

MODUL V
ANTARMUKA PENGGUNA LANJUTAN



Disusun Oleh :
Salman Alfa R
2211104056
SE-06-02

Asisten Praktikum : Muhammad
Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia

Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS
INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO 2024

ANTARMUKA PENGGUNA LANJUTAN

Tujuan Praktikum

Mahasiswa mampu memahami konsep layout pada Flutter

Mahasiswa dapat mengimplementasikan desain user interface pada Flutter

1. ListView.builder

Widget ListView jenis ini cocok digunakan ketika memiliki data list yang lebih besar. ListView.builder membutuhkan itemBuilder dan itemCount. Parameter itemBuilder merupakan fungsi yang mengembalikan widget untuk ditampilkan. Sedangkan itemCount kita isi dengan jumlah seluruh item yang ingin ditampilkan.

Berikut contoh penerapan ListView.builder

```
final List<String> entries = <String>['A', 'B', 'C'];
final List<int> colorCodes = <int>[600, 500, 100];

ListView.builder(
  padding: const EdgeInsets.all(8),
  itemCount: entries.length,
  itemBuilder: (BuildContext context, int index) {
    return Container(
      height: 50,
      color: Colors.amber[colorCodes[index]],
      child: Center(
        child: Text('Entry ${entries[index]}'),
      ),
    );
  },
);
```

Output kode:

entry A

entry B

entry C

entry D

2. ListView.Separated

ListView jenis ini akan menampilkan daftar item yang dipisahkan dengan separator. Penggunaan ListView.separated mirip dengan builder, yang membedakan adalah terdapat satu parameter tambahan wajib yaitu separatorBuilder yang mengembalikan Widget yang akan berperan sebagai separator.

Berikut contoh penerapan ListView.Separated

```
final List<String> entries = <String>['A', 'B', 'C'];
final List<int> colorCodes = <int>[600, 500, 100];

ListView.separated(
  padding: const EdgeInsets.all(8),
  itemCount: entries.length,
  itemBuilder: (BuildContext context, int index) {
    return Container(
      height: 50,
      color: Colors.amber[colorCodes[index]],
      child: Center(child: Text('Entry ${entries[index]}')),
    );
  },
  separatorBuilder: (BuildContext context, int index) => const
  Divider(),
);
```

Output kode:

entry A

entry B

entry C

entry D

3. Flexible dan Expanded

a) Flexible

Flexible digunakan ketika Anda ingin memberikan ruang fleksibel kepada widget di dalam kolom atau baris. Dengan Flexible, widget dapat mengambil ruang yang tersisa di dalam layout, tetapi tetap memiliki batas maksimal yang disesuaikan dengan kebutuhan ruangnya.

b) Expanded

Expanded adalah turunan dari Flexible yang secara otomatis mengisi semua ruang yang tersisa di dalam kolom atau baris. Ketika Anda menggunakan Expanded, widget tersebut akan mengambil ruang sebanyak mungkin tanpa memperhatikan kebutuhan ruang minimum.

Berikut contoh penerapan Flexible dan Expanded:

```
Column(  
  children: [  
  
    //Flexible  
    Row(  
      children: <Widget>[  
        Container(  
          width: 50,  
          height: 100,  
          color: Colors.red,
```



```

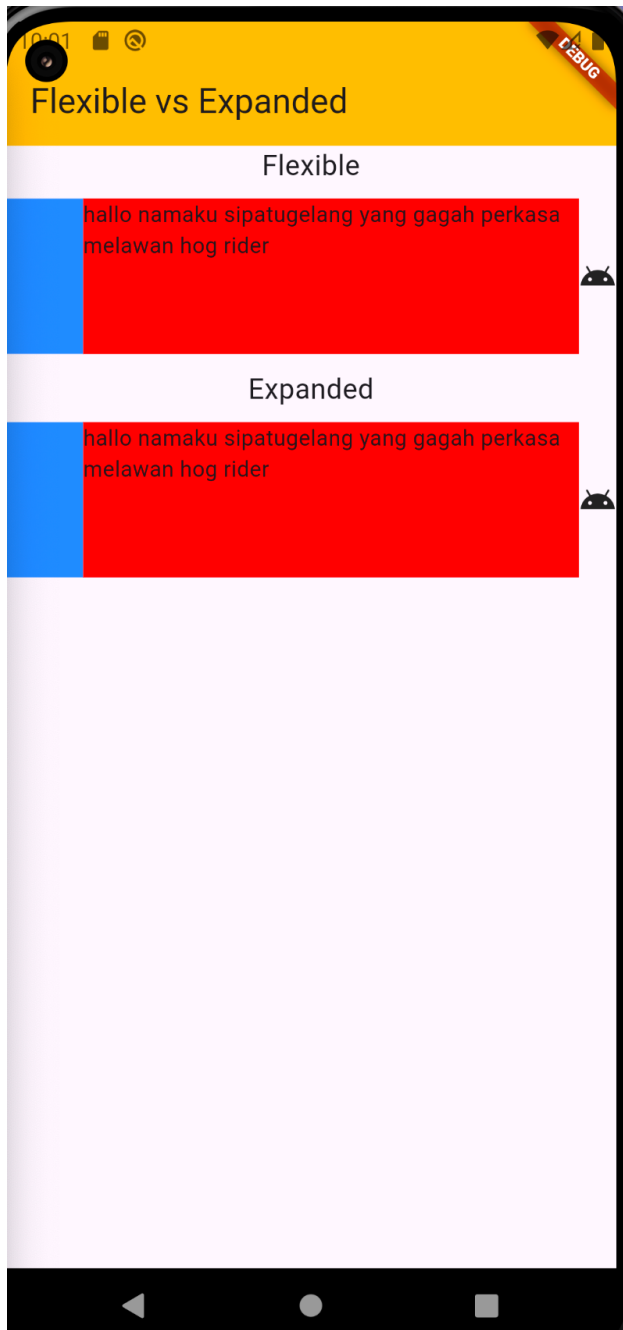
    ),
    Flexible(
      child: Container(
        height: 100,
        color: Colors.green,
        child: const Text(
          "Flexible takes up the remaining space but
can shrink if needed.",
        ),
      ),
    ),
    const Icon(Icons.sentiment_very_satisfied),
  ],
),
 SizedBox(height: 20),

//Expanded
Row(
  children: <Widget>[
    Container(
      width: 50,
      height: 100,
      color: Colors.red,
    ),
    Expanded(
      child: Container(
        height: 100,
        color: Colors.green,
        child: const Text(
          "Expanded forces the widget to take up all
the remaining space.",
        ),
      ),
    ),
    const Icon(Icons.sentiment_very_satisfied),
  ],
),
],

```

),

Output kode:



Note: terlihat sama tapi ketika dijalankan di emu chrome expanded bisa panjang mengikuti TAB

4. CustomScrollView

Widget ini memungkinkan membuat efek pada list, grid, maupun header yang lebar. Misalnya, ketika ingin membuat scroll view yang berisi app bar yang lebar yang meliputi list dan grid secara bersamaan, maka bisa menggunakan 3 widget sliver, yaitu SliverAppBar, SliverList, dan SliverGrid.

Berikut adalah barisan kode yang menunjukkan scroll view dengan meliputi app bar yang fleksibel, grid dan infinite list.

```
CustomScrollView(  
  slivers: <Widget>[  
    //Sliver App Bar  
    const SliverAppBar(  
      pinned: true,  
      expandedHeight: 250.0,  
      flexibleSpace: FlexibleSpaceBar(  
        title: Text('Demo'),  
      ),  
    ),  
  
    //Sliver Grid  
    SliverGrid(  
      gridDelegate: const
```

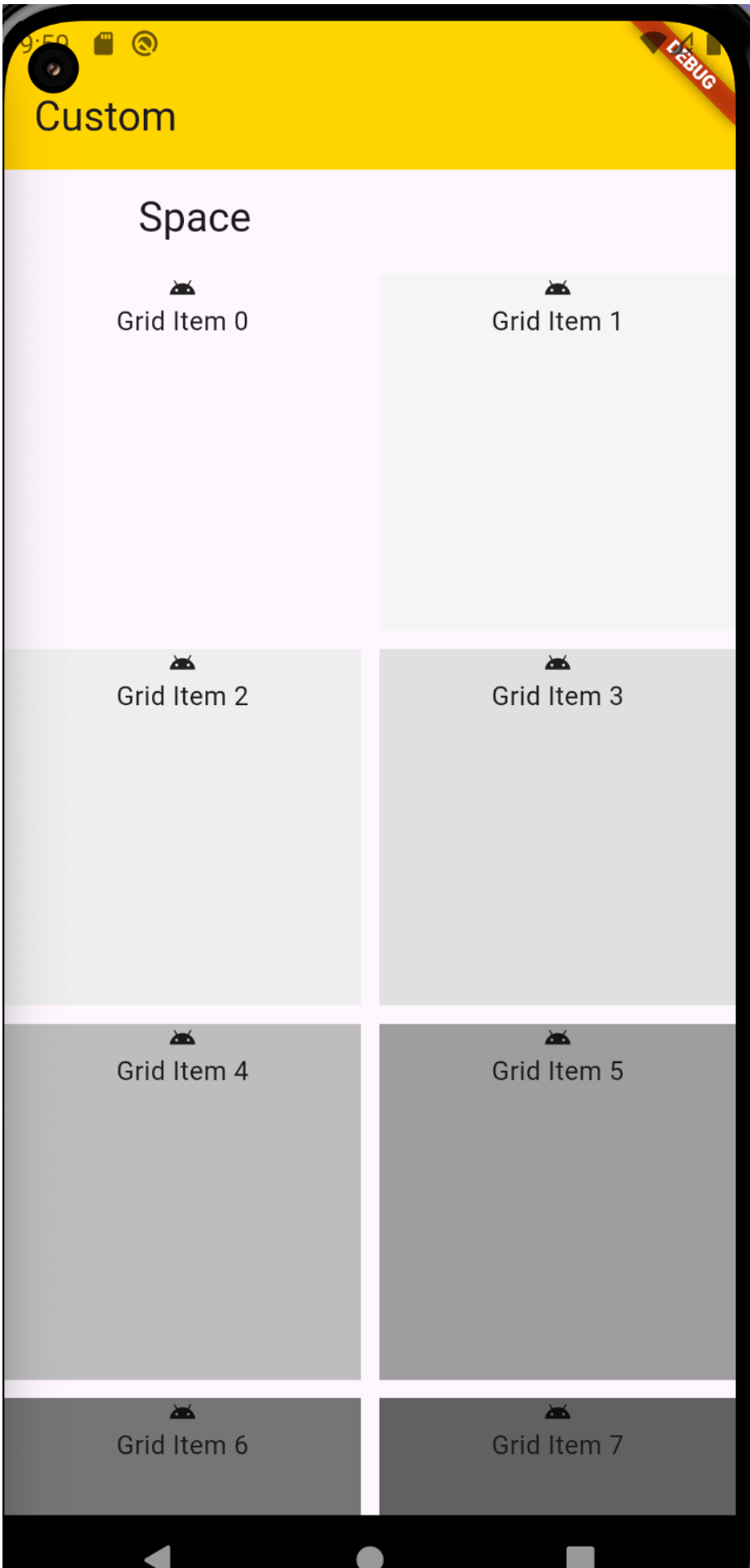
```

SliverGridDelegateWithMaxCrossAxisExtent(
  maxCrossAxisExtent: 200.0,
  mainAxisSpacing: 10.0,
  crossAxisSpacing: 10.0,
  childAspectRatio: 4.0,
),
delegate: SliverChildBuilderDelegate(
  (BuildContext context, int index) {
    return Container(
      alignment: Alignment.center,
      color: Colors.teal[100 * (index % 9)],
      child: Text('Grid Item $index'),
    );
  },
  childCount: 20,
),

//Sliver Fixed List
SliverFixedExtentList(
  itemExtent: 50.0,
  delegate: SliverChildBuilderDelegate(
    (BuildContext context, int index) {
      return Container(
        alignment: Alignment.center,
        color: Colors.lightBlue[100 * (index % 9)],
        child: Text('List Item $index'),
      );
    },
  ),
),
],
),

```

Output kode:



Tugas Mandiri (Unguided)

1. Modifikasi project Rekomendasi Wisata pada Tugas Unguided 04 modul Antarmuka Pengguna dengan mengimplementasikan widget CustomScrollView, SliverAppBar, dan SliverList untuk merekomendasikan beberapa tempat wisata yang ada di Banyumas disertai foto, nama wisata, dan deskripsi singkat! (buatlah se kreatif mungkin).

Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program.