

Modul 11

Improving User Experience (1)

Flutter's support for accessibility and Flutter internationalization

Module Overview

Menyediakan Accessibility di Flutter

Menambahkan terjemahan ke dalam Aplikasi

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat memberikan pengalaman pengguna yang lebih baik dengan aksesibilitas dan internasionalisasi

Ketika anda ingin memasarkan aplikasi yang anda bangun, pastinya anda akan berusaha untuk menciptakan aplikasi yang dapat digunakan banyak orang. Menambahkan internasionalisasi ke aplikasi akan sangat berkontribusi pada pertumbuhan penjualan aplikasi anda di pasar karena aplikasi yang menambahkan internasionalisasi dapat menjangkau lebih banyak masyarakat. Dengan cara yang sama, membuat aplikasi dengan aksesibilitas adalah langkah penting untuk menjangkau sebanyak mungkin orang dan memberikan pengalaman pengguna yang lebih baik.

Salah satu aksesibilitas yang disediakan Flutter yang berfokus pada pengguna dengan beberapa bentuk kecacatan adalah dengan menyediakan cara untuk membuat aplikasi lebih mudah diakses, seperti beberapa komponen untuk memberikan dukungan aksesibilitas:

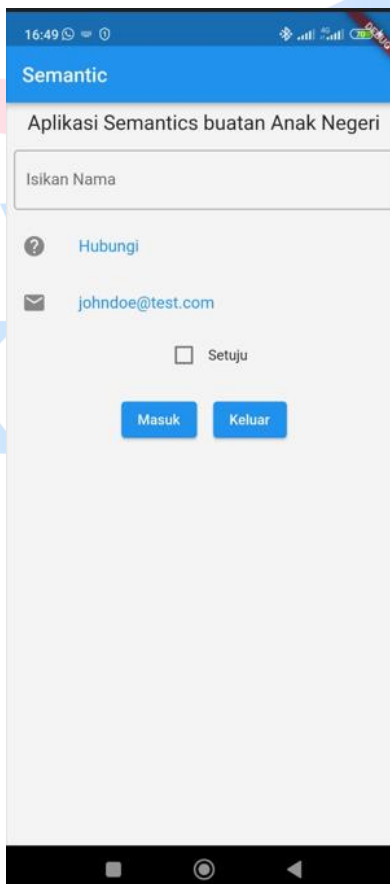
- **Kontras:** Flutter mengekspos widget sehingga pengembang dapat mewarnai widget dengan kontras yang tepat.

- **Font besar:** Di Flutter, widget teks mengikuti pengaturan Sistem Operasi saat menentukan ukuran font dan dapat diperbesar sesuai keinginan pengguna.
- **Pembaca layar:** Merupakan fitur yang telah menjadi bawaan di Sistem Operasi Android (**TalkBack**) dan iOS (**VoiceOver**) yang digunakan pengguna untuk mendapatkan umpan balik mengenai isi layar.

Semantics

Flutter menyediakan widget **Semantics** bagi pengembang untuk mengizinkan deskripsi arti widget sehingga pengguna yang tidak dapat melihat layar (tunanetra/rabun) dapat bekerja dengan aplikasi yang dibangun dengan baik.

Pada contoh kali ini, anda akan mencoba untuk menggunakan widget **Semantics** dan widget **MergeSemantics** di Flutter. Cobalah lakukan langkah-langkah berikut ini dalam menyelesaikan tampilan berikut:



1. Rancangan tampilan aplikasi berikut

```

1 import 'package:flutter/material.dart';
2
3 class Myhome extends StatefulWidget {
4   const Myhome({Key? key}) : super(key: key);
5
6   @override
7   State<Myhome> createState() => _MyhomeState();
8 }
9
10 class _MyhomeState extends State<Myhome> {
11   bool _isChecked = false;
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       appBar: AppBar(title: Text("Semantic")),
16       body: Column(
17         children: [
18           const Padding(
19             padding: EdgeInsets.all(8.0),
20             child: Text("Aplikasi Semantics buatan Anak Negeri",
21               style: TextStyle(fontSize: 20)),
22           ),
23           const Padding(
24             padding: EdgeInsets.all(8.0),
25             child: TextField(
26               decoration: InputDecoration(
27                 border: OutlineInputBorder(), labelText: 'Isikan Nama'),
28             ),
29           Padding(
30             padding: const EdgeInsets.all(18.0),
31             child: Column(
32               mainAxisAlignment: MainAxisAlignment.spaceEvenly,
33               children: <Widget>[
34                 ListTile(
35                   leading: const Icon(Icons.help),
36                   title: const Text(
37                     "Hubungi",
38                     style: TextStyle(color: Colors.blue),
39                   ),
40                   onTap: () {},
41                 ),
42                 ListTile(
43                   leading: const Icon(
44                     Icons.email,
45                   ),
46                   title: const Text(
47                     "johndoe@test.com",
48                     style: TextStyle(color: Colors.blue),
49                   ),
50                   onTap: () {},
51                 ),
52               ]),
53           Padding(
54             padding: EdgeInsets.all(18.0),
55             child: Row(
56               mainAxisAlignment: MainAxisAlignment.center,
57               children: <Widget>[
58                 Checkbox(
59                   value: _isChecked,
60                   onChanged: (value) {

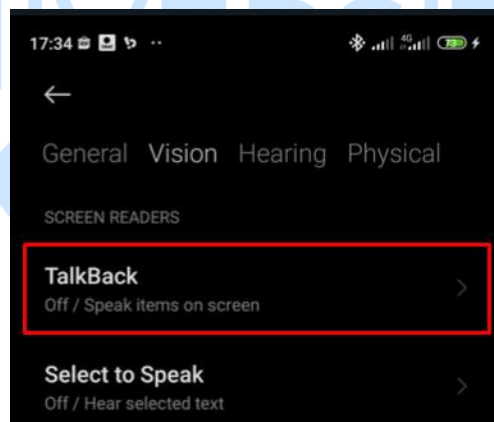
```

```

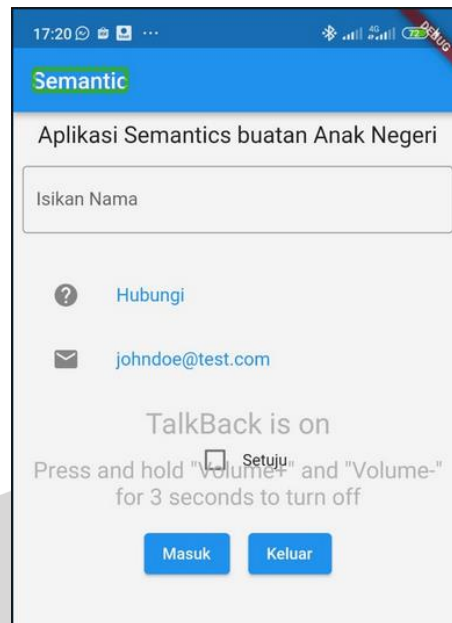
61         setState(() {
62             _isChecked = value!;
63         });
64     },
65 ),
66 Text('Setuju'),
67 ],
68 ),
69 ),
70 Padding(
71     padding: const EdgeInsets.all(8.0),
72     child: Row(
73         mainAxisAlignment: MainAxisAlignment.center,
74         children: [
75             Padding(
76                 padding: const EdgeInsets.all(8.0),
77                 child:
78                     ElevatedButton(onPressed: () {}, child: Text("Masuk")),
79             ),
80             Padding(
81                 padding: const EdgeInsets.all(8.0),
82                 child:
83                     ElevatedButton(onPressed: () {}, child: Text("Keluar")),
84             )
85         ],
86     ),
87 ),
88 ],
89 ));
90 }
91 }

```

2. Aktifkan Aksesibilitas **TalkBack** dalam **pengaturan** Android anda pada menu **Aksesibilitas** dengan tampilan berikut



Dengan mengaktifkan **TalkBack**, anda akan melihat sebuah kotak yang akan menyorot widget anda, dan membacakan komponen widget dan isinya sesuai dengan yang sedang di soroti.



Anda dapat mengatur apa yang akan di sampaikan ke pengguna, dengan menggunakan widget **Semantics** dan widget **MergeSemantics**. Widget **Semantics** digunakan hanya untuk mendeskripsikan 1 widget sedangkan widget **MergeSemantics** digunakan untuk mendeskripsikan kelompok widget yang masih saling berkaitan. Sebagai contoh, kita akan menerapkan beberapa widget semantics, pada aplikasi yang telah kita bangun.

Semantics-UI Text

Anda dapat menambahkan deskripsi widget **Text** menggunakan properti **'label'** pada widget semantics. Setiap kali Text di klik, maka Talkback bukan hanya membaca isi dari label/teks anda, tetapi juga ini properti **'label'** pada widget semantics.

```

18      Semantics(
19        label: "Judul Aplikasi",
20        child: const Padding(
21          padding: EdgeInsets.all(8.0),
22          child: Text("Aplikasi Semantics buatan Anak Negeri",
23            style: TextStyle(fontSize: 20)),
24        ),
25      ),

```

Semantics-UI Button

Anda dapat menambahkan deskripsi widget **button** menggunakan properti **'onTapHint'** pada widget semantics. Setiap kali Button di klik, maka Talkback bukan hanya membaca teks dari tombol anda, tetapi juga ini properti **'onTapHint'** pada widget semantics.

```

81         Padding(
82           padding: const EdgeInsets.all(8.0),
83           child: Row(
84             mainAxisAlignment: MainAxisAlignment.center,
85             children: [
86               Padding(
87                 padding: const EdgeInsets.all(8.0),
88                 child: ElevatedButton(
89                   onPressed: () {},
90                   child: Semantics(
91                     onTapHint: 'Ketuk 2 kali untuk masuk ke aplikasi',
92                     child: Text("Masuk"))),
93               ),
94               Padding(
95                 padding: const EdgeInsets.all(8.0),
96                 child: ElevatedButton(
97                   onPressed: () {},
98                   child: Semantics(
99                     onTapHint: 'Ketuk 2 kali untuk keluar ke aplikasi',
100                     child: Text("Keluar"))),
101               ),
102             ],
103           ),
104         ),

```

MergeSemantics-UI Checkbox-Label

Merge Semantics, digunakan untuk menggabungkan widget yang saling berhubungan, sehingga ketika widget dibaca oleh TalkBack, anda bukan hanya dibacakan isi dari 1 widget, tetapi beberapa widget, sebagai contoh:

```

60 MergeSemantics(
61   child: Row(
62     mainAxisAlignment: MainAxisAlignment.center,
63     children: <Widget>[
64       Checkbox(
65         value: _isChecked,
66         onChanged: (value) {
67           setState(() {
68             _isChecked = value!;
69           });
70       },
71     ),
72     Semantics(
73       hint: _isChecked
74         ? 'Anda Memilih untuk Setuju'

```

```

75         : "Anda Belum Setuju",
76         child: const Text('Setuju'),
77       ),
78     ],
79   ),
80 ),

```

MergeSemantics-UI ListTile

Contoh lainnya, dapat anda gunakan pada ListTile, sehingga, setiap isi ListTile dapat langsung dibacakan oleh **Talkback**, yang sebelumnya dibaca masih per item ListTile.

```

MergeSemantics(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: <Widget>[
      ListTile(
        leading: const Icon(Icons.help, semanticLabel: 'Bantuan'),
        title: const Text(
          "Hubungi",
          style: TextStyle(color: Colors.blue),
        ),
        onTap: () {},
      ),
      ListTile(
        leading: const Icon(
          Icons.email,
          semanticLabel: 'email',
        ),
        title: const Text(
          "johndoe@test.com",
          style: TextStyle(color: Colors.blue),
        ),
        onTap: () {},
      ),
    ]),
)

```

Untuk Widget **Icon**, anda dapat menambahkan semantics, tanpa perlu menggunakan widget semantics, dikarenakan pada widget icon, telah memiliki properti **semanticLabel**.

Flutter_localizations

Flutter_localizations merupakan paket yang menyediakan penerjemahan bahasa yang dapat digunakan dengan widget Flutter. Secara *default*, widget Flutter hanya disediakan dengan lokalisasi bahasa Inggris AS, jadi, untuk mendukung bahasa lain, paket flutter_localizations dapat anda gunakan. Pada contoh ini, anda akan membuat program

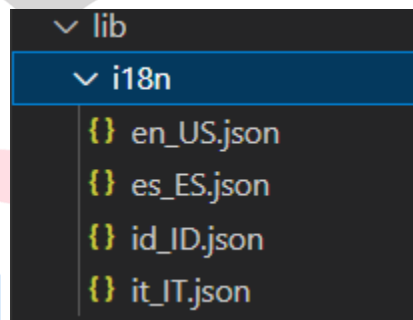
sederhana untuk menerjemahkan bahasa di dalam widget text anda ke dalam beberapa bahasa seperti Indonesia, Inggris, Spanyol dan Italia. Untuk dapat bekerja dengan localization, pastikan anda menambahkan dengan flutter_localization dan localization pada pubspec.yaml anda.

```
flutter_localizations:  
  sdk: flutter  
  version: ^0.0.0  
  localization: 2.1.0
```

Setelah itu, tambahkan sebuah **aset** yang akan digunakan untuk mengakses ke dalam file Json yang berisikan bahasa anda, dengan folder :

```
flutter:  
  assets:  
    - lib/i18n/
```

Buat folder aset anda pada folder lib, yaitu folder i18n dan buat 4 file Json pada folder i18n dengan en_US.json, es_ES.json, id_ID.json, dan it_IT.json.



Penamaan file Json, harus sesuai dengan Penanda bahasa dengan wilayah yang berlaku secara umum di seluruh dunia. Anda dapat mendapatkan semua kode penamaan bahasa dan wilayah pada <https://docs.plesk.com/en-US/obsidian/localization-guide/appendix-locale-codes.64474/>. Penamaan file Json ini harus dimulai dari kode bahasa dalam huruf kecil dan wilayah dengan huruf kapital. Setelah anda membuat file tersebut, isikan file tersebut dengan:

1. en_US.json (English - United States)

```
1. {  
2.   "Button-sign-in": "Enter",  
3.   "Button-sign-out" : "Exit",  
4.   "Checkbox-agree" : "Agree",  
5.   "Text_call" : "Call",  
6.   "Name" : "Name"
```



```
7. }
```

2. es_ES.json (Spanish – Spain)

```
1. {
2.   "Button-sign-in": "Ingresar",
3.   "Button-sign-out": "Salir",
4.   "Checkbox-agree" : "Aceptar",
5.   "Text_call" : "Contacto",
6.   "Name" : "Nombre"
7. }
```

3. id_ID.json (Indonesian – Indonesia)

```
1. {
2.   "Button-sign-in": "Masuk",
3.   "Button-sign-out": "Keluar",
4.   "Checkbox-agree" : "Setuju",
5.   "Text_call" : "Hubungi",
6.   "Name" : "Nama"
7. }
```

4. it_IT.json (Italian – Italy)

```
1. {
2.   "Button-sign-in": "Accedere",
3.   "Button-sign-out": "Esci",
4.   "Checkbox-agree" : "Concordare",
5.   "Text_call" : "Contatto",
6.   "Name" : "Nome"
7. }
```

Ubah setiap teks yang ada dapat aplikasi semantics kita dengan:

```
1 import 'package:flutter/material.dart';
2 import 'package:localization/localization.dart';
3
4 class Myhome extends StatefulWidget {
5   const Myhome({Key? key}) : super(key: key);
6
7   @override
8   State<Myhome> createState() => _MyhomeState();
9 }
10
11 class _MyhomeState extends State<Myhome> {
12   bool _isChecked = false;
13   @override
14   Widget build(BuildContext context) {
15     return Scaffold(
16       appBar: AppBar(title: Text("Semantic")),
17       body: Column(
18         children: [
19           const Padding(
20             padding: EdgeInsets.all(8.0),
```

```

21      child: Text("Aplikasi Semantics buatan Anak Negeri",
22        style: TextStyle(fontSize: 20)),
23    ),
24    Padding(
25      padding: EdgeInsets.all(8.0),
26      child: TextField(
27        decoration: InputDecoration(
28          border: OutlineInputBorder(), labelText: "Name".i18n()),
29      ),
30    Padding(
31      padding: const EdgeInsets.all(18),
32      child: Column(
33        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
34        children: <Widget>[
35          ListTile(
36            leading: const Icon(Icons.help),
37            title: Text(
38              "Text_call".i18n(),
39              style: TextStyle(color: Colors.blue),
40            ),
41            onTap: () {},
42          ),
43          ListTile(
44            leading: const Icon(
45              Icons.email,
46            ),
47            title: const Text(
48              "johndoe@test.com",
49              style: TextStyle(color: Colors.blue),
50            ),
51            onTap: () {},
52          ),
53        ]),
54    Padding(
55      padding: EdgeInsets.all(18.0),
56      child: Row(
57        mainAxisAlignment: MainAxisAlignment.center,
58        children: <Widget>[
59          Checkbox(
60            value: _isChecked,
61            onChanged: (value) {
62              setState(() {
63                _isChecked = value!;
64              });
65            },
66          ),
67          Text("Checkbox-agree".i18n()),
68        ],
69      ),
70    ),
71    Padding(
72      padding: const EdgeInsets.all(8.0),
73      child: Row(
74        mainAxisAlignment: MainAxisAlignment.center,
75        children: [
76          Padding(
77            padding: const EdgeInsets.all(8.0),
78            child: ElevatedButton(
79              onPressed: () {}, child: Text("Button-sign-in".i18n()),
80            ),

```

```

81         Padding(
82             padding: const EdgeInsets.all(8.0),
83             child: ElevatedButton(
84                 onPressed: () {},
85                 child: Text("Button-sign-out".i18n()),
86             ),
87         ),
88     ),
89 ),
90 ],
91 ));
92 }
93 }

```

Metode/fungsi **i18n()** merupakan fungsi yang telah ditambahkan ke kelas String melalui ekstensi, sehingga dengan menambahkan kunci file json akan menerjemahkan string gunakan metode **i18n()** dengan membuka terjemahan dari file json. Untuk mendapatkan penyesuaian lokasi perangkat dan bahasa yang akan digunakan perangkat anda, tambahkan pengaturan berikut pada `main.dart` anda.

1. Deklarasikan lokasi file json yang akan digunakan untuk terjemahan teks aplikasi anda

```

14 @override
15 Widget build(BuildContext context) {
16     LocalizedJsonLocalization.delegate.directories = ['lib/i18n'];
17     return MaterialApp(

```

2. Daftarkan bahasa yang di dukung oleh aplikasi anda

```

17     return MaterialApp(
18         supportedLocales: const [
19             Locale('en', 'US'),
20             Locale('es', 'ES'),
21             Locale('id', 'ID'),
22             Locale('it', 'IT'),
23         ],

```

3. Delegasikan lokasi paket untuk material, widget dan lokalisasi.

```

24         localizationsDelegates: [
25             GlobalMaterialLocalizations.delegate,
26             GlobalWidgetsLocalizations.delegate,
27             LocalizedJsonLocalization.delegate
28         ],

```

4. Terakhir tambahkan pengaturan untuk menentukan *default* bahasa yang akan anda gunakan, Ketika perangkat anda tidak menemukan terjemahan sesuai yang telah anda deklarasikan.

```

29     localeResolutionCallback: (locale, supportedLocales) {
30         if (supportedLocales.contains(locale)) {
31             return locale;
32         }
33         return const Locale('en', 'US');
34     },

```

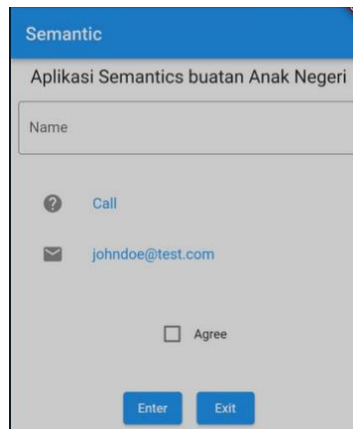
Adapun hasil akhir dari kode keseluruhan adalah:

```

6 void main() {
7     runApp(const MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11     const MyApp({Key? key}) : super(key: key);
12
13     // This widget is the root of your application.
14     @override
15     Widget build(BuildContext context) {
16         LocalizedLocalization.delegate.directories = ['lib/i18n'];
17         return MaterialApp(
18             supportedLocales: const [
19                 Locale('en', 'US'),
20                 Locale('es', 'ES'),
21                 Locale('id', 'ID'),
22                 Locale('it', 'IT'),
23             ],
24             localizationsDelegates: [
25                 GlobalMaterialLocalizations.delegate,
26                 GlobalWidgetsLocalizations.delegate,
27                 LocalizedLocalization.delegate
28             ],
29             localeResolutionCallback: (locale, supportedLocales) {
30                 if (supportedLocales.contains(locale)) {
31                     return locale;
32                 }
33                 return const Locale('en', 'US');
34             },
35             title: 'Flutter Demo',
36             theme: ThemeData(
37                 primarySwatch: Colors.blue,
38             ),
39             home: const Myhome(),
40         );
41     }
42 }

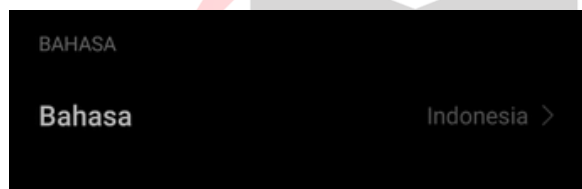
```

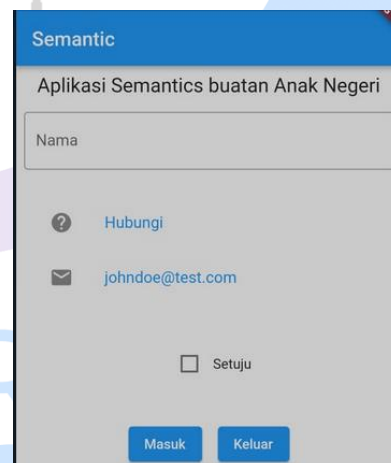
Jalankan aplikasi anda. Selama perangkat dan aplikasi anda mengenali bahasa dan lokasi yang telah anda deklarasikan, maka aplikasi anda dapat menyesuaikan bahasa yang digunakan sesuai dengan file json yang telah anda buat sebelumnya. Ketika pengaturan bahasa dan lokasi tidak sesuai dengan file json yang telah anda deklarasikan, maka secara otomatis, bahasa *default* yang akan diberikan adalah bahasa Inggris.



Ketika anda mengubah lokasi dan bahasa yang ada dalam perangkat anda, maka anda akan mendapatkan bahwa aplikasi anda telah menterjemahkan teks secara otomatis mengikuti pengaturan lokasi dan bahasa anda.

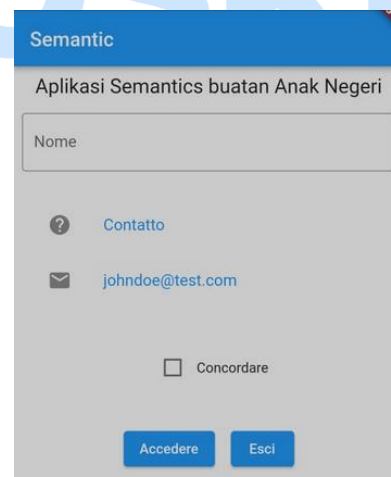
Contoh 1 :





Contoh 2 :





Latihan

1. Lakukan pembuatan widget **Semantics** minimal 5 pada widget lainnya yang belum ada pada contoh, seperti
 - a. FAB
 - b. Textbox
 - c. Radio Button
 - d. Spinner
 - e. Chips
 - f. dan lainnya
2. Lakukan proses lokalisasi, pada perangkat anda yang penentuan bahasa dilakukan melalui pengaturan aplikasi, bukan pada kondisi perangkat! Anda dapat menggunakan referensi berikut : <https://medium.com/nusanet/flutter-localization-bb22547ca653>

UNIVERSITAS
MIKROSKIL