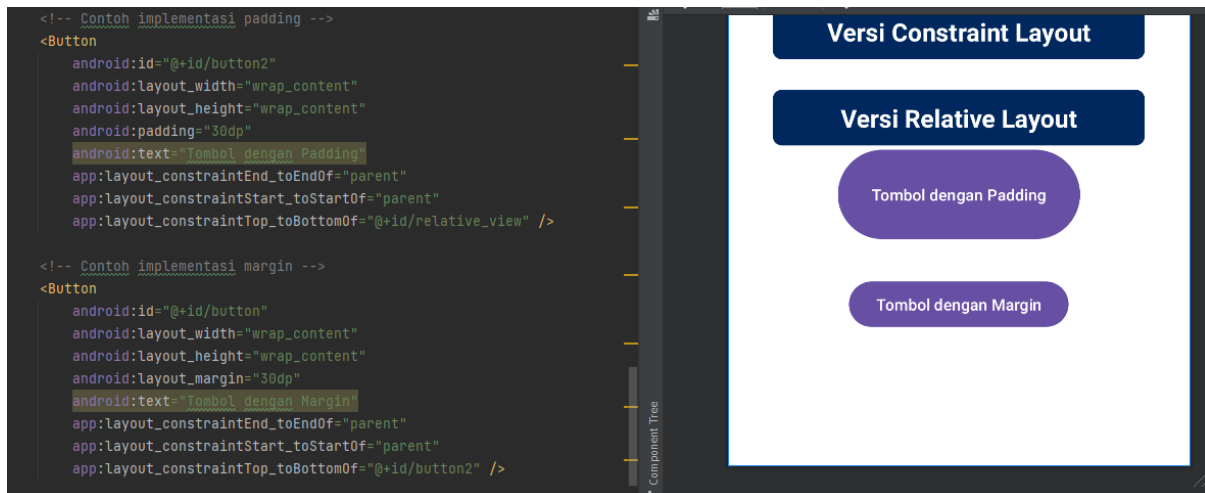
- **android:layout_marginBottom** = Digunakan untuk memberikan margin ke bawah dari item XML tersebut.
- **android:layout_marginStart** = Digunakan untuk memberikan margin di sebelah kiri dari item XML tersebut.
- **android:layout_marginEnd** = Digunakan untuk memberikan margin di sebelah kanan dari item XML tersebut.
- **android:padding** = Digunakan untuk memberikan padding ke seluruh arah (kiri kanan atas bawah)
- **android:paddingTop** = Digunakan untuk memberikan padding ke atas dari item XML tersebut.
- **android:paddingBottom** = Digunakan untuk memberikan padding ke bawah dari item XML tersebut.
- **android:paddingStart** = Digunakan untuk memberikan padding di sebelah kiri dari item XML tersebut.
- **android:paddingEnd** = Digunakan untuk memberikan padding di sebelah kanan dari item XML tersebut.



Gambar 1. Perbedaan antara Padding dan Margin

Padding: Jarak antara elemen di dalam elemen tersebut (Contoh: Jarak antara text yang terdapat di dalam button)

Margin: Jarak antara elemen dengan elemen lain di luar elemen tersebut (Contoh: Jarak antara button 1 dengan button yang lain)



Gambar 2. Contoh Implementasi Padding dan Margin

Linear Layout

- **android:orientation** = Mengatur orientasi dari item yang terdapat di linear layout, jika diset vertical maka item akan berurutan secara tegak lurus dan jika diset horizontal maka item akan berurutan secara garis datar (mendatar)
- **horizontal** = Mengatur orientasi dari item yang terdapat di linear layout menjadi mendatar
- **vertical** = Mengatur orientasi dari item yang terdapat di linear layout menjadi tegak lurus

Constraint Layout (untuk ilustrasinya silahkan cek link [berikut ini](#), lihat materi dengan judul **Analogi Materi XML ConstraintLayout**)

- **app:layout_constraintStart_toStartOf** = start dari item xml tersebut akan sejajar dengan start dari item xml yang lain.
- **app:layout_constraintStart_toEndOf** = start dari item xml tersebut akan sejajar dengan end dari item xml yang lain.
- **app:layout_constraintEnd_toEndOf** = end dari item xml tersebut akan sejajar dengan end dari item xml yang lain.
- **app:layout_constraintEnd_toStartOf** = end dari item xml tersebut akan sejajar dengan start dari item xml yang lain.
- **app:layout_constraintTop_toTopOf** = top dari item xml tersebut akan sejajar dengan top dari item xml yang lain.
- **app:layout_constraintTop_toBottomOf** = top dari item xml tersebut akan sejajar dengan bottom dari item xml yang lain.
- **app:layout_constraintBottom_toBottomOf** = bottom dari item xml tersebut akan sejajar dengan bottom dari item xml yang lain.
- **app:layout_constraintBottom_toTopOf** = bottom dari item xml tersebut akan sejajar dengan top dari item xml yang lain.

Relative Layout (untuk ilustrasinya silahkan cek link [berikut ini](#), lihat materi dengan judul **Analogi Materi XML RelativeLayout**)

- **android:layout_alignParentStart** = start dari item tersebut akan terhubung dengan start dari parent view
- **android:layout_alignParentEnd** = end dari item tersebut akan terhubung dengan end dari parent view
- **android:layout_alignParentTop** = top dari item tersebut akan terhubung dengan top dari parent view
- **android:layout_alignParentBottom** = bottom dari item tersebut akan terhubung dengan bottom dari parent view
- **android:layout_alignStart** = start dari item tersebut akan terhubung dengan start dari item lain
- **android:layout_alignEnd** = end dari item tersebut akan terhubung dengan end dari item lain
- **android:layout_alignTop** = top dari item tersebut akan terhubung dengan top dari item lain
- **android:layout_alignBottom** = bottom dari item tersebut akan terhubung dengan bottom dari item lain
- **android:layout_toStartOf** = end dari item tersebut akan terhubung dengan start dari item lain
- **android:layout_toEndOf** = start dari item tersebut akan terhubung dengan end dari item lain
- **android:layout_below** = top dari item tersebut akan berada di bottom dari item lain
- **android:layout_above** = bottom dari item tersebut akan berada di top dari item lain