

Modul 10

Komponen dalam mendesain Aplikasi



Banners, Dialog, dan Snackbars

Module Overview

Mengenal komponen atau widget Baners, Dialog dan Snackbars pada Flutter. Widget ini digunakan untuk tampil, memberikan informasi dan berinteraksi kepada pengguna dalam bentuk dialog yang tampil di atas konten.

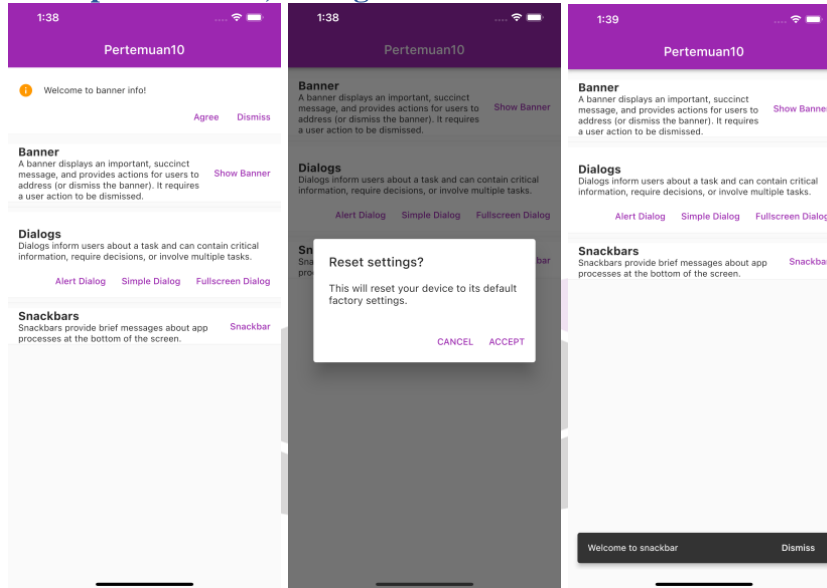
Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

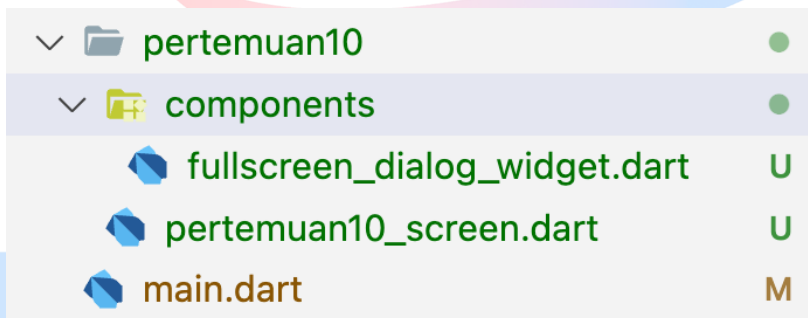
- Mengetahui kegunaan dari Baners, Dialog dan Snackbars.
- Mampu menerapkan Baners, Dialog dan Snackbars pada flutter.

Baners, Dialog dan Snackbars

Menerapkan Baners, Dialog dan Snackbars



1. Berikut adalah struktur folder untuk pertemuan 09.



Keterangan:

- pertemuan10_screen.dart digunakan untuk screen UI flutter (root home)
- fullscreen_dialog_widget.dart digunakan untuk dialog fullscreen.

2. Buatlah screen baru dengan nama file **pertemuan10_screen.dart** pada folder pertemuan10. Import **material**, Gunakan **StatefulWidget** dan gunakan **Scaffold** untuk memulai mendesain komponen. Namai class **statefulWidget** dengan **Pertemuan10Screen**.

```

1  import 'package:flutter/material.dart';
2
3  class Pertemuan10Screen extends StatefulWidget {
4    const Pertemuan10Screen({Key? key}) : super(key: key);
5
6    @override
7    State<Pertemuan10Screen> createState() => _Pertemuan10ScreenState();
8  }
9
10 class _Pertemuan10ScreenState extends State<Pertemuan10Screen> {
11   @override
12   Widget build(BuildContext context) {
13 >     return Scaffold( // Scaffold ...
14   }
15 }

```

3. Menggunakan **Banner**. Banner akan tampil di atas layar, maka sebelum memanggil banner buatlah sebuah fungsi untuk menampilkan banner. Di bawah method `Widget Build()` buatlah sebuah method baru dengan nama: **showBanner** (penamaan bebas), kemudian return sebuah **MaterialBanner** seperti contoh berikut:



```

1  showBanner(BuildContext context) {
2    return MaterialBanner(
3      content: const Text('Welcome to banner info!'),
4      leading: const Icon(
5        Icons.info,
6        color: Colors.orange,
7      ),
8      actions: [
9        TextButton(onPressed: () {}, child: const Text('Agree')),
10       TextButton(
11         onPressed: () =>
12           ScaffoldMessenger.of(context).hideCurrentMaterialBanner(),
13         child: const Text('Dismiss')
14       );
15   ]
16 }

```

Keterangan:

- `MaterialBanner()`, sebuah widget untuk menampilkan banner dengan style Material.
- property `actions` berupa list untuk menampilkan tombol aksi pada banner.

- `ScaffoldMessenger.of(context).hideCurrentMaterialBanner()` untuk dismiss banner yang tampil.

4. Menampilkan **Banners**. Setelah membuat fungsi `showBanner` di atas, maka dibutuhkan aksi untuk menampilkan banner. Pada bagian **body scaffold** gunakan widget `column` dan `MaterialBanner` dengan action berupa `TextButton` untuk memanggil **`showBanner()`**.



```

1  body: Column(children: [
2    const SizedBox(height: 16),
3    MaterialBanner(
4      content: Column(
5        crossAxisAlignment: CrossAxisAlignment.start,
6        children: const [
7          Text(
8            'Banner',
9            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
10         ),
11         Text('Text.')
12       ],
13     actions: [
14       TextButton(
15         onPressed: () => ScaffoldMessenger.of(context)
16           ..removeCurrentMaterialBanner()
17           ..showMaterialBanner(showBanner(context)),
18         child: const Text('Show Banner'))
19     ],

```



Keterangan:

- **MaterialBanner** disini dapat digunakan sebagai widget yang tampil pada body (mirip seperti widget ListTile atau Card), namun ini hanya contoh! gunakanlah MaterialBanner selayaknya banner yang tampil di atas konten.
- **ScaffoldMessenger.of(Context)** digunakan untuk menampilkan banner di atas dari scaffold dan dapat juga digunakan untuk membersihkan banner untuk menghindari penumpukkan banner.

!!Restart dan klik button "ShowBanner" untuk melihat hasilnya.

5. Menggunakan **Dialog**. Beberapa contoh dialog yaitu AlertDialog, SimpleDialog dan FullScreenDialog akan digunakan pada contoh ini. Persiapkan fungsi untuk memanggil widget ini di bawah fungsi/method showBanner() yang telah dibuat sebelumnya.

fungsi showAlertDialog()

```
1  showAlertDialog(BuildContext context) {  
2    return AlertDialog(  
3      title: const Text('Reset settings?'),  
4      content: const Text(  
5        'This will reset your device to its default factory settings.'),  
6      actions: [  
7        TextButton(  
8          onPressed: () {  
9            Navigator.pop(context);  
10           },  
11          child: Text('CANCEL'),  
12        ),  
13        TextButton(  
14          onPressed: () {  
15            Navigator.pop(context);  
16           },  
17          child: Text('ACCEPT'),  
18        ),  
19      ],  
20    );  
21  }
```

Keterangan:

- fungsi ini akan return sebuah widget AlertDialog untuk menampilkan title, content, dan action pada Alert dialog.

Fungsi showSimpleDialog()



```

1  showSimpleDialog(BuildContext context) {
2    return SimpleDialog(
3      title: const Text('Set backup account'),
4      children: List.generate(4, ((index) {
5        return SimpleDialogOption(
6          onPressed: () => Navigator.pop(context, 'mail$index@mail.com'),
7          child: Row(
8            mainAxisAlignment: MainAxisAlignment.start,
9            crossAxisAlignment: CrossAxisAlignment.center,
10           children: [
11             const Icon(Icons.abc, size: 36.0, color: Colors.amber),
12             Padding(
13               padding: const EdgeInsetsDirectional.only(start: 16.0),
14               child: Text('Username$index'),
15             ),
16           ],
17         ),
18       })),
19     );
20   };
21 }

```

Keterangan:

- **SimpleDialog** adalah widget berupa dialog sederhana untuk menampilkan title dan childer.
 - **List.generator** disini untuk build childer dari simple dialog. Cara ini sama seperti membuat manual children pada list sebanyak 4 kali.
 - **SimpleDialogOption** sebuah widget pada SimpleDialog untuk menampilkan aksi dialog (mirip seperti textButton, IconButton atau tombol lainnya.)
6. Menampilkan AlertDialog, SimpleDialog dan FullScreenDialog. Sama seperti cara menampilkan Banner pada tahap sebelumnya. Buatlah sebuah MaterialBanner pada column body, lalu terapkan button pada action. Contoh:



```
1  MaterialBanner(  
2    content: Column(  
3      crossAxisAlignment: CrossAxisAlignment.start,  
4      children: const [  
5        Text(  
6          'Dialogs',  
7          style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),  
8        ),  
9        Text('Text')  
10     ]),  
11    actions: [  
12      TextButton(  
13        onPressed: () => showDialog(  
14          context: context,  
15          builder: (context) => showAlertDialog(context)),  
16        child: const Text('Alert Dialog')),  
17      TextButton(  
18        onPressed: () => showDialog(  
19          context: context,  
20          builder: (context) => showSimpleDialog(context)),  
21        child: const Text('Simple Dialog')),  
22      TextButton(  
23        onPressed: () => Navigator.push(  
24          context,  
25          MaterialPageRoute(  
26            fullscreenDialog: true,  
27            builder: ((context) => FullScreenDialog()))),  
28        child: const Text('Fullscreen Dialog'))  
29    ]),
```

MIKROSKIL

Keterangan:

- Pada actions ada 3 TextButton untuk memanggil fungsi yang telah dibuat sebelumnya.
- **showDialog()** berfungsi untuk menampilkan dialog yang dibuat sebelumnya.
- sedangkan untuk **FullScreenDialog** hanya membutuhkan sebuah navigator dengan MaterialPageRoute yang memiliki property **fullscreenDialog: true**.
Ini akan membuat screen tujuan berlaku sebagai sebuah fullscreen.

7. Widget **FullScreenDialog**. Buatlah sebuah screen baru menggunakan statelessWidget dengan nama file: **fullscreen_dialog_widget.dart**.



```
1  import 'package:flutter/material.dart';
2
3  class FullScreenDialog extends StatelessWidget {
4    const FullScreenDialog({Key? key}) : super(key: key);
5    @override
6    Widget build(BuildContext context) {
7      return Scaffold(
8        appBar: AppBar(
9          title: const Text('Full-screen Dialog'),
10       ),
11       body: const Center(
12         child: Text("Full-screen dialog"),
13       ),
14     );
15   }
16 }
```


8. Menggunakan **SnackBar**. Seperi cara sebelumnya, **buatlah sebuah fungsi `showSnackBar()`** untuk snackbar di bawah fungsi `showAlertDialog()`, seperti contoh berikut:

```
1  showSnackBar(BuildContext context) {  
2    return SnackBar(  
3      behavior: SnackBarBehavior.floating,  
4      duration: Duration(seconds: 6),  
5      content: Text('Welcome to snackbar'),  
6      action: SnackBarAction(  
7        label: 'Dismiss',  
8        textColor: Colors.white,  
9        onPressed: () => ScaffoldMessenger.of(context).removeCurrentSnackBar(),  
10     ),  
11   );  
12 }  
13 }
```

Keterangan:

- **SnackBar()** adalah widget untuk menampilkan dialog dari bawah screen.
- Widget ini akan tampil selama 4 detik secara default, namun dapat diubah dengan **Duration()** pada property duration.
- **SnackBarAction()** pada property action dibutuhkan untuk memberikan interaksi dismiss kepada pengguna. Gunakan `removeCurentSnackBar()` pada `ScaffoldMessenger()`.

9. Menampilkan SnackBar. Sama seperti cara menampilkan Banner. SnackBar dipanggil pada **ScaffoldMessenger** dalam property **showSnackBar()**. Tambahkan MaterialBanner pada column body seperti gambar berikut:

```
1  MaterialBanner(  
2    content: Column(  
3      crossAxisAlignment: CrossAxisAlignment.start,  
4      children: const [  
5        Text(  
6          'Snackbars',  
7          style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),  
8        ),  
9        Text(  
10       'Snackbars provide brief messages about app processes at the bottom of the screen.')  
11      ]),  
12    actions: [  
13      TextButton(  
14        child: const Text('Snackbar'),  
15        onPressed: () => ScaffoldMessenger.of(context)  
16          .showSnackBar(showSnackBar(context)),  
17      ),  
18    ],  
19  ),
```

!!Restart dan klik button "Snackbar" untuk melihat hasilnya.

UNIVERSITAS
MIKROSKIL

Latihan Praktek

Menggunakan Baners, Dialog dan Snackbars buatlah tampilan berikut:

Buatlah sebuah **Dialog** untuk menampilkan list ringtone seperti gambar di bawah, kemudian, setelah salah satu ringtone terpilih dan klik "OK", maka tampilkan sebuah **Banner** (dengan style di bawah) yang nampilkan nama ringtone yang terpilih. Kemudian, saat action "Update" pada banner di klik, maka akan tampil sebuah **snackBar** yang menampilkan pesan "Ringtone "nama_ringtone" berhasil diterapkan.":

Referensi tampilan (sesuaikan dengan narasi latihan):

