



## Flutter Layout

*Pertemuan ke 9: Layout & Styling*

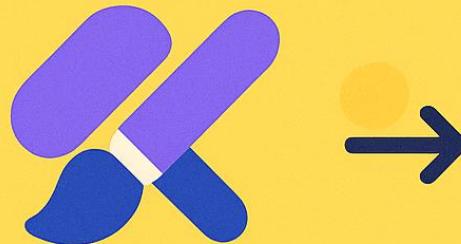
# Flutter Layout dan Styling

Flutter menyediakan berbagai widget untuk membuat layout dan gaya yang fleksibel.

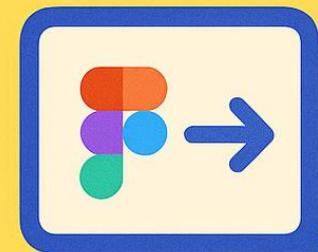
Dalam materi ini, kita akan membahas beberapa teknik dasar untuk membuat tampilan yang responsif dan menarik



Widget Layout



Styling



Slicing dari Figma

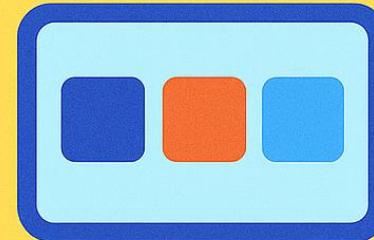
# Flexible dan Expanded Widgets

## Perbedaan **Flexible vs Expanded:**

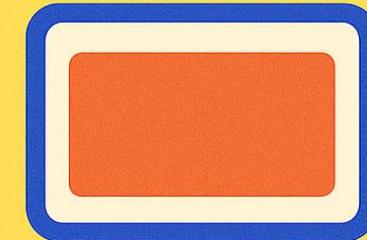
Expanded mengambil seluruh ruang kosong, sementara Flexible menyesuaikan ruang sesuai proporsi.

**Flexible:** Menyesuaikan ukuran widget di dalam baris atau kolom sesuai dengan ruang yang tersedia.

**Expanded:** Memperluas widget untuk mengisi ruang yang tersisa



**Flexible**



**Expanded**

# Flexible dan Expanded Widgets

main.dart

Dart

```
Column(
  children: [
    Expanded(
      child: Container(
        color: Colors.green,
      ),
    ),
    Flexible(
      fit: FlexFit.loose,
      child: Container(
        height: 100,
        color: Colors.blue,
      ),
    ),
  ],
)
```



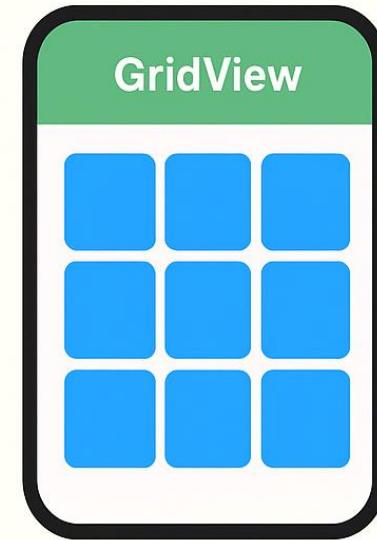
# GridView dan ListView

 **ListView:** Menggunakan widget untuk menampilkan daftar vertikal yang bisa di-scroll.

 **GridView:** Menggunakan widget untuk menampilkan daftar dalam bentuk grid. ♡♡

## Variasi ListView:

- ✓ **ListView():** Daftar statis dengan item yang didefinisikan langsung.
- ✓ **GridView.builder():** Efisien untuk daftar panjang dengan lazy loading.
- ✓ **GridView.separated():** Sama seperti ListView.builder(), tetapi dengan pemisah antar item.
- ⋮
- ⋮



## Variasi GridView:

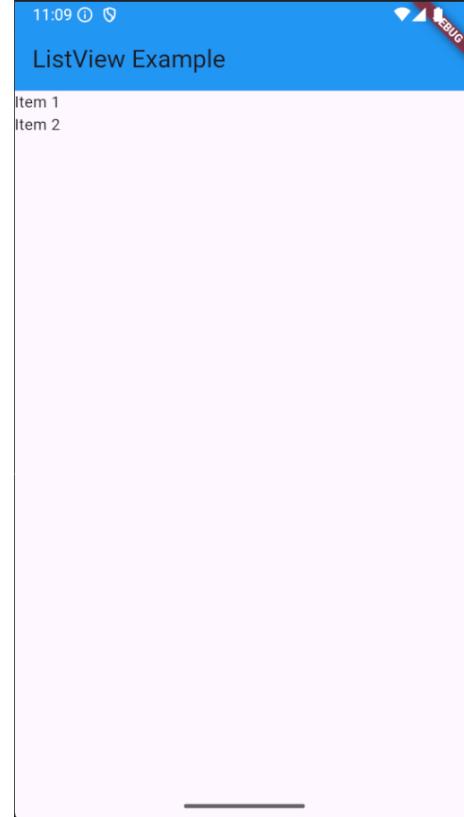
- ✓ **GridView():** Membuat grid dari daftar child yang didefinisikan secara langsung.
- ✓ **GridView.builder():** Efisien untuk daftar besar dengan lazy loading.
- ✓ **GridView.count () :** Mudah digunakan untuk grid dengan jumlah kolom tetap.

# ListView()

main.dart

Dart

```
ListView(  
  children: [  
    Text('Item 1'),  
    Text('Item 2'),  
  ],  
)
```



# ListView.builder()

main.dart

Dart

```
ListView.builder(  
  itemCount: 10,  
  itemBuilder: (context, index) {  
    return Text('Item $index');  
  },  
)
```

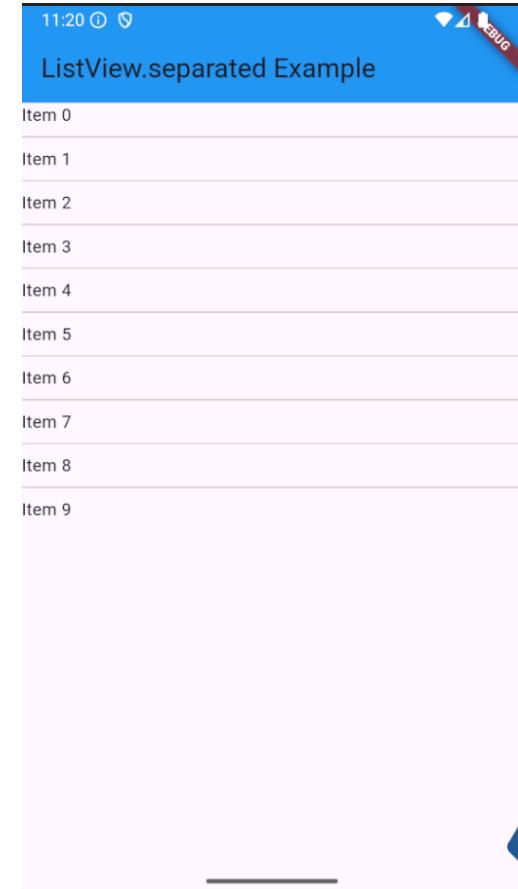


# ListView.separated()

main.dart

Dart

```
ListView.separated(  
    itemCount: 10,  
    itemBuilder: (context, index) {  
        return Text('Item $index');  
    },  
    separatorBuilder: (context, index) {  
        return Divider();  
    },  
)
```

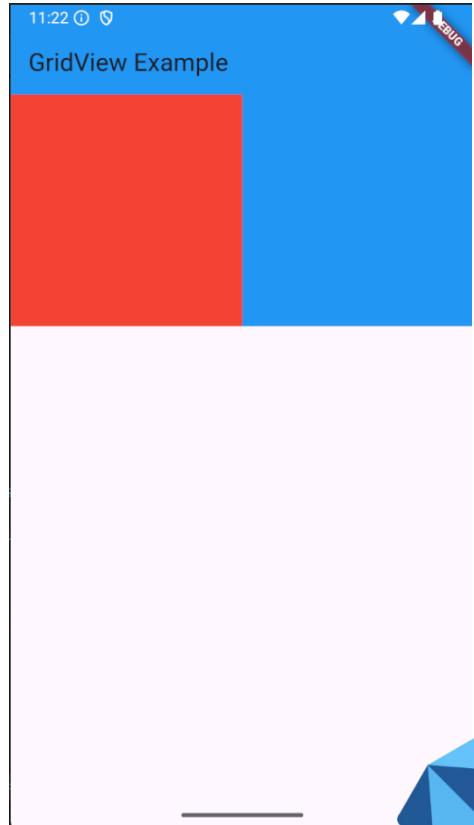


# GridView()

main.dart

Dart

```
GridView(  
  children: [  
    Container(color: Colors.red),  
    Container(color: Colors.blue),  
  ],  
  gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 2),  
)
```

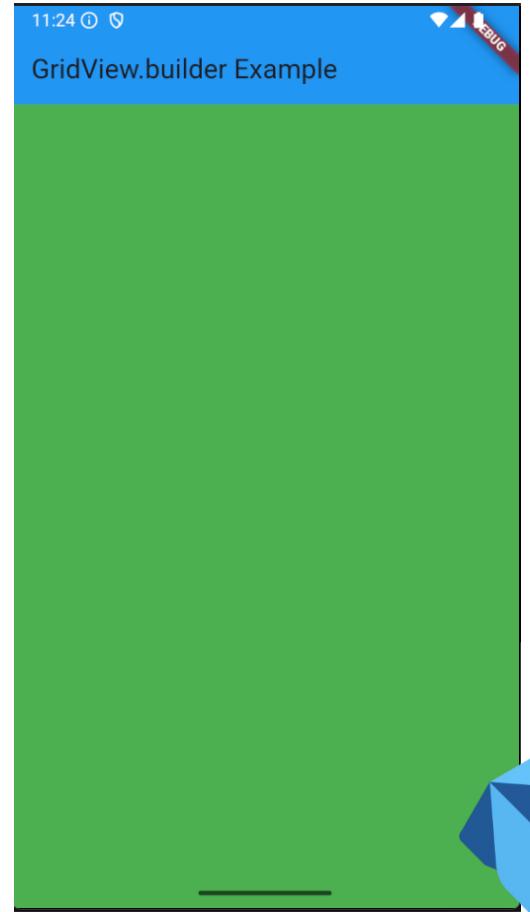


# GridView.builder()

main.dart

Dart

```
GridView.builder(  
  gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 2),  
  itemCount: 10,  
  itemBuilder: (context, index) {  
    return Container(color: Colors.green);  
  },  
)
```

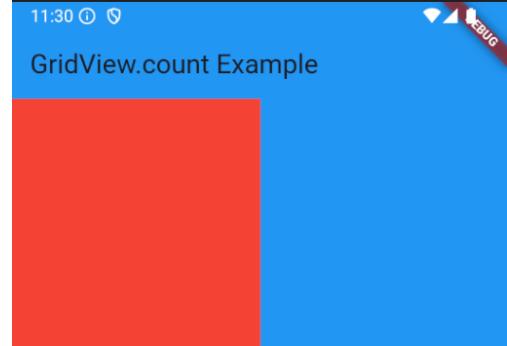


# GridView.count()

main.dart

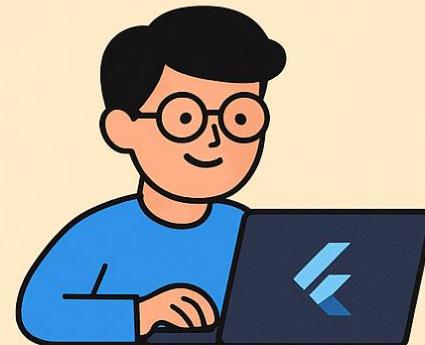
Dart

```
GridView.count(  
  crossAxisCount: 2,  
  children: [  
    Container(color: Colors.red),  
    Container(color: Colors.blue),  
  ],  
)
```



# Custom Widget dan Styling dengan Themes

## CUSTOM WIDGET



Membuat widget baru  
untuk digunakan kembali

## THEMEDATA



Mengatur tampilan aplikasi  
secara konsisten melalui  
**Theme.of(context)**

# Custom Widget dan Styling dengan Themes

main.dart

Dart

```
class MyButton extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
      style: ElevatedButton.styleFrom(
        primary: Theme.of(context).primaryColor,
      ),
      onPressed: () {},
      child: Text('Klik Saya'),
    );
  }
}
```

