	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 1 dari 20

A. Tujuan

Setelah menyelesaikan job sheet ini, diharapkan mahasiswa mampu:

1. Memahami container widget pada flutter
2. Menggunakan container widget
3. Mengetahui properti pada container widget
4. Menerapkan penggunaan container widget pada project yang dikerjakan

B. Petunjuk

1. Awali setiap aktivitas anda dengan doa, agar anda lancar dalam belajar
2. Kerjakan tugas-tugas praktikum dengan baik, teliti dan sabar
3. Tanyakan kepada dosen apabila ada hal-hal yang kurang jelas

C. Dasar Teori


Pengertian Container Widget

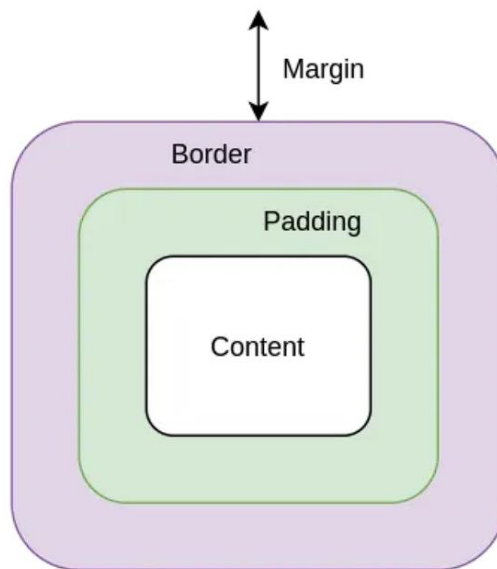
Widget container adalah sebuah widget yang bertugas sebagai pembungkus dari widget lain.

Dinamakan container karena nantinya, ia bisa memiliki properti-properti bingkai seperti:

- padding
- margin
- border
- border radius
- box shadow
- dan lain-lain

Ilustrasi:

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 2 dari 20



Pada ilustrasi di atas, kotak berwarna putih adalah child widget. Sedangkan semuanya dari margin, border, dan padding adalah widget container.

D. Latihan

1. Persiapan


Siapkan code pada main.dart dan ubah agar seperti berikut ini:

```
import 'package:flutter/material.dart';

class StateTeksUtama extends State<TeksUtama> {

  var listNama = [
    'Budi', 'Andi', 'Sari', 'Rudi',
    'Nurul', 'Huda', 'Lendis', 'Fabri'
  ];

  var listWarna = [
    Colors.red, Colors.purple, Colors.teal, Colors.lime,
    Colors.indigo, Colors.deepPurple, Colors.cyan,
```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 3 dari 20


```

Colors.blue, Colors.yellow
];
int index = 0;

void incrementIndex () {
  setState(() {
    index++;
  });
}

@override
Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.symmetric(vertical: 20, horizontal: 50),
    margin: EdgeInsets.all(20),
    decoration: BoxDecoration(
      color: Colors.cyan[50],
      borderRadius: BorderRadius.circular(5),
      boxShadow: [BoxShadow(
        color: Colors.black12,
        spreadRadius: 2,
        offset: Offset.fromDirection(-5.5, 6.0),
        blurRadius: 2
      )]
    ),
    child: Column(
      children: [
        Text(
          'Apa kabar',
          textDirection: TextDirection.ltr,

```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 4 dari 20

```

    ),
    Text(
      listNama[index % listNama.length],
      textDirection: TextDirection.ltr,
      style: TextStyle(
        fontSize: 45,
        fontWeight: FontWeight.bold,
        color: listWarna[index % listWarna.length]
      ),
    ),
  ],
),
);
}

}

class TeksUtama extends StatefulWidget {


  final state = StateTeksUtama();

  @override
  State<StatefulWidget> createState() => state;
}

void main() {
  TeksUtama teksUtama = TeksUtama();

  runApp(
    MaterialApp(

```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 5 dari 20

```

    title: 'Halo Dunia',
    home: Scaffold(
      appBar: AppBar(
        title: Text('Halo Dunia'),
      ),
      body: Center(
        child: Column(
          children: [
            teksUtama
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        tooltip: 'Random',
        child: Icon(Icons.refresh),
        onPressed: teksUtama.state.incrementIndex,
      ),
    )
  );
}

```

2. Bungkus UI Widget TeksUtama Dengan Container

Langkah pertama, kita akan membungkus ui dari widget `TeksUtama` dengan sebuah `Container`.


Caranya:

Ubah kode program ini (pada kelas `StateTeksUtama`):

```

@override
Widget build(BuildContext context) {

```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 6 dari 20

```

return Column(
  children: [
    Text(
      'Apa kabar',
      textDirection: TextDirection.ltr,
    ),
    Text(
      listNama[index % listNama.length],
      textDirection: TextDirection.ltr,
      style: TextStyle(
        fontSize: 45,
        fontWeight: FontWeight.bold,
        color: listWarna[index % listWarna.length]
      ),
    ),
  ],
);


```

Lalu ganti dengan kode berikut (*lihat kode program yang ter-highlight*):

```

@override
Widget build(BuildContext context) {
  return Container(
    child: Column(
      children: [
        Text(
          'Apa kabar',
          textDirection: TextDirection.ltr,
        ),

```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 7 dari 20

```

        Text (
            listNama[index % listNama.length],
            textDirection: TextDirection.ltr,
            style: TextStyle(
                fontSize: 45,
                fontWeight: FontWeight.bold,
                color: listWarna[index %
listWarna.length]
            ),
        ),
    ],
),
);
}

```

Coba jalankan aplikasi. Kita akan dapati bahwa tidak ada perbedaan berarti pada tampilan aplikasi.


Itu terjadi karena memang widget container ini belum kita beri property atau argumen apa pun.

3. Properti Decoration

Properti `Container.decoration` berfungsi untuk mendefinisikan dekorasi dari sebuah container.

Ada beberapa jenis dekorasi pada flutter semisal `BoxDecoration`, `ShapeDecoration`, dan lain-lain.

Di sini kita hanya akan menggunakan `BoxDecoration` karena tampilan yang kita inginkan dari object kita adalah sebuah box.

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 8 dari 20

4. Properti Border

Kita mulai dari properti yang pertama: yaitu border.

Kenapa border? Border berfungsi agar kita tahu batas dari objek yang kita bangun.


Istilah lainnya: border akan menjadi sebagai garis pembantu buat kita untuk mempermudah proses penyusunan layout.

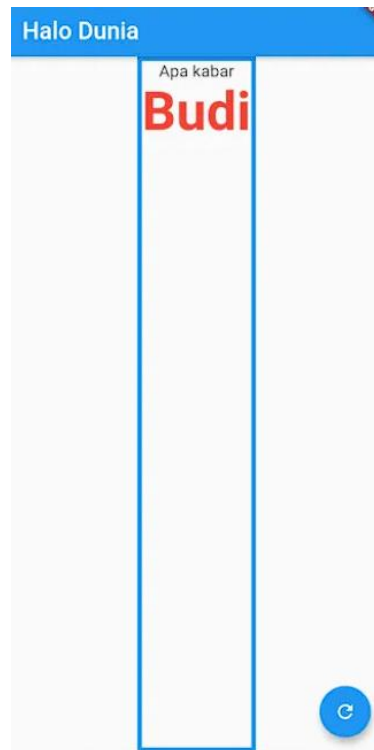
Untuk garis bantuan sendiri, kita bisa menggunakan devtools inspector dari flutter, akan tetapi kita skip dulu untuk pertemuan kali ini.

Langsung saja, tambahkan kode program berikut ini pada `Container` yang telah kita buat tadi.

```
@override
Widget build(BuildContext context) {
  return Container(
    decoration: BoxDecoration(
      border: Border.all(
        color: Colors.blue,
        width: 5,
      ),
    ),
    child: Column(
      ...
    ),
  )
}
```


Ketika dijalankan, kita akan mendapatkan output:

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 9 dari 20

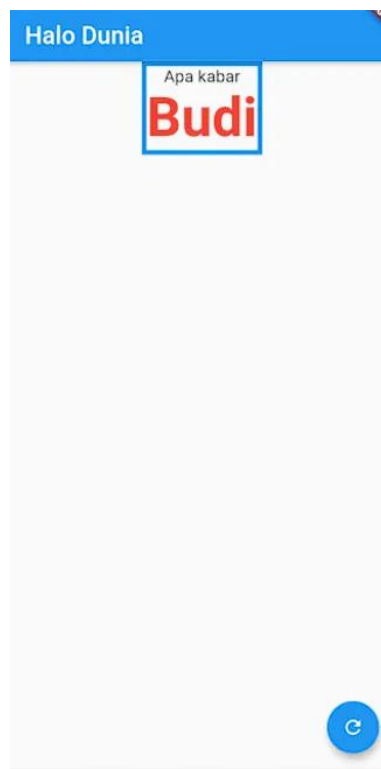


Pada fungsi utama main(), masukkan widget teksUtama ke-dalam widget Column agar posisinya berada di atas. Insyaallah kita akan membahas lebih lanjut tentang Column pada pertemuan-pertemuan yang akan datang.

```
...
body: Center(
  child: Column(
    children: [
      teksUtama
    ],
  ),
),
...
```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 10 dari 20

Sekarang, hasil komponen kita menjadi seperti berikut:




5. Properti Padding

Seperti pada ilustrasi sebelumnya, padding adalah sebuah jarak antara content (dalam hal ini child widget) dengan border.

Kita bisa menambahkannya dengan menambahkan arguman `Container.padding`:

```
@override
Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.all(20),
    decoration: BoxDecoration(
      border: Border.all(
```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 11 dari 20

```

        color: Colors.blue,
        width: 5,
      ),
    ),
    ...
  )
}

```

Anda juga bisa menggunakan variasi lain dari padding seperti:

```

// ganti left, top, right, dan bottom dengan angka
EdgeInsets.fromLTRB(left, top, right, bottom)

```

Atau dengan mendefinisikan padding vertical dan padding horizontal:

```

EdgeInsets.symmetric(vertical: 20, horizontal: 50),

```


Atau menggunakan fungsi `EdgeInsets.only()` untuk mendefinisikan arah tertentu saja yang diberi padding:

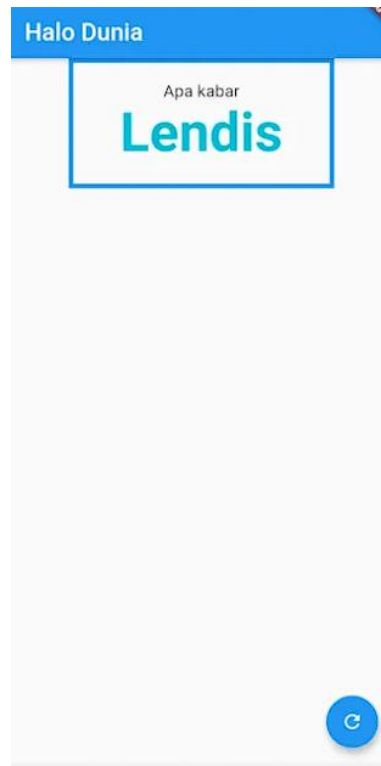
```

EdgeInsets.only(top: 50, right: 150)

```

Kira-kira output-nya adalah seperti tangkapan layar berikut:

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 12 dari 20




6. Properti Margin

Kebalikan dari padding, margin adalah jarak antara border dengan dunia luar.

Kita bisa mendefinisikannya dengan menambahkan argumen `margin` saat membuat objek Container seperti berikut:

```
@override
Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.symmetric(vertical: 20, horizontal: 50),
    margin: EdgeInsets.all(20),
    decoration: BoxDecoration(
      border: Border.all(
        color: Colors.blue,
        width: 5,

```

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 13 dari 20

```

    ),
  ),
  ...
)
}

```

Sama seperti padding, Anda juga bisa menggunakan variasi lain untuk mendefinisikan margin, seperti:

```

// ganti left, top, right, dan bottom dengan angka
EdgeInsets.fromLTRB(left, top, right, bottom)

```

Atau dengan mendefinisikan margin vertical dan margin horizontal:

```

EdgeInsets.symmetric(vertical: 20, horizontal: 50),

```


Atau menggunakan fungsi `EdgeInsets.only()` untuk mendefinisikan arah tertentu saja yang diberi margin:

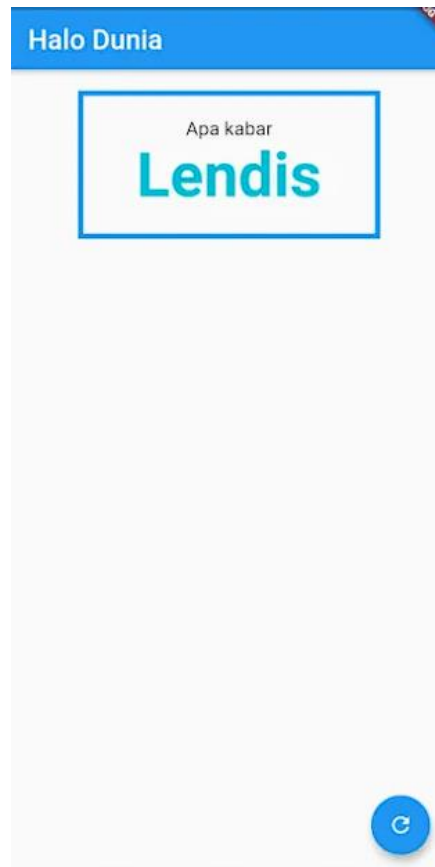
```

EdgeInsets.only(top: 50, right: 150)

```

Output aplikasi kita setelah diberi margin, kira-kira seperti berikut:

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 14 dari 20




7. Properti Color (Background)

Dengan menggunakan container widget, kita bisa membungkus sebuah UI tampilan dan memberikannya warna latar belakang.

Caranya adalah dengan mendefinisikan atribut atau argumen `decoration.color`.

Simak contoh berikut:

```
@override
Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.symmetric(vertical: 20, horizontal: 50),
    margin: EdgeInsets.all(20),
    decoration: BoxDecoration(
```

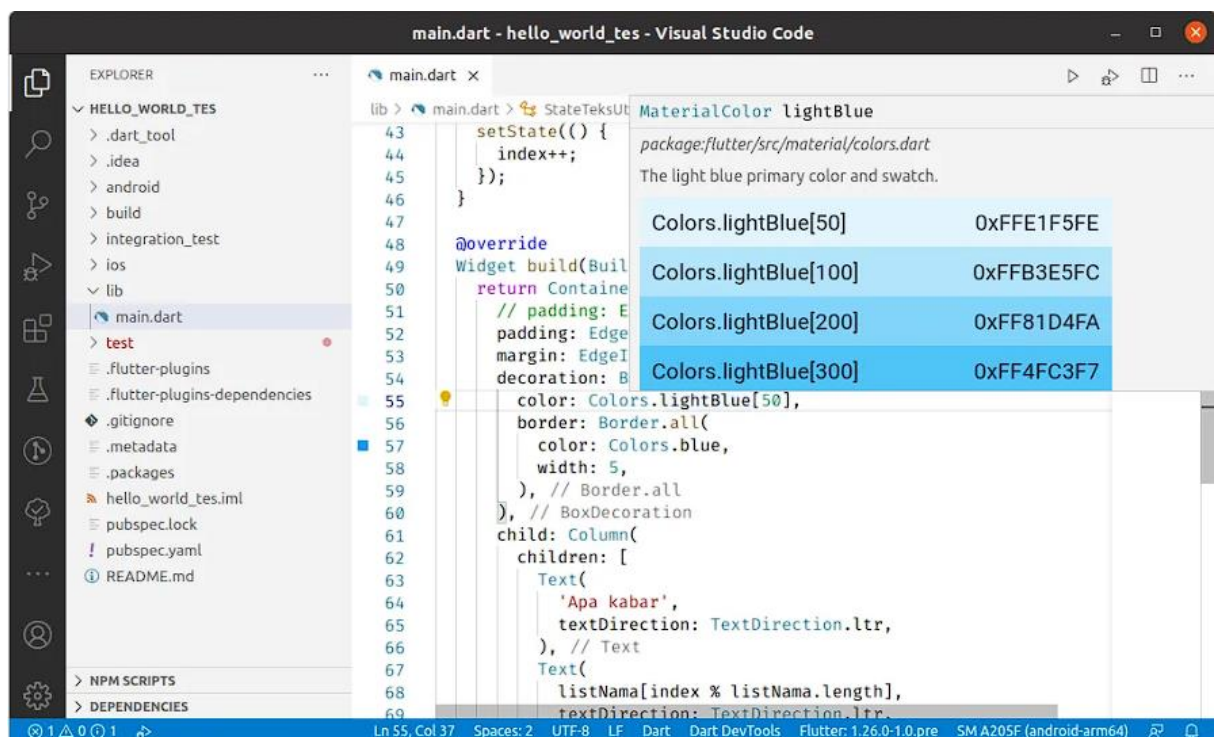
	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 15 dari 20


```

        color: Colors.lightBlue[50],
        border: Border.all(
          color: Colors.blue,
          width: 5,
        ),
      ),
      ...
    )
  }

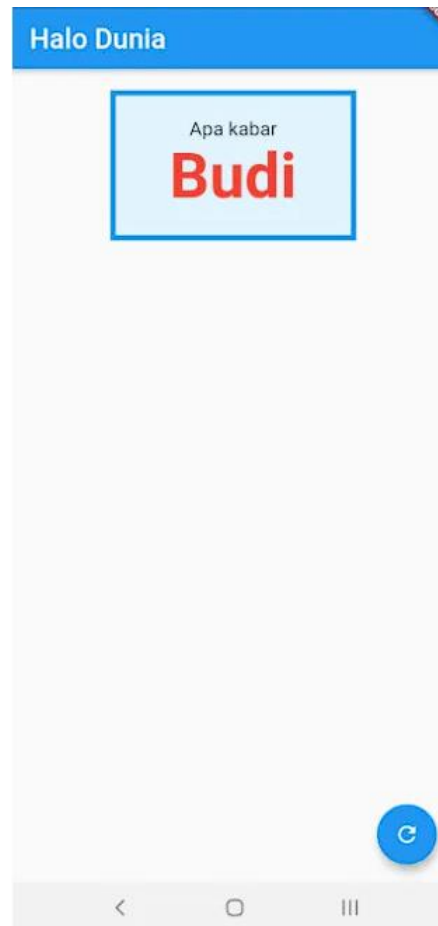
```

Pada VSCode, anda bisa mengarahkan cursor ke arah bagian Colors.<warna>, maka akan muncul preview colornya seperti tangkapan layar berikut:



	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 16 dari 20

Hasil output dari potongan kode program di atas adalah seperti ini:




8. Properti Border Radius

Selain border, kita juga bisa mendefinisikan border radius dari sebuah container. Border radius akan membuat box container kita tidak 90% pada sudut-sudutnya.

Perhatikan contoh berikut:

```
Widget build(BuildContext context) {
  return Container (
    padding: EdgeInsets.symmetric(vertical: 20, horizontal:
    50) ,
```

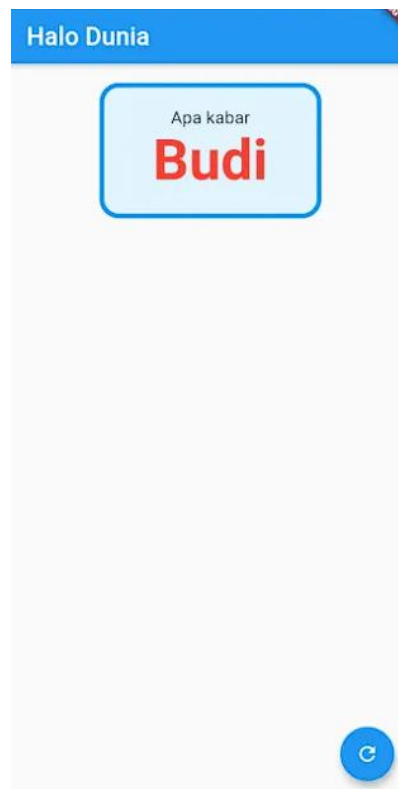

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 17 dari 20


```

margin: EdgeInsets.all(20),
decoration: BoxDecoration(
  color: Colors.lightBlue[50],
  borderRadius: BorderRadius.circular(20),
  border: Border.all(
    color: Colors.blue,
    width: 5,
  ),
),
...
)
}

```

Output dari kode program di atas:



	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 18 dari 20

9. Properti Box Shadow

Properti terakhir yang akan kita bahas adalah: box shadow.

Box shadow adalah properti yang mendefinisikan objek bayangan dari suatu widget component. Dengan box shadow, kita bisa memberikan efek 3D kepada suatu objek, atau kita juga bisa mempertegas tumpukan berbagai objek: mana objek yang ada di atas dan mana yang ada di bawah.


Agar bayangan lebih terlihat, kita hapus property **border**, kemudian kita tambahkan property **boxShadow**.

Perhatikan contoh berikut:

```
Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.symmetric(vertical: 20, horizontal: 50),
    margin: EdgeInsets.all(20),
    decoration: BoxDecoration(
      color: Colors.lightBlue[50],
      borderRadius: BorderRadius.circular(20),
      boxShadow: [BoxShadow(
        color: Colors.black12,
        spreadRadius: 5,
        blurRadius: 5,
      )],
      ...
    ),
    ...
  )
}
```

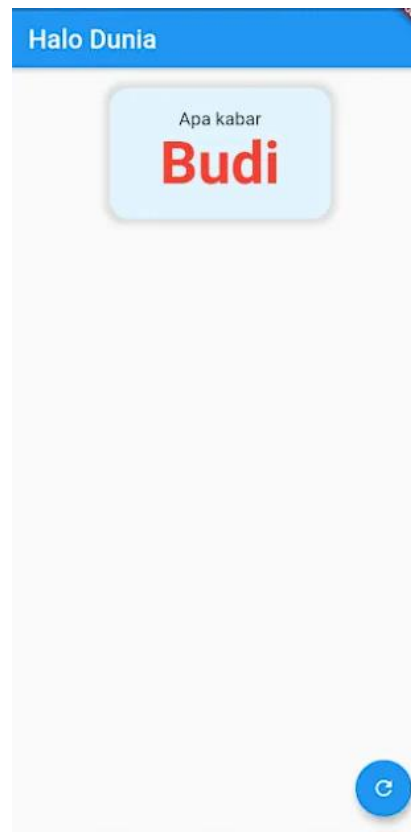
Di dalam kode program di atas:


- bagian `spreadRadius` untuk mendefinisikan radius penyebaran bayangan
- dan `blurRadius` adalah untuk daya buram/kabur dari bayangan

	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 19 dari 20

Save dan jalankan kembali aplikasi.

Sekarang anda akan mendapatkan tampilan yang kira-kira seperti ini:



	STMIK Widya Utama	Mobile Programming Lanjut
	FLUTTER CONTAINER WIDGET	Job Sheet 07
Semester 6	Dosen : Amin Sholeh, M.Kom.	Hal: 20 dari 20

E. Tugas Praktikum

1. Buatlah project baru untuk menampilkan seperti gambar disamping dengan memanfaatkan properti properti yang ada di flutter!
Upload project tersebut ke github dengan nama repository **jobsheet7**

Kirimkan link repository tugas ke tautan berikut:

<https://s.id/kelasflutterswu>

