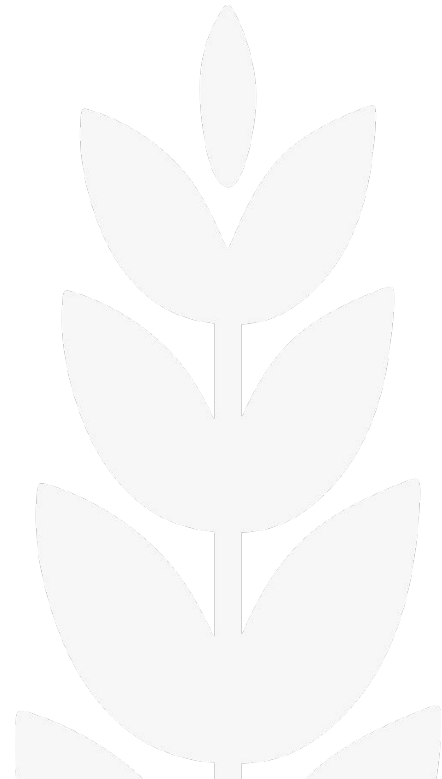


# Flutter Layout & Styling

Niki Rahmadi Wiharto



# Flutter Layout dan Styling

1. Flutter menyediakan berbagai widget untuk membuat layout dan gaya yang fleksibel.
2. Dalam materi ini, kita akan membahas beberapa teknik dasar untuk membuat tampilan yang responsif dan menarik.
3. Fokus utama: penggunaan widget layout, styling, font, icon, dan slicing dari desain Figma.



# Flexible dan Expanded Widgets

1. **Flexible:** Menyesuaikan ukuran widget di dalam baris atau kolom sesuai dengan ruang yang tersedia.
2. **Expanded:** Memperluas widget untuk mengisi ruang yang tersisa.
3. **Perbedaan Flexible vs Expanded:**  
Expanded mengambil seluruh ruang kosong, sementara Flexible menyesuaikan ruang sesuai proporsi.



# 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

1. **ListView:** Menggunakan widget untuk menampilkan daftar vertikal yang bisa di-scroll.
2. **GridView:** Menggunakan widget untuk menampilkan daftar dalam bentuk grid.
3. **Variasi ListView:**
  - a. `ListView()`: Daftar statis dengan item yang didefinisikan langsung.
  - b. `ListView.builder()`: Efisien untuk daftar panjang dengan lazy loading.
  - c. `ListView.separated()`: Sama seperti `ListView.builder()`, tetapi dengan pemisah antar item.
4. **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

1. **Custom Widget:** Membuat widget baru untuk digunakan kembali.
2. **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'),  
    );  
  }  
}
```



# Menambahkan Font

pubspec.yaml

YAML

```
fonts:  
  - family: 'Roboto'  
    fonts:  
      - asset: fonts/Roboto-Regular.ttf  
  
// Letakan file .ttf yang sudah di-download ke folder fonts
```



# Menggunakan Icon

main.dart

Dart

```
Icon(  
  Icons.home,  
  color: Colors.blue,  
)
```





# Terima Kasih