

Layout

TI2137 - Pengembangan Aplikasi Mobil Front-End



Team Teaching

- Sunaryo Winardi, S.Kom., M.Tl.
- Sio Jurnal Pipin, S.Kom., M.Kom.
- M. Taufiq Hidayat Pohan, S.Kom.
- Richy Rotuahta Saragih, S.Kom.



Context

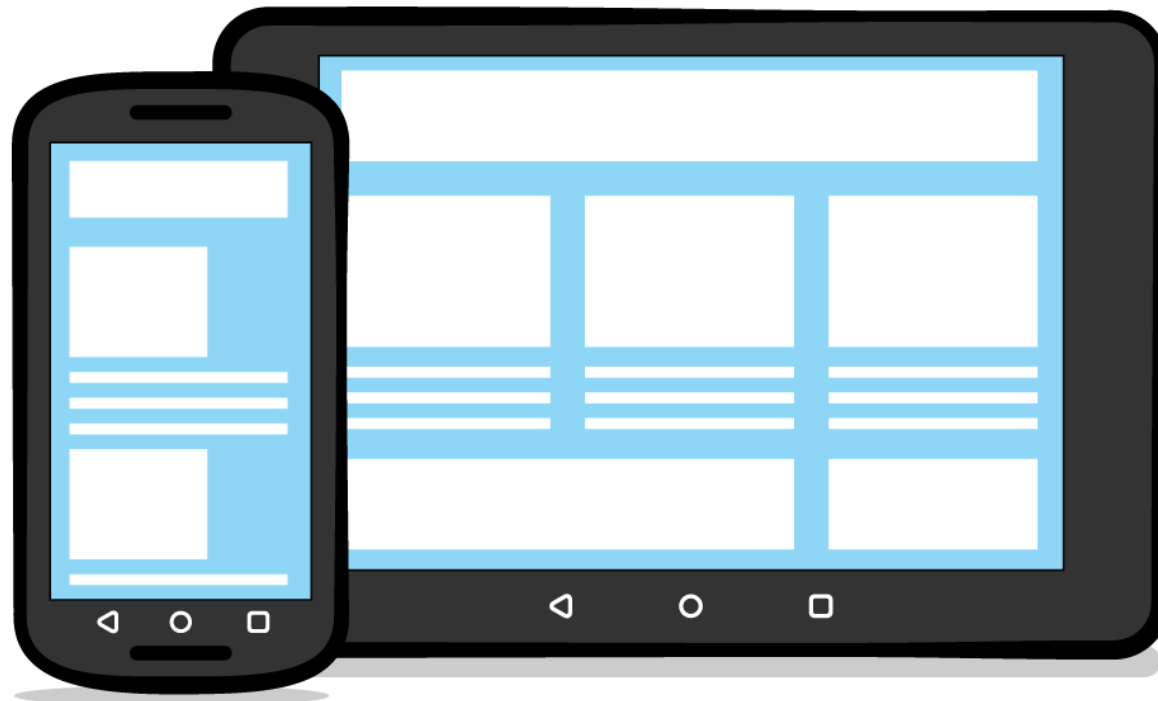
- Single-child layout
 - Container
 - Expanded
- Multi-child layout
 - Column
 - Row
 - GridView
- Navigation



Apa itu Layout?

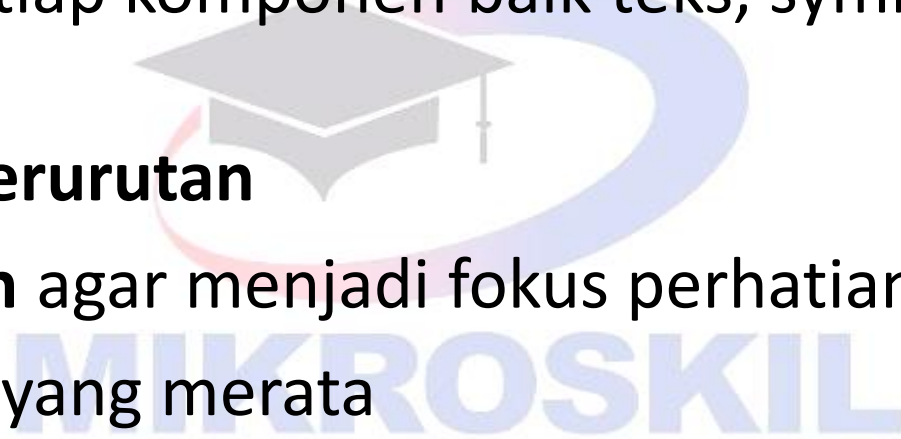


Layout

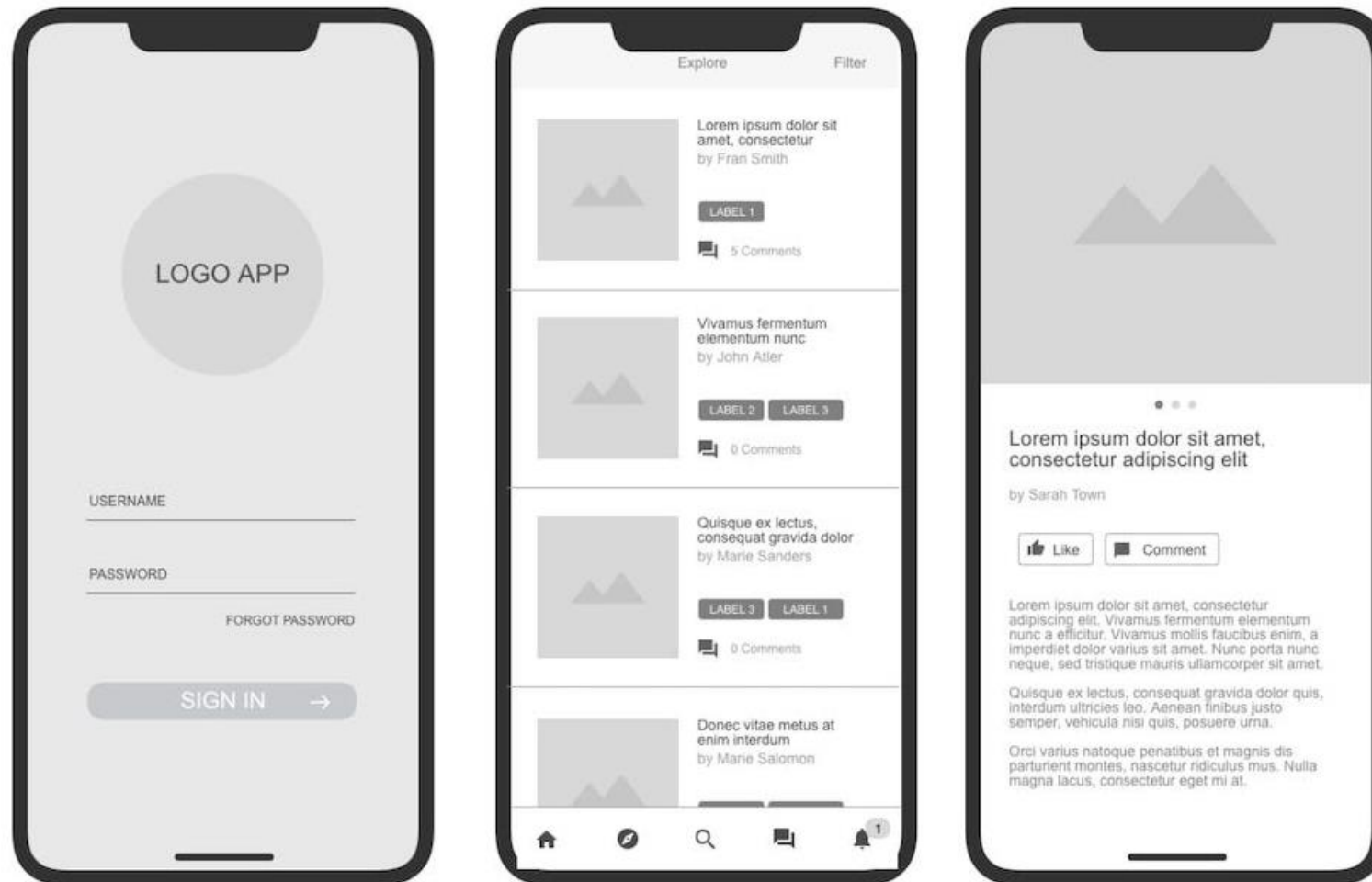


Layout adalah...

- **Tata Letak**
- Pengaturan letak setiap komponen baik teks, symbol (icon), gambar maupun video.
- Setiap komponen **berurutan**
- Memiliki **penekanan** agar menjadi fokus perhatian
- **Pembagian** elemen yang merata
- Memiliki **kesatuan** baik warna, gaya huruf, dan gaya elemen lainnya



Layout itu seperti...



Layout pada Flutter

- Single-child Layout

- Align
- Center
- Container
- Expanded
- Padding
- Dan lainnya...

- Multi-child Layout

- Column
- Row
- GridView
- ListView
- Stack
- Dan lainnya...



Single-child Layout

- Hanya terdapat **1 widget** pada child
- Sangat cocok untuk meletakkan element widget yang hanya terdiri dari **1 elemen** seperti Button, Label atau Text tunggal
- Terdiri dari :
 - Align
 - Center
 - Container
 - Expanded
 - OverflowBox
 - Padding
- Lebih banyak? [Lihat disini](#)

Container

Container merupakan salah satu **parent widget (induk)** yang memungkinkan user untuk melakukan dekorasi seperti border, warna background, atau transform.



Container

```
Center(  
  child: Container(  
    margin: const EdgeInsets.all(10.0),  
    color: Colors.amber[600],  
    width: 48.0,  
    height: 48.0,  
  ),  
)
```



Container

```
Container(  
  constraints: BoxConstraints.expand(  
    height: Theme.of(context).textTheme.headline4!.fontSize! * 1.1 + 200.0,  
  ),  
  padding: const EdgeInsets.all(8.0),  
  color: Colors.blue[600],  
  alignment: Alignment.center,  
  child: Text('Hello World',  
    style: Theme.of(context)  
      .textTheme  
      .headline4!  
      .copyWith(color: Colors.white)),  
  transform: Matrix4.rotationZ(0.1),  
)
```

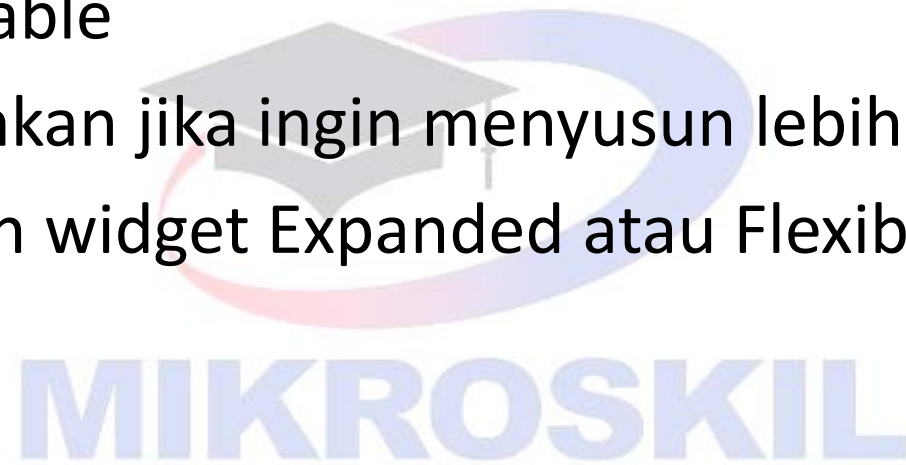
Multi-child Layout

- Dapat menyusun **lebih dari 1 widget** pada children
- Properti children bertipe data array atau list
- Sering terjadi **overflow pixel**
- Terdiri dari :
 - Column
 - Row
 - Stack
 - Flex
 - ListView
- Lebih banyak? [Lihat disini](#)



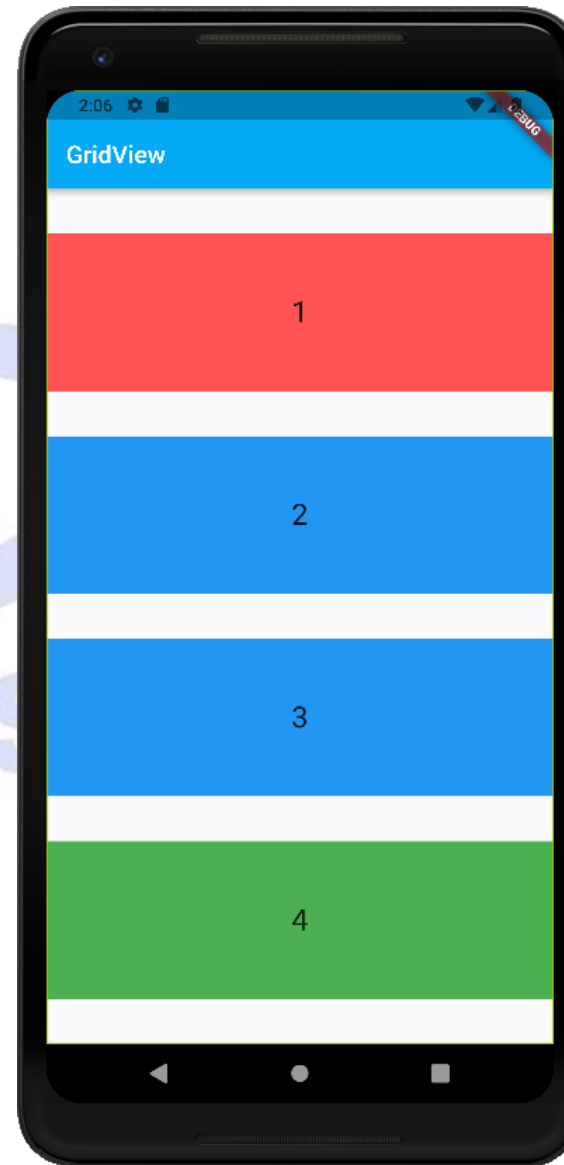
Column

- Untuk menyusun children widget secara vertical
- Tidak bersifat scrollable
- Penggunaan disarankan jika ingin menyusun lebih dari 1 widget
- Dapat menggunakan widget Expanded atau Flexible agar bersifat flex



Column

```
Container(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      Container(
        padding: EdgeInsets.symmetric(vertical: 50),
        color: Colors.redAccent,
        child: const Center(
          child: Text("1", style: TextStyle(fontSize: 24.0)),
        ),
      ),
      Container(
        padding: EdgeInsets.symmetric(vertical: 50),
        color: Colors.blue,
        child: const Center(
          child: Text("2", style: TextStyle(fontSize: 24.0)),
        ),
      ),
      Container(
        padding: EdgeInsets.symmetric(vertical: 50),
        color: Colors.blue,
        child: const Center(
          child: Text("3", style: TextStyle(fontSize: 24.0)),
        ),
      ),
      Container(
        padding: EdgeInsets.symmetric(vertical: 50),
        color: Colors.green,
        child: const Center(
          child: Text("4", style: TextStyle(fontSize: 24.0)),
        ),
      ),
    ],
  ),
)
```



Aligment Column

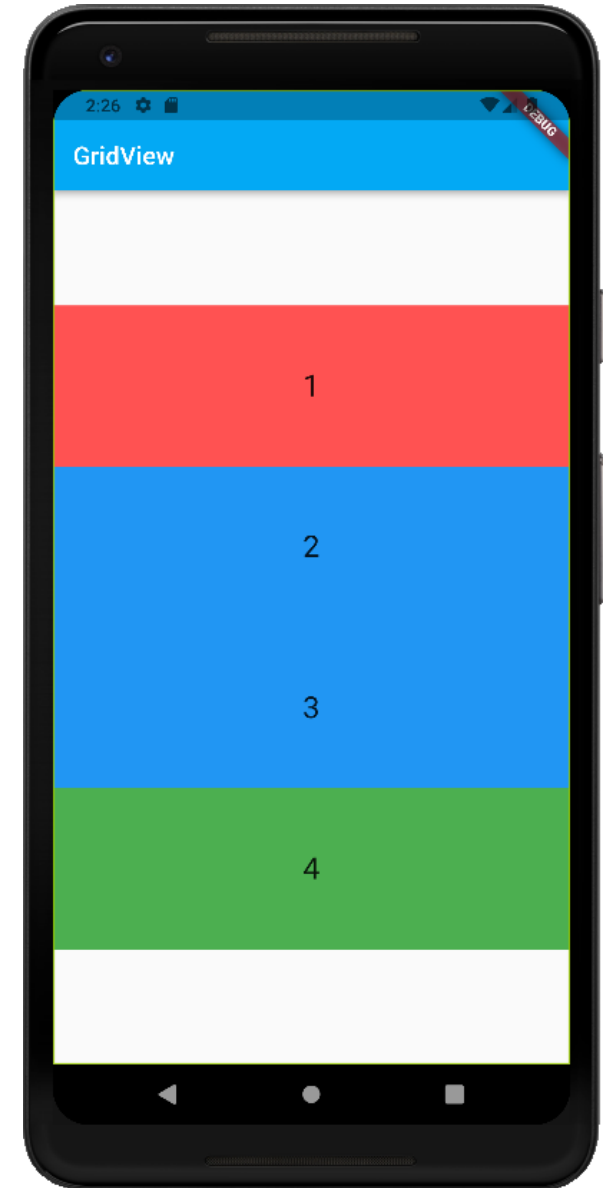
- Terdapat beberapa jenis Aligment untuk memudahkan penyusunan layout pada Column, yaitu :
 - **center**, seluruh children berada tepat di tengah widget
 - **end**, seluruh children berada dibagian akhir (bawah) widget
 - **start**, seluruh children berada dibagian awal (atas) widget
 - **spaceAround**, menempatkan children di ruang kosong secara merata sebanyak setengah dari ruang tersebut dari child pertama hingga akhir
 - **spaceBetween**, menempatkan children secara merata, namun anak pertama dan terakhir tidak memiliki ruang kosong pada atas atau bawahnya
 - **spaceEvenly**, menempatkan children secara merata beserta seluruh ruang kosongnya

Alignment Column

Center

seluruh children berada tepat di tengah widget

```
Column(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ],  
)
```

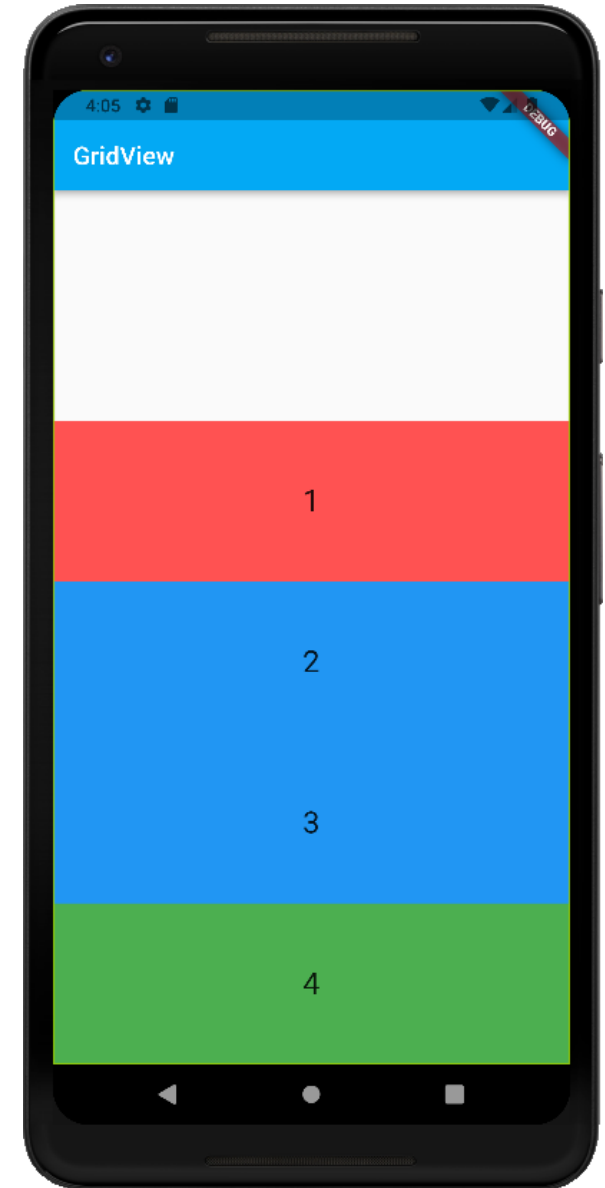


Alignment Column

End

seluruh children berada dibagian akhir
(bawah) widget

```
Column(  
  mainAxisAlignment: MainAxisAlignment.end,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

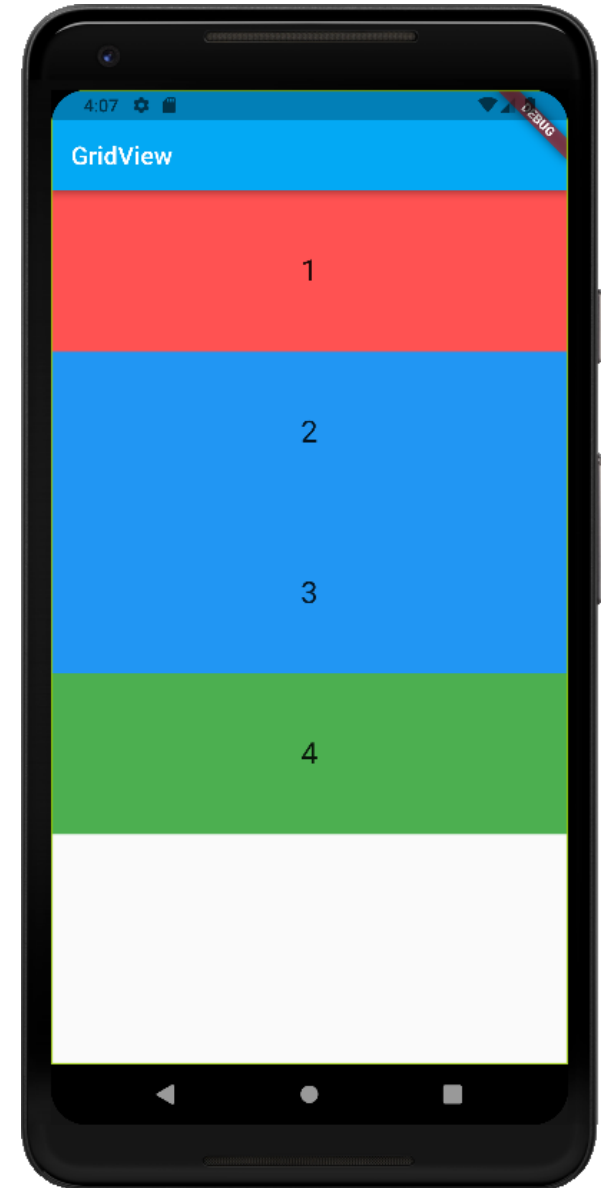


Alignment Column

Start

seluruh children berada dibagian awal (atas) widget

```
Column(  
  mainAxisAlignment: MainAxisAlignment.start,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ],  
)
```

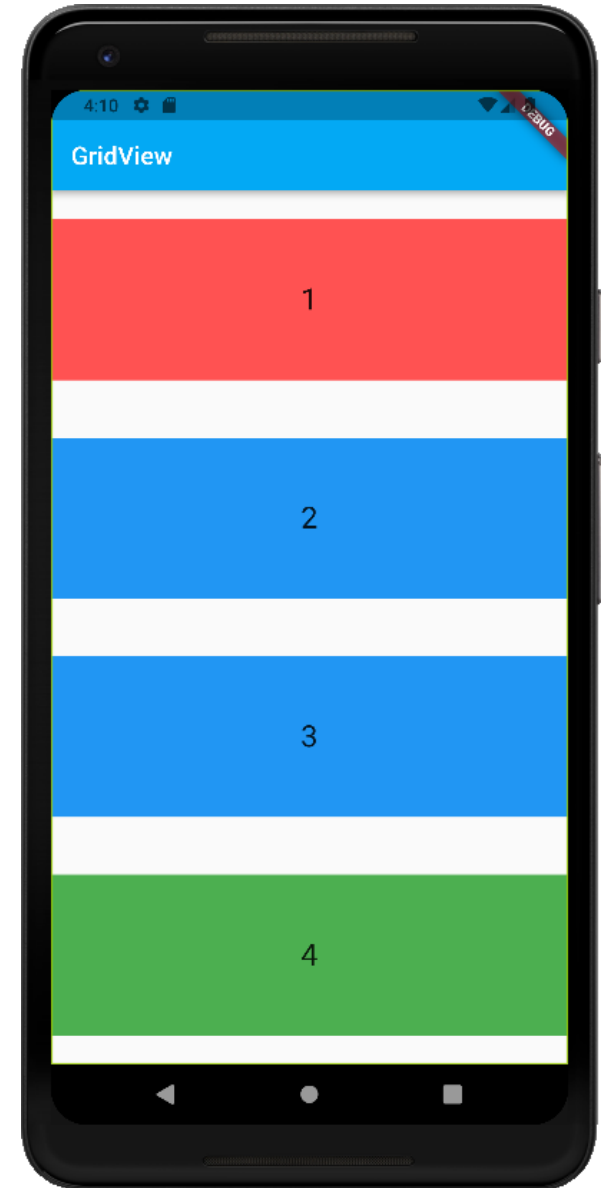


Alignment Column

spaceAround

menempatkan children di ruang kosong secara merata sebanyak setengah dari ruang tersebut dari child pertama hingga akhir

```
Column(  
  mainAxisAlignment: MainAxisAlignment.spaceAround,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

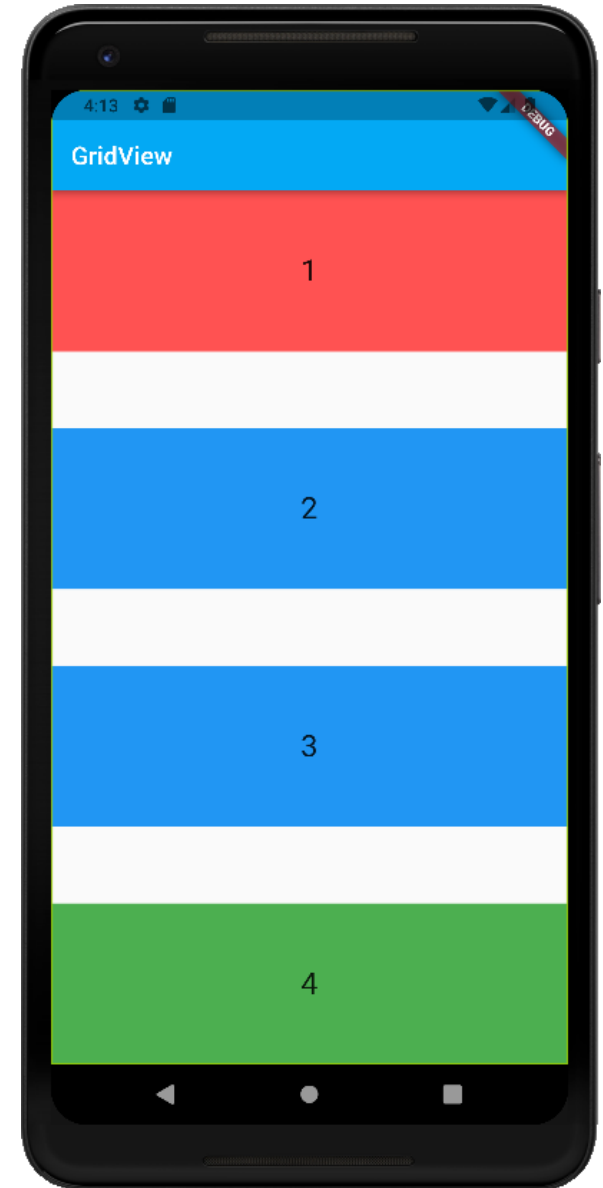


Alignment Column

spaceBetween

menempatkan children secara merata, namun anak pertama dan terakhir tidak memiliki ruang kosong pada atas atau bawahnya

```
Column(  
  mainAxisAlignment: MainAxisAlignment.spaceBetween,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ],  
)
```

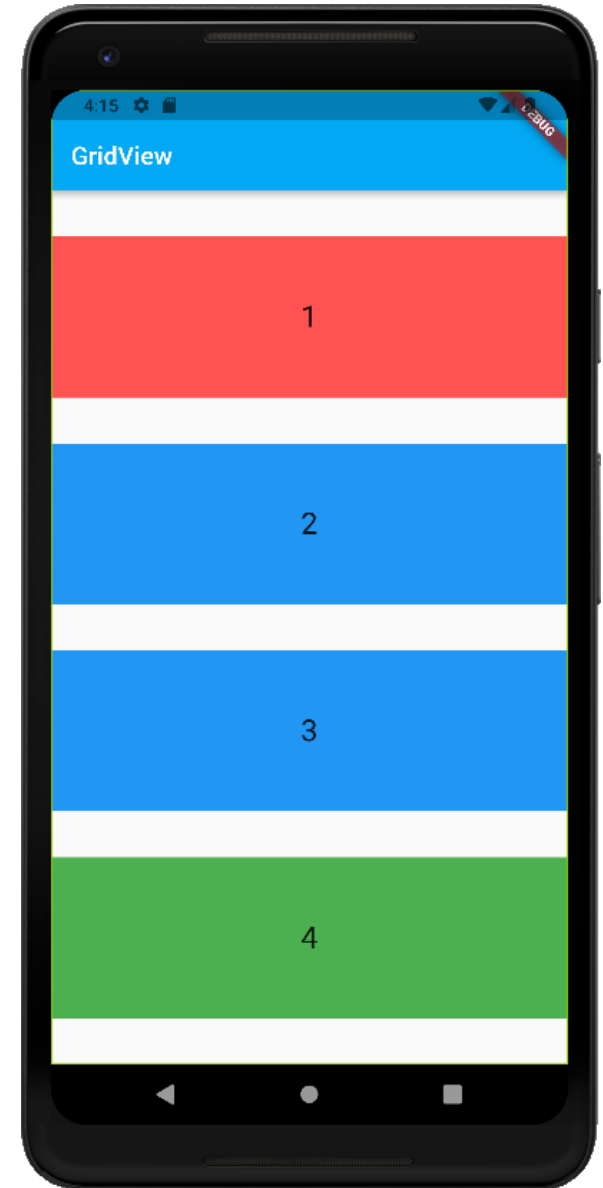


Alignment Column

spaceEvenly

menempatkan children secara merata beserta seluruh ruang kosongnya

```
Column(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ],  
)
```



Row

- Untuk Menyusun widget secara horizontal
- Seperti Column, Row tidak dapat di scroll
- Jika memiliki beberapa widget yang disusun secara horizontal, tapi ingin dapat di scroll jika tidak memiliki cukup ruang, gunakanlah ListView.

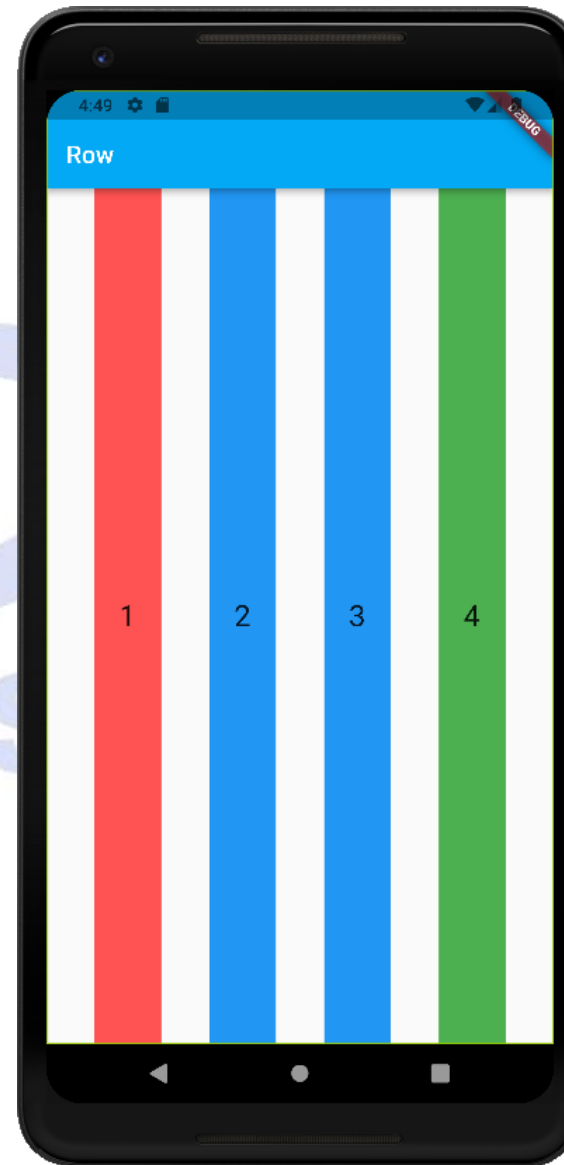
MIKROSKIL

Contoh Row

```
Row(  
  children: [  
    Container(  
      color: Colors.orange,  
      child: FlutterLogo(  
        size: 60.0,  
      ),  
    ),  
    Container(  
      color: Colors.blue,  
      child: FlutterLogo(  
        size: 60.0,  
      ),  
    ),  
    Container(  
      color: Colors.purple,  
      child: FlutterLogo(  
        size: 60.0,  
      ),  
    ),  
  ],  
)
```


Column

```
Container(
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      Container(
        padding: EdgeInsets.symmetric(horizontal: 20),
        color: Colors.redAccent,
        child: const Center(
          child: Text("1", style: TextStyle(fontSize: 24.0)),
        ),
      ),
      Container(
        padding: EdgeInsets.symmetric(horizontal: 20),
        color: Colors.blue,
        child: const Center(
          child: Text("2", style: TextStyle(fontSize: 24.0)),
        ),
      ),
      Container(
        padding: EdgeInsets.symmetric(horizontal: 20),
        color: Colors.blue,
        child: const Center(
          child: Text("3", style: TextStyle(fontSize: 24.0)),
        ),
      ),
      Container(
        padding: EdgeInsets.symmetric(horizontal: 20),
        color: Colors.green,
        child: const Center(
          child: Text("4", style: TextStyle(fontSize: 24.0)),
        ),
      ),
    ],
  ),
)
```



Alignment Row

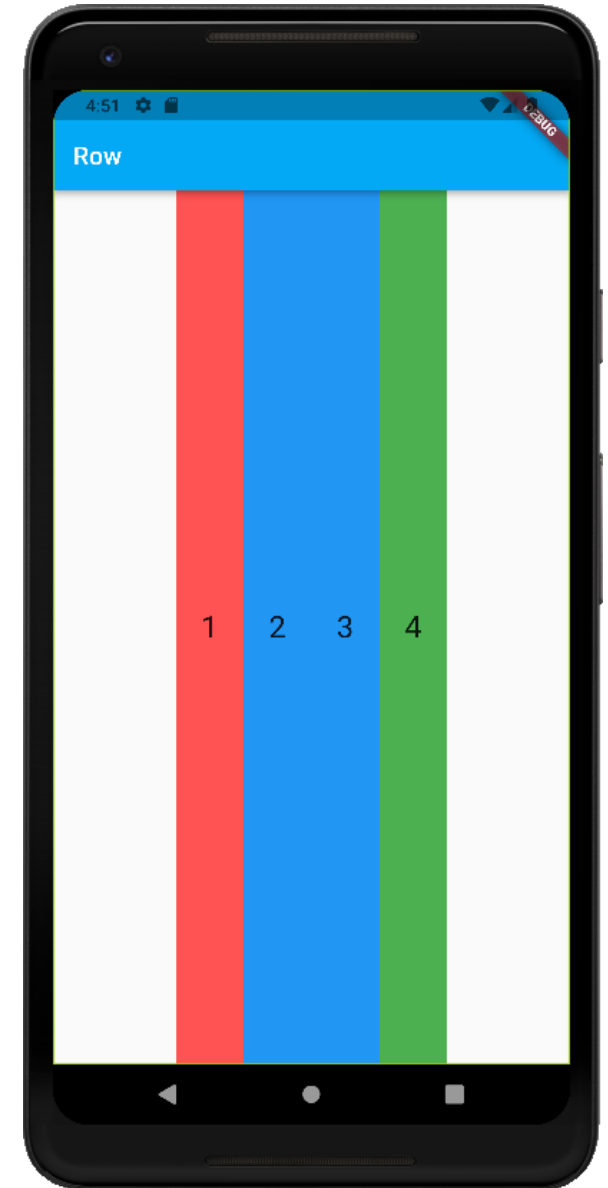
- Seperti pada Column, terdapat beberapa jenis Alignment untuk memudahkan penyusunan layout, yaitu :
 - **center**, seluruh children berada tepat di tengah widget
 - **end**, seluruh children berada dibagian akhir (bawah) widget
 - **start**, seluruh children berada dibagian awal (atas) widget
 - **spaceAround**, menempatkan children di ruang kosong secara merata sebanyak setengah dari ruang tersebut dari child pertama hingga akhir
 - **spaceBetween**, menempatkan children secara merata, namun anak pertama dan terakhir tidak memiliki ruang kosong pada atas atau bawahnya
 - **spaceEvenly**, menempatkan children secara merata beserta seluruh ruang kosongnya

Alignment Row

Center

seluruh children berada tepat di tengah widget

```
Row(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

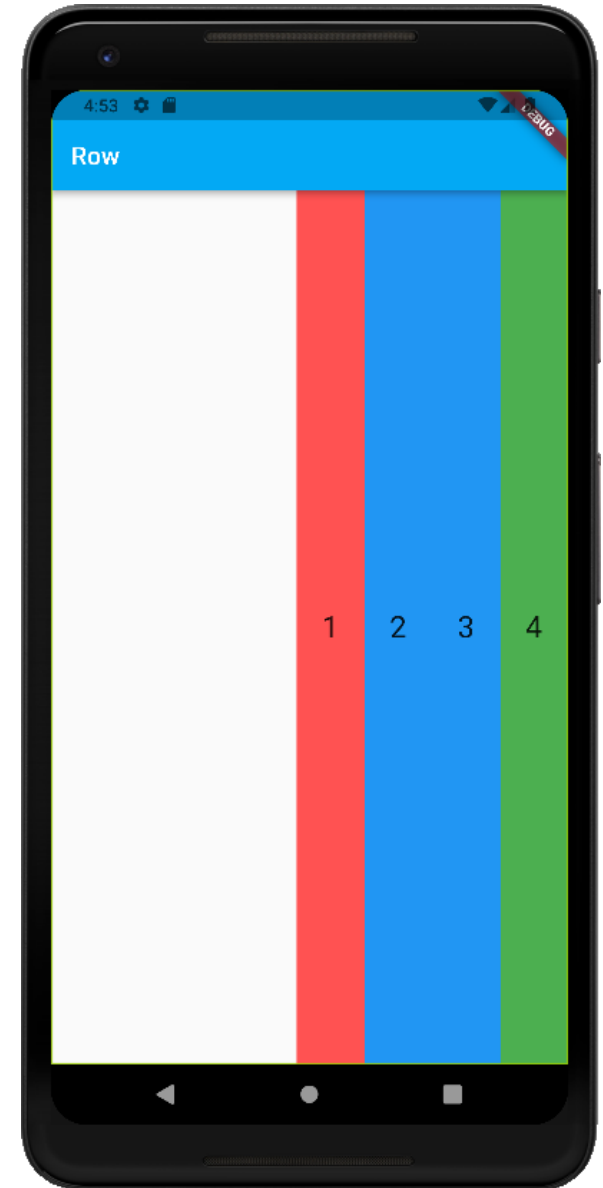


Alignment Row

End

seluruh children berada dibagian akhir (kanan) widget

```
Row(  
  mainAxisAlignment: MainAxisAlignment.end,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

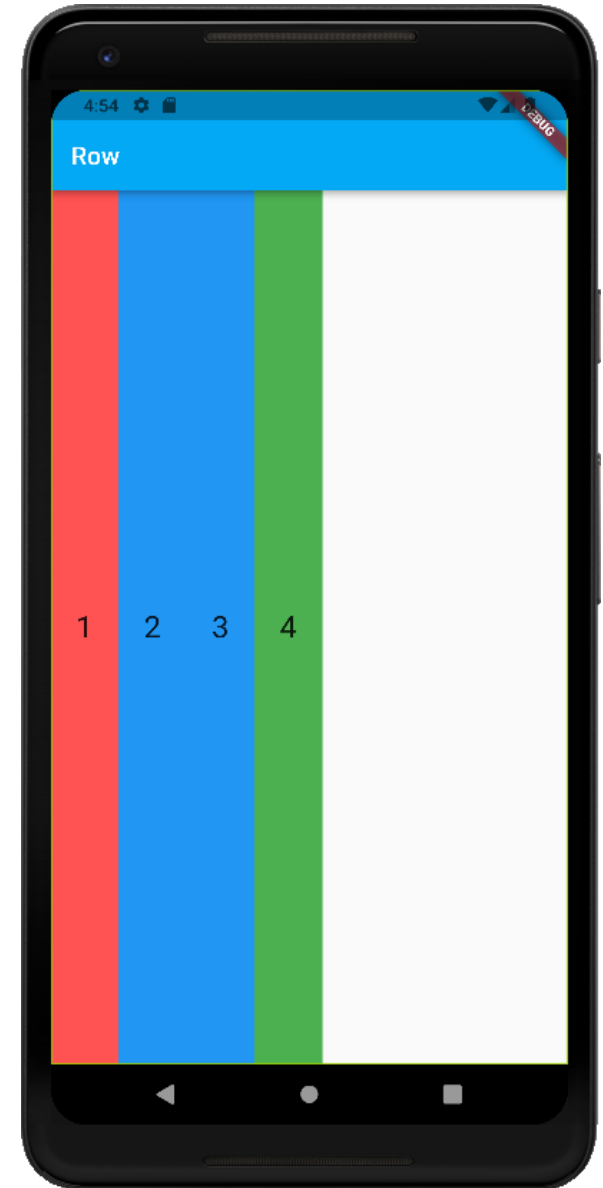


Alignment Row

Start

seluruh children berada dibagian awal (kiri) widget

```
Row(  
  mainAxisAlignment: MainAxisAlignment.start,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

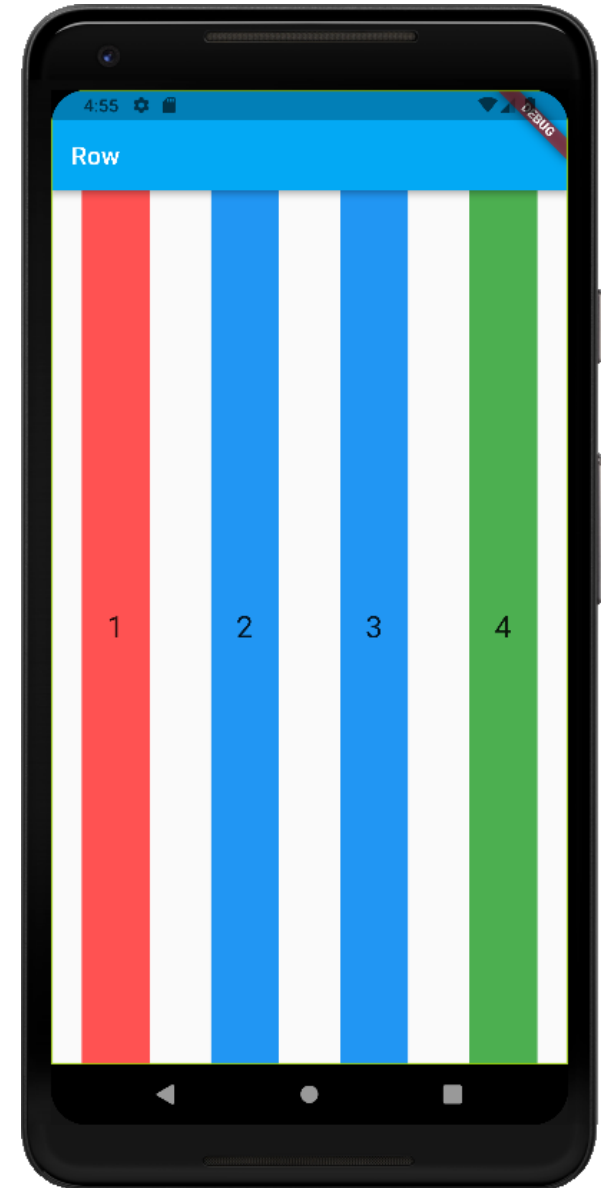


Alignment Row

spaceAround

menempatkan children di ruang kosong secara merata sebanyak setengah dari ruang tersebut dari child pertama hingga akhir

```
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceAround,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

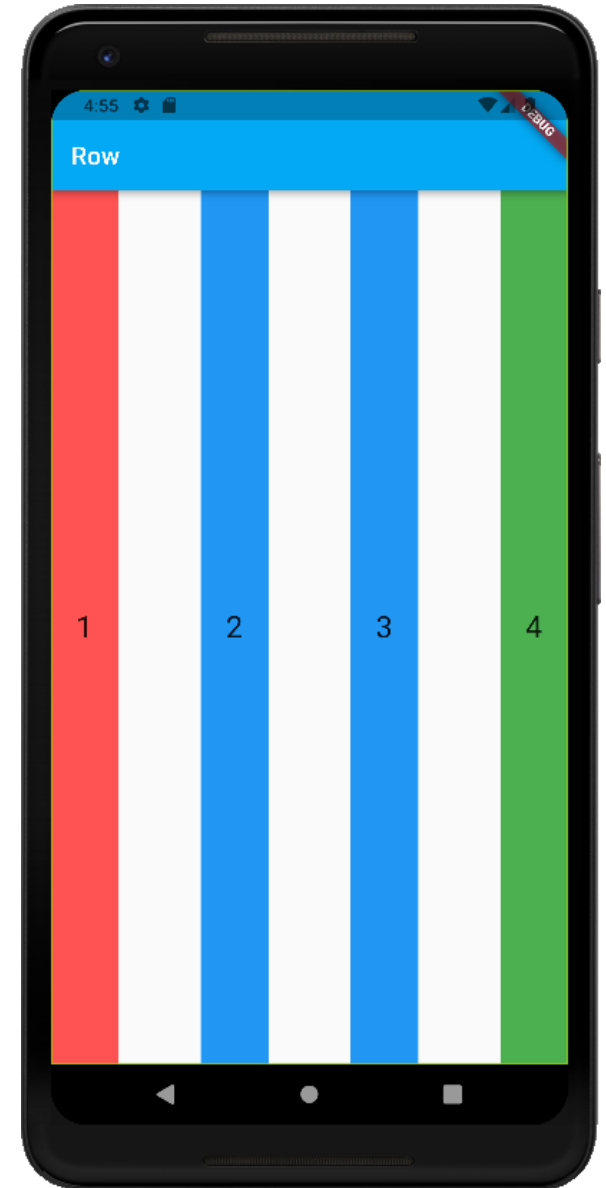


Alignment Row

spaceBetween

menempatkan children secara merata, namun anak pertama dan terakhir tidak memiliki ruang kosong pada atas atau bawahnya

```
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceBetween,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

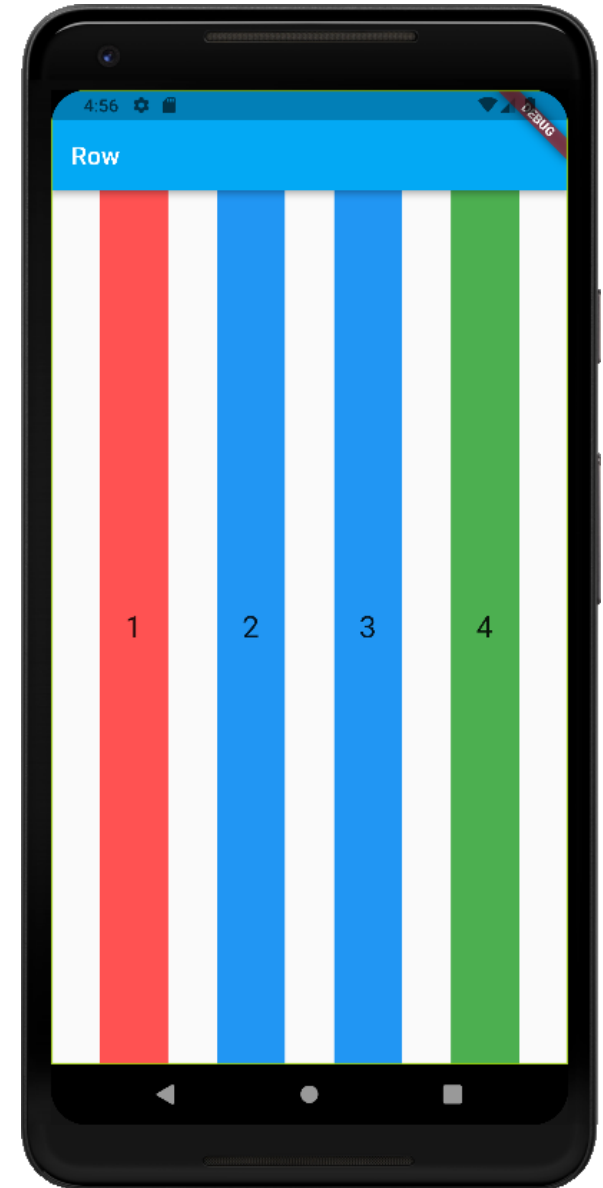


Alignment Row

spaceEvenly

menempatkan children secara merata beserta seluruh ruang kosongnya

```
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  children: [  
    Container(  
      padding: EdgeInsets.symmetric(vertical: 50),  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    ...  
    ...  
  ]  
)
```

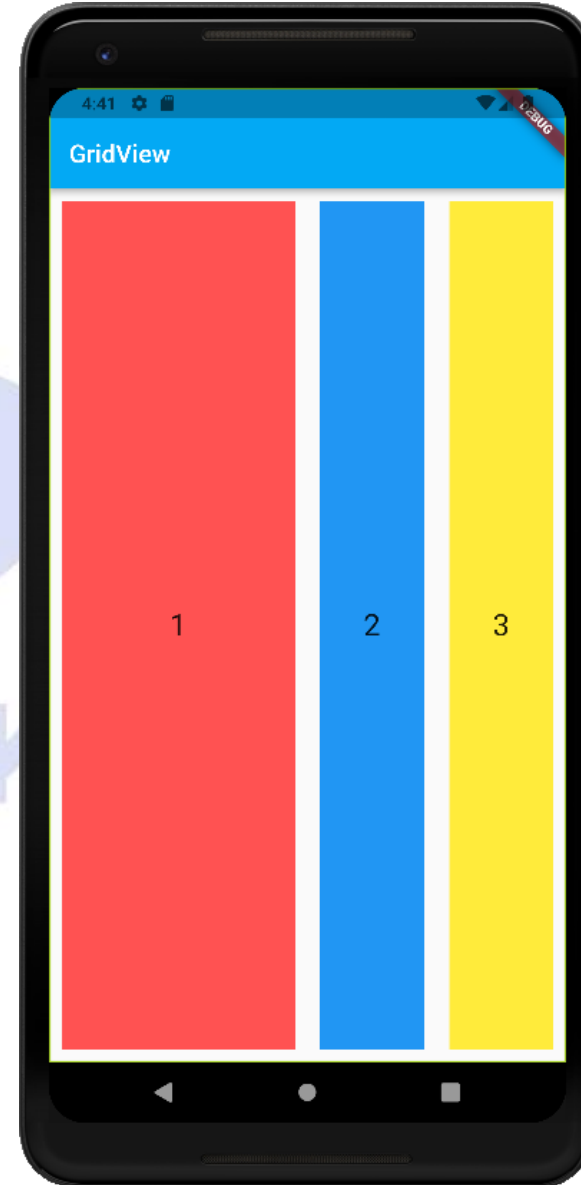


Expanded

- Hanya dapat digunakan sebagai pelengkap pada widget Row, Column dan Flex
- Harus berada pada Row, Column atau Flex yang terdapat di dalam Statefull atau Stateless widget
- Berfungsi untuk membuat sebuah child mengekspansi dan mengisi ruang kosong widget secara vertical atau horizontal (sesuai widget parentnya)
- Lebih dari 1 child yang menggunakan widget expanded, akan terjadi pembagian secara flex

Expanded

```
Container(
  child: Row(
    children: [
      Expanded(
        flex: 6,
        child: Container(
          margin: EdgeInsets.symmetric(vertical: 10, horizontal: 10),
          padding: EdgeInsets.symmetric(vertical: 50),
          color: Colors.redAccent,
          child: const Center(
            child: Text("1", style: TextStyle(fontSize: 24.0)),
          ),
        ),
      ),
      Expanded(
        flex: 3,
        child: Container(
          margin: EdgeInsets.symmetric(vertical: 10, horizontal: 10),
          padding: EdgeInsets.symmetric(vertical: 50),
          color: Colors.blue,
          child: const Center(
            child: Text("2", style: TextStyle(fontSize: 24.0)),
          ),
        ),
      ),
      Expanded(
        flex: 3,
        child: Container(
          margin: EdgeInsets.symmetric(vertical: 10, horizontal: 10),
          padding: EdgeInsets.symmetric(vertical: 50),
          color: Colors.yellow,
          child: const Center(
            child: Text("3", style: TextStyle(fontSize: 24.0)),
          ),
        ),
      ),
    ],
  ),
)
```



GridView

- Merupakan widget multi-child
- Secara default bersifat scrollable
- Berfungsi untuk membuat tampilan menjadi 2 dimensi (horizontal dan vertikal)



GridView

- Terdapat beberapa jenis GridView, seperti berikut :
 - GridView
 - GridView.count
 - GridView.builder
 - GridView.custom
 - GridView.extent



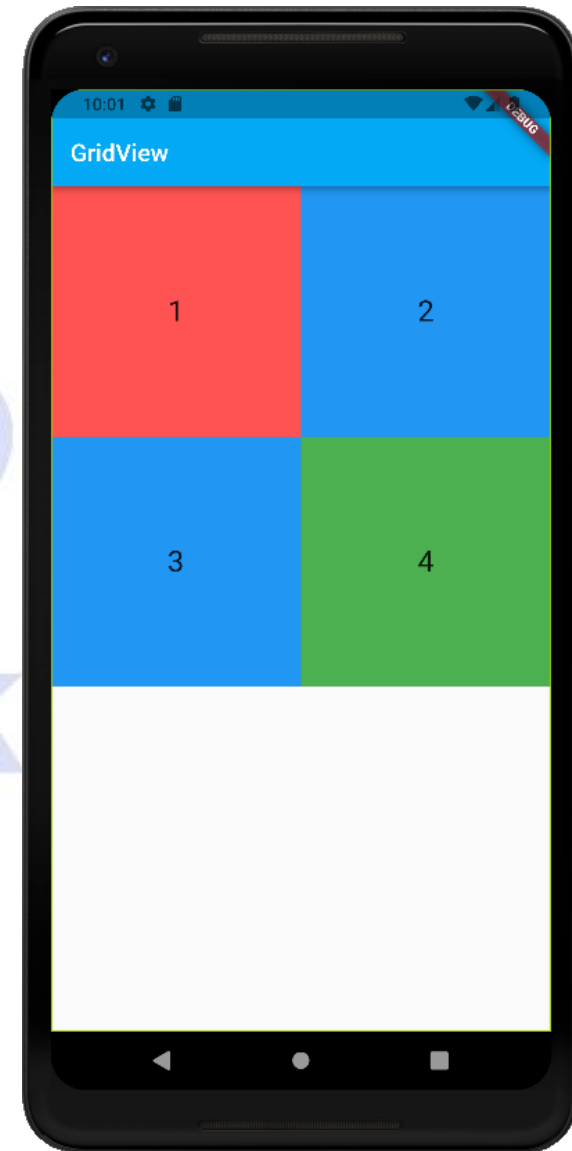
Gridview()

- Merupakan gabungan antara GridView.count dan GridView.builder
- Menentukan jumlah kolom menggunakan gridDelegate
- Seluruh widget pada children akan ditampilkan



GridView()

```
GridView(  
  gridDelegate : SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 2),  
  children: [  
    Container(  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.blue,  
      child: const Center(  
        child: Text("2", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.blue,  
      child: const Center(  
        child: Text("3", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.green,  
      child: const Center(  
        child: Text("4", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
  ],  
)
```



Properti GridView()

Properties	Fungsi
scrollDirection	Mengubah arah scrolling pada GridView
Reverse	Membalikkan susunan element child
Controller	Mengkostumisasi scroll controller
Primary	Menjadikan widget sebagai yang utama
physics	Membatasi scrolling saat konten penuh atau tidak
shrinkWrap	Membatasi child widget agar memenuhi layer dan pindah kebawah jika sudah penuh
padding	Memberikan jarak padding antar child widget
gridDelegate	Mengatur jumlah kolom dan jarak antar child widget
children	Memuat array widget yang berisikan lebih dari 1 widget

Properti GridView()

Properties	Fungsi
semanticChildCount	Menyimpan informasi semantic
dragStartBehavior	Mengubah alur drag gesture
keyboardDismissBehavior	Menghilangkan keyboard saat melakukan scrolling
restorationId	Menyimpan pemulihan ID agar mempertahankan posisi scrolling
cacheExtent	Jumlah cache yang akan disimpan, child akan disimpan sesudah ditampilkan maupun sebelum ditampilkan.
clipBehavior	Memberikan efek halus pada box dan berpengaruh pada frame rate

Properti GridView()

Properties	Fungsi
addAutomaticKeepAlives	Membungkus children agar dapat digunakan untuk KeepAliveNotifications
addRepaintBoundaries	Mencetak children dalam batas boundaries agar tidak perlu dicetak ulang saat scrolling
addSemanticIndexes	Mengatur index semantic dari children

MIKROSKIL

GridView.count

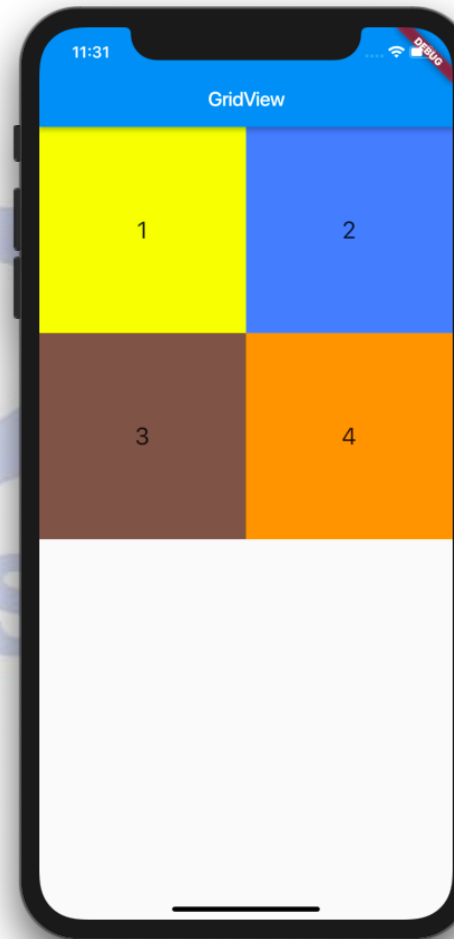
- Jumlah kolom dapat ditentukan
- Menampilkan widget sesuai dengan jumlah array atau child
- Property **crossAxisCount** harus ada
- Property **crossAxisCount** merupakan jumlah kolom untuk setiap element.



MIKROSKIL

GridView.count

```
GridView.count(
  crossAxisCount: 2,
  children: <Widget>[
    Container(
      color: Colors.yellowAccent,
      child: const Center(
        child: Text("1", style: TextStyle(fontSize: 24.0)),
      ),
    ),
    Container(
      color: Colors.blueAccent,
      child: const Center(
        child: Text("2", style: TextStyle(fontSize: 24.0)),
      ),
    ),
    Container(
      color: Colors.brown,
      child: const Center(
        child: Text("3", style: TextStyle(fontSize: 24.0)),
      ),
    ),
    Container(
      color: Colors.orange,
      child: const Center(
        child: Text("4", style: TextStyle(fontSize: 24.0)),
      ),
    ),
  ],
),
```



iPhone XR - 12.1

Properti GridView.count

Properties	Fungsi
scrollDirection	Mengubah arah scrolling pada GridView
Reverse	Membalikkan susunan element child
Controller	Mengkostumisasi scroll controller
Primary	Menjadikan widget sebagai yang utama
Physics	Membatasi scrolling saat konten penuh atau tidak
shrinkWrap	Membatasi child widget agar memenuhi layer dan pindah kebawah jika sudah penuh
padding	Memberikan jarak padding antar child widget
crossAxisCount	Mengatur jumlah kolom children
children	Memuat array widget yang berisikan lebih dari 1 widget

Properti GridView.count

Properties	Fungsi
mainAxisSpacing	Mengatur jarak vertical antar widget
crossAxisSpacing	Mengatur jarak horizontal antar widget
childAspectRatio	Mengatur aspek rasio setiap child widget
addAutomaticKeepAlives	Membungkus children agar dapat digunakan untuk KeepAliveNotifications
addRepaintBoundaries	Mencetak children dalam batas boundaries agar tidak perlu dicetak ulang saat scrolling
semanticChildCount	Menyimpan informasi semantic
dragStartBehavior	Mengubah alur drag gesture
keyboardDismissBehavior	Menghilangkan keyboard saat melakukan scrolling
addSemanticIndexes	Mengatur index semantic dari children

Properti GridView.count

Properties	Fungsi
restorationId	Menyimpan pemulihan ID agar mempertahankan posisi scrolling
cacheExtent	Jumlah cache yang akan disimpan, child akan disimpan sesudah ditampilkan maupun sebelum ditampilkan.
clipBehavior	Memberikan efek halus pada box dan berpengaruh pada frame rate

MIKROSKIL

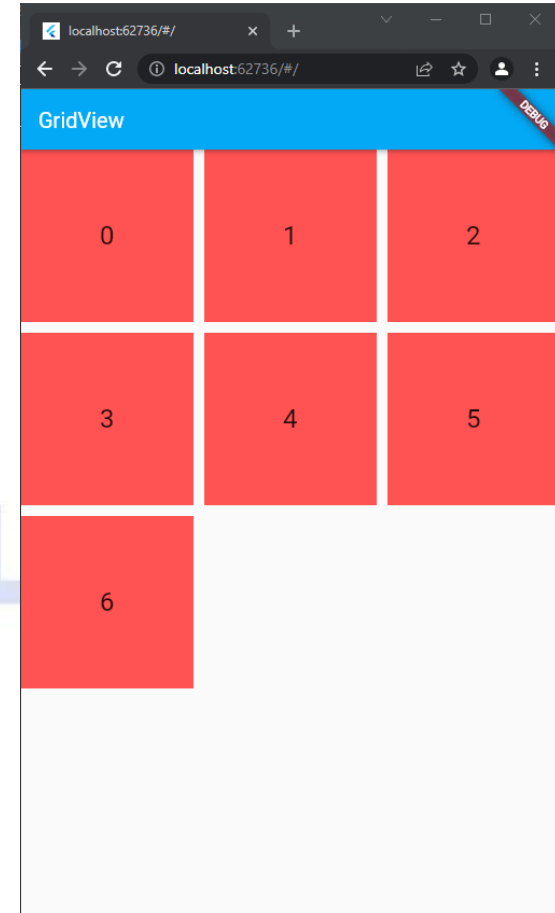
GridView.builder

- Jumlah kolom dapat ditentukan
- Dapat menampilkan widget dengan jumlah tertentu
- Membutuhkan property **gridDelegate** untuk mengatur jarak antar widget dan jumlah kolom (crossAxisCount)
- Membutuhkan property **itemBuilder** untuk menampilkan widget secara dinamis

MIKROSKIL

GridView.builder

```
GridView.builder(
  gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
    crossAxisCount: 3,
    mainAxisSpacing: 10,
    crossAxisSpacing: 10
  ),
  itemBuilder: (_, index) => Container(
    color: Colors.redAccent,
    child: Center(
      child: Text(index.toString(), style: TextStyle(fontSize: 24.0)),
    ),
  ),
  itemCount: 7,
)
```



Properti GridView.builder

Properties	Fungsi
scrollDirection	Mengubah arah scrolling pada GridView
Reverse	Membalikkan susunan element child
Controller	Mengkostumisasi scroll controller
Primary	Menjadikan widget sebagai yang utama
shrinkWrap	Membatasi child widget agar memenuhi layer dan pindah kebawah jika sudah penuh
padding	Memberikan jarak padding antar child widget
gridDelegate	Mengatur jumlah kolom dan jarak antar child widget
childrenDelegate	Berisi sejumlah widget yang menjadi child dari GridView
cacheExtent	Jumlah cache yang akan disimpan, child akan disimpan sesudah ditampilkan maupun sebelum ditampilkan.

Properti GridView.builder

Properties	Fungsi
semanticChildCount	Menyimpan informasi semantic
dragStartBehavior	Mengubah alur drag gesture
keyboardDismissBehavior	Menghilangkan keyboard saat melakukan scrolling
restorationId	Menyimpan pemulihan ID agar mempertahankan posisi scrolling
clipBehavior	Memberikan efek halus pada box dan berpengaruh pada frame rate

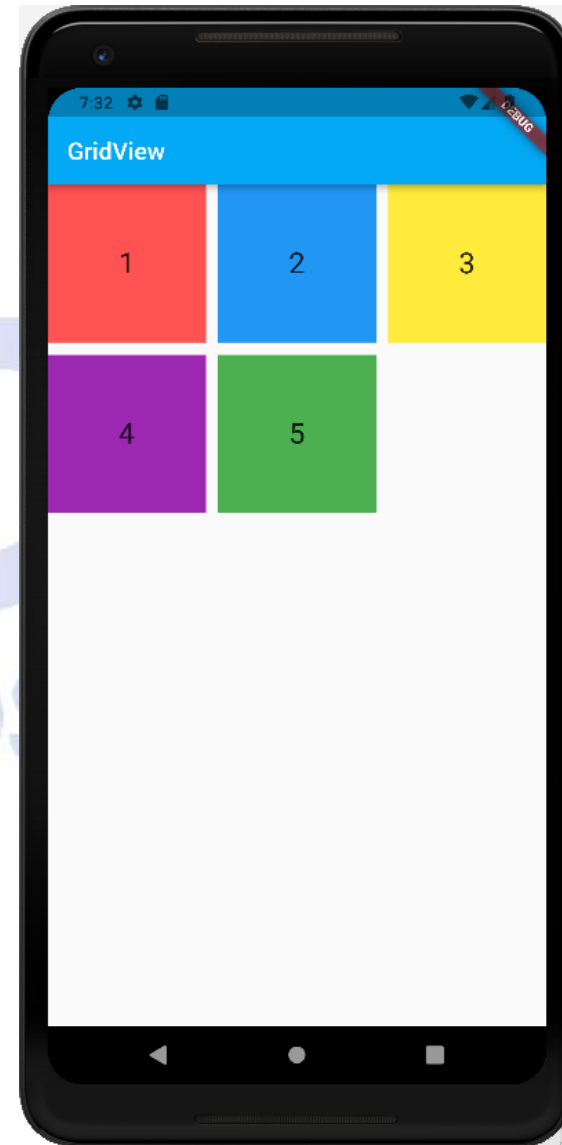
GridView.custom

- Merupakan widget GridView yang dapat digabungkan dengan SliverGridDelegate dan SliverChildDelegate
- Properti gridDelegate bersifat wajib
- Properti childrenDelegate bersifat wajib



GridView.custom

```
GridView.custom(  
  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
    crossAxisCount: 3,  
    mainAxisSpacing: 10,  
    crossAxisSpacing: 10  
  ),  
  childrenDelegate: SliverChildListDelegate([  
    Container(  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.blue,  
      child: const Center(  
        child: Text("2", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.yellow,  
      child: const Center(  
        child: Text("3", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.purple,  
      child: const Center(  
        child: Text("4", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.green,  
      child: const Center(  
        child: Text("5", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
  ])  
)
```



Properti GridView.custom

Properties	Fungsi
scrollDirection	Mengubah arah scrolling pada GridView
Reverse	Membalikkan susunan element child
Controller	Mengkostumisasi scroll controller
Primary	Menjadikan widget sebagai yang utama
shrinkWrap	Membatasi child widget agar memenuhi layer dan pindah kebawah jika sudah penuh
padding	Memberikan jarak padding antar child widget
gridDelegate	Mengatur jumlah kolom dan jarak antar child widget
childrenDelegate	Berisi sejumlah widget yang menjadi child dari GridView
cacheExtent	Jumlah cache yang akan disimpan, child akan disimpan sesudah ditampilkan maupun sebelum ditampilkan.

Properti GridView.custom

Properties	Fungsi
semanticChildCount	Menyimpan informasi semantic
dragStartBehavior	Mengubah alur drag gesture
keyboardDismissBehavior	Menghilangkan keyboard saat melakukan scrolling
restorationId	Menyimpan pemulihan ID agar mempertahankan posisi scrolling
clipBehavior	Memberikan efek halus pada box dan berpengaruh pada frame rate

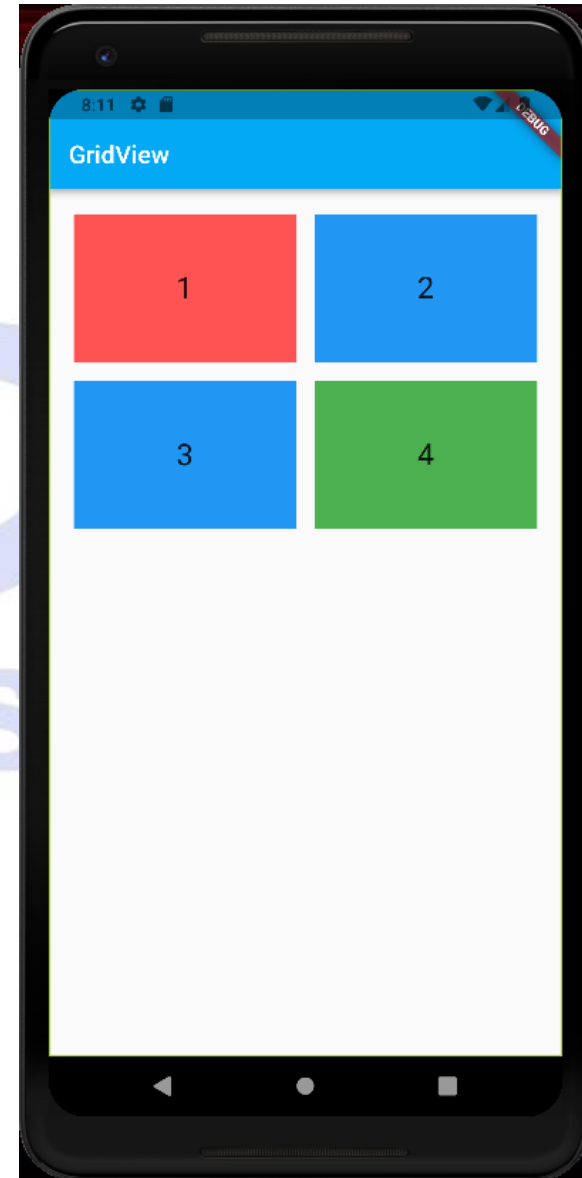
GridView.extent

- Jumlah kolom tidak dapat ditentukan
- Kolom ditentukan berdasarkan **ukuran** child widget
- Properti **maxCrossAxisExtent** dibutuhkan
- Jarak atau spacing antar element dapat langsung diatur
- Aspek ratio pada child widget lebih mudah diatur

MIKROSKIL

GridView.extent

```
GridView.extent(  
  maxCrossAxisExtent: 200,  
  mainAxisSpacing: 15,  
  crossAxisSpacing: 15,  
  childAspectRatio: 1.5,  
  padding: const EdgeInsets.all(20),  
  // shrinkWrap: true,  
  children: [  
    Container(  
      color: Colors.redAccent,  
      child: const Center(  
        child: Text("1", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.blue,  
      child: const Center(  
        child: Text("2", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.blue,  
      child: const Center(  
        child: Text("3", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
    Container(  
      color: Colors.green,  
      child: const Center(  
        child: Text("4", style: TextStyle(fontSize: 24.0)),  
      ),  
    ),  
  ],  
)
```



Properti GridView.extent

Properties	Fungsi
scrollDirection	Mengubah arah scrolling pada GridView
Reverse	Membalikkan susunan element child
Controller	Mengkostumisasi scroll controller
Primary	Menjadikan widget sebagai yang utama
physics	Membatasi scrolling saat konten penuh atau tidak
shrinkWrap	Membatasi child widget agar memenuhi layer dan pindah kebawah jika sudah penuh
padding	Memberikan jarak padding antar child widget
maxCrossAxisExtent	Mengatur setiap ukuran maksimal child widget untuk dibagi menjadi beberapa kolom
children	Memuat array widget yang berisikan lebih dari 1 widget

Properti GridView.extent

Properties	Fungsi
mainAxisSpacing	Mengatur jarak vertical antar widget
crossAxisSpacing	Mengatur jarak horizontal antar widget
childAspectRatio	Mengatur aspek rasio setiap child widget
semanticChildCount	Menyimpan informasi semantic
dragStartBehavior	Mengubah alur drag gesture
keyboardDismissBehavior	Menghilangkan keyboard saat melakukan scrolling
addAutomaticKeepAlives	Membungkus children agar dapat digunakan untuk KeepAliveNotifications
addRepaintBoundaries	Mencetak children dalam batas boundaries agar tidak perlu dicetak ulang saat scrolling
addSemanticIndexes	Mengatur index semantic dari children

GridView.extend

Properties	Fungsi
restorationId	Menyimpan pemulihan ID agar mempertahankan posisi scrolling
cacheExtent	Jumlah cache yang akan disimpan, child akan disimpan sesudah ditampilkan maupun sebelum ditampilkan.
clipBehavior	Memberikan efek halus pada box dan berpengaruh pada frame rate

MIKROSKIL

Navigation

- Navigation memiliki fungsi untuk menentukan bagaimana alur peralihan dari suatu screen ke screen lainnya.
- Misalnya dari dari halaman login ke home page, dst.
- Contoh navigasi sederhana, seperti berikut




Navigation

- Navigator bekerja seperti tumpukan layar (stack)
- Menggunakan prinsip LIFO (Last-In, First-Out).
- Ada dua method yang dapat digunakan pada Navigator widget yaitu :
 - 1.**Navigator.push ()**: Metode push digunakan untuk **menambahkan** rute lain ke atas tumpukan screen (stack) saat ini. Halaman baru ditampilkan di atas halaman sebelumnya.
 - 2.**Navigator.pop ()**: Metode pop menghapus rute paling atas dari tumpukan. Ini menampilkan halaman sebelumnya kepada pengguna.

Navigator : Penggunaan Push pada button

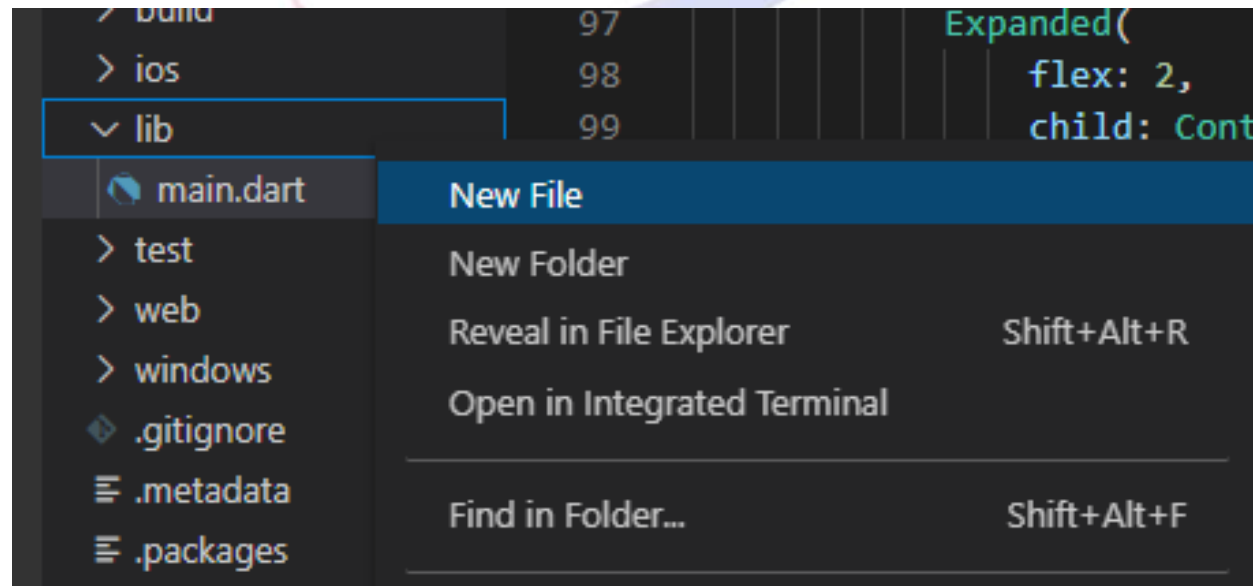
- Contoh

```
body: Center(  
  child: ElevatedButton (  
    onPressed: () {  
      Route route = MaterialPageRoute(builder: (context) => AboutPage());  
      Navigator.push(context, route);  
    },  
    child: Text('Tap Untuk ke AboutPage'),  
  ),  
)
```



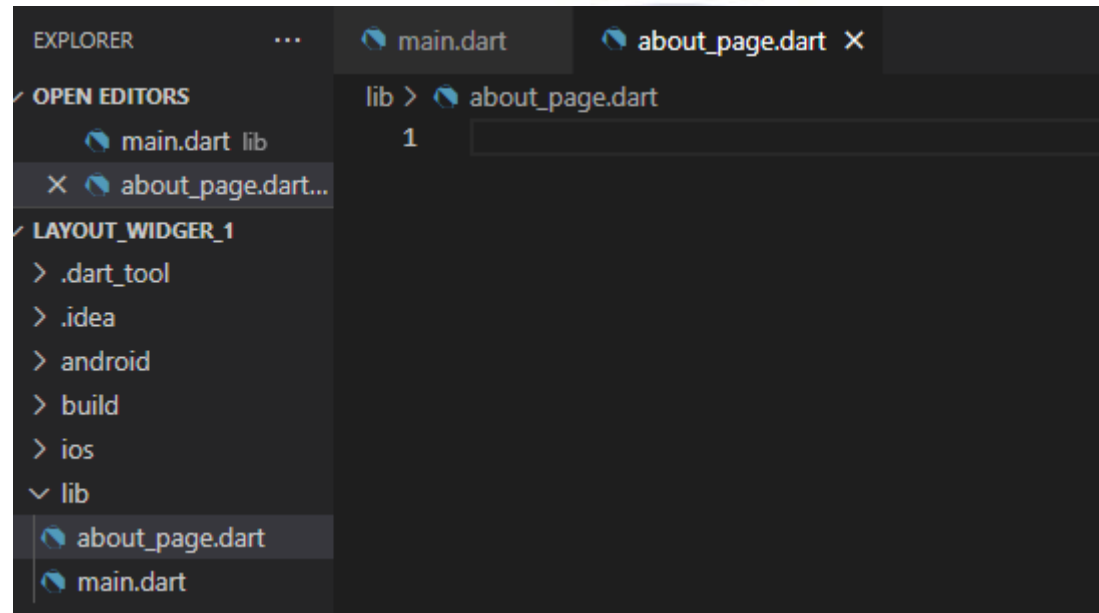
Catatan

- Saat ini main.dart anda belum mengenali AboutPage(), atau layer tujuan dari navigasi anda.
- Oleh karena itu anda perlu membuat sebuah file .dart baru
- Klik kanan pada folder lib, dan pilih new file



Catatan

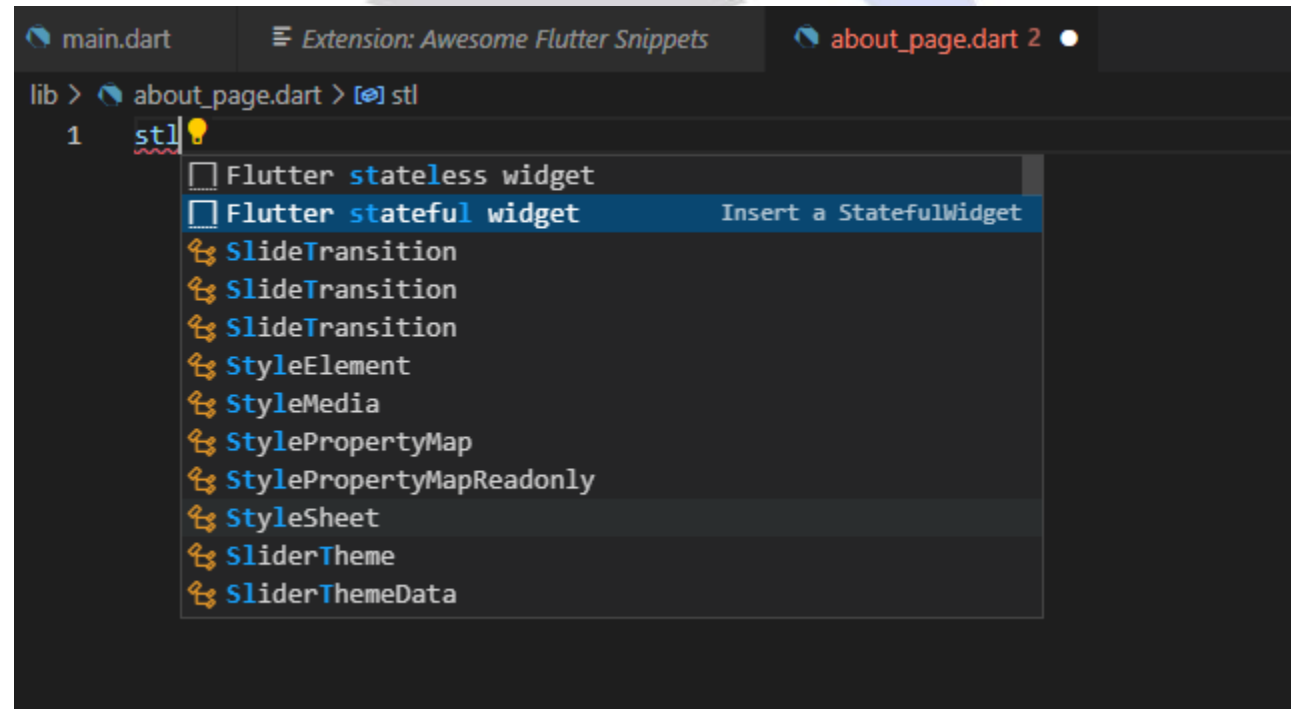
- Berikan nama file **about_page.dart**



- Maka saat ini anda telah memiliki 2 layar/page pada 1 aplikasi

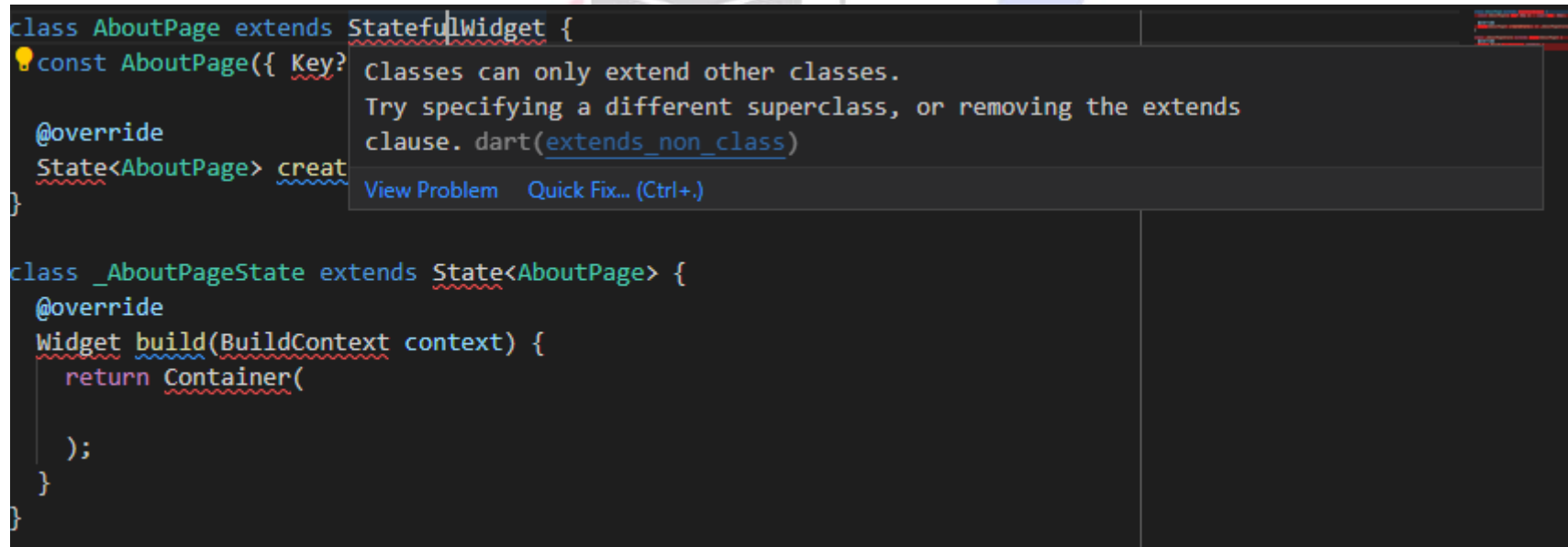
Catatan

- Setelah itu kita akan membuat Stateful class dengan mengetikkan “stl”, lalu masukkan namanya Classnya menjadi AboutPage



Catatan

- Karena pasti ada yang merah, tekan “ctrl + .” pada tulisan yang error(error yang mana saja) untuk menyelesaikannya lalu pilih **material.dart**.



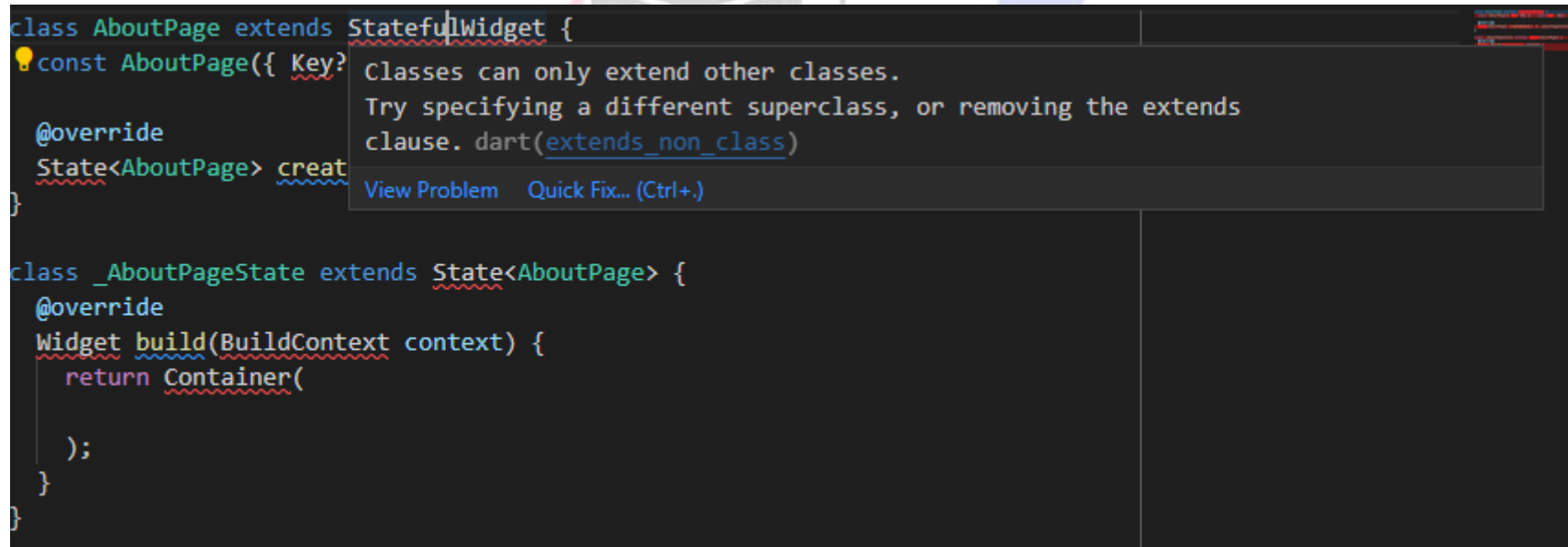
```
class AboutPage extends StatefulWidget {  
  ⚠ const AboutPage({ Key?  
    @override  
    State<AboutPage> creat  
}  
  
class _AboutPageState extends State<AboutPage> {  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
  );  
}
```

Classes can only extend other classes.
Try specifying a different superclass, or removing the extends clause. dart(material.dart)

View Problem Quick Fix... (Ctrl+.)

Catatan

- Karena pasti ada yang merah, tekan “ctrl + .” pada tulisan yang error(error yang mana saja) untuk menyelesaikannya lalu pilih **material.dart**.



```
class AboutPage extends StatefulWidget {  
  ⚠ const AboutPage({ Key?  
    @override  
    State<AboutPage> creat  
}  
  
class _AboutPageState extends State<AboutPage> {  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
  );  
}
```

Classes can only extend other classes.
Try specifying a different superclass, or removing the extends clause. dart(material.dart)

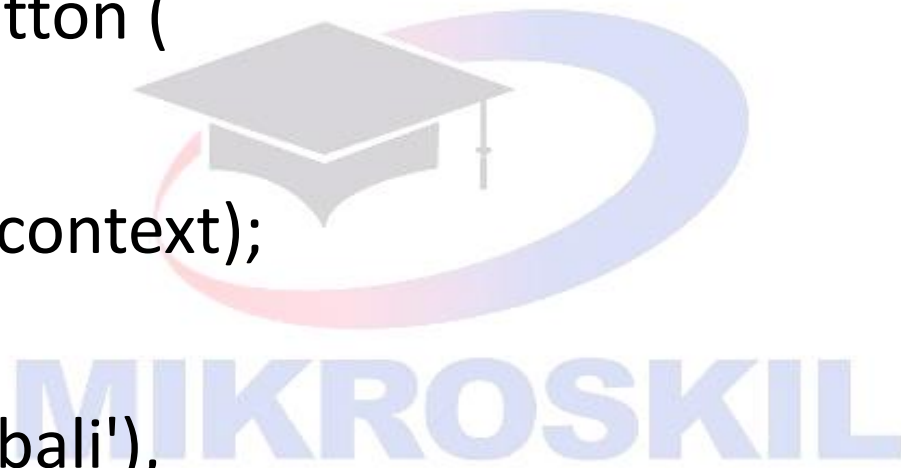
View Problem Quick Fix... (Ctrl+.)

Hasilnya

```
lib > about_page.dart > AboutPage
1  import 'package:flutter/material.dart';
2
3  class AboutPage extends StatefulWidget {
4    const AboutPage({ Key? key }) : super(key: key);
5
6    @override
7    State<AboutPage> createState() => _AboutPageState();
8  }
9
10 class _AboutPageState extends State<AboutPage> {
11   @override
12   Widget build(BuildContext context) {
13     return Container(
14
15     );
16   }
17 }
```

Navigator : Penggunaan Pop pada button

```
Center(  
  child: ElevatedButton (  
    onPressed: () {  
      Navigator.pop(context);  
    },  
    child: Text('Kembali'),  
  ),  
,
```



Hasil



Terima Kasih

