MINGGU 5: LISTVIEW, INPUTTEXT, ALERTDIALOG, DAN SNACKBAR

Pokok Bahasan	**: ListView, InputText, AlertDialog, dan Snackbar**
Minggu	**: 5**
Tempat	**: Daring / Luring Politeknik Negeri Jember**
Alokasi Waktu	**: 100 menit x 4**

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep ListView, InputText, AlertDialog, dan Snackbar dalam Flutter
2. Mahasiswa mampu menerapkan ListView, InputText, AlertDialog, dan Snackbar dalam Flutter

b. Indikator

Kemampuan mahasiswa dalam memahami dan menerapkan ListView, InputText, AlertDialog, dan Snackbar dalam Flutter

c. Dasar Teori

ListView adalah widget yang dapat digulir (scroll) dan diatur secara linier. ListView merupakan widget yang paling umum atau banyak digunakan untuk scroll widget. Widget ini menampilkan children satu per satu dalam arah scroll. Terdapat empat opsi dalam membuat ListView: Pertama adalah dengan konstruktor default yang mengambil List<Widget> eksplisit dari children, kedua dengan konstruktor ListView.builder yang mengambil IndexedWidgetBuilder, dan membangun children sesuai permintaan. Ketiga, konstruktor ListView.separated yang mengambil dua IndexedWidgetBuilders: itemBuilder membangun item children sesuai permintaan, dan separatorBuilder juga membangun children pemisah yang muncul di antara item children. Keempat, konstruktor ListView.custom yang mengambil SliverChildDelegate, yang menyediakan kemampuan untuk menyesuaikan aspek tambahan dari model child.

Input text pada flutter bisa menggunakan widget TextField. Widget tersebut memungkinkan pengguna memasukkan teks, baik dengan keyboard perangkat keras atau dengan keyboard layar. Widget TextField dapat memanggil callback onChanged setiap kali pengguna mengubah teks dalam TextField. Apabila pengguna menunjukan kalau telah selesai menginput teks, maka Widget TextField dapan memanggil callback onSubmitted. Widget TextField membutuhkan suatu controller untuk mengkontrol teks

yang ditampilkannya, sebagai contoh untuk menentukan nilai awal dari TextField dapat digunakan controller yang sudah berisi teks tersebut. Controller juga dapat mengontrol area selection.

AlertDialog adalah widget yang menginformasikan pengguna tentang situasi yang membutuhkan acknowledgement. AlertDialog memiliki judul dan juga aksi (action). Judul ditampilkan di atas konten dan aksi (action) ditampilkan di bawah konten.

Snackbar adalah widget yang berisi pesan ringan dengan aksi (action) opsional yang ditampilkan secara singkat di bagian bawah layar. Widget snackbar ini dapat ditampilkan dengan memanggil ScaffoldMessenger.of(context).showSnackBar(), melewati sebuah instance dari SnackBar yang menjelaskan pesan. Snackbar juga dapat dikontrol berapa lama durasi untuk tetap terlihat.

d. Alat dan Bahan

1. Java Development Kit (JDK)
2. Android Studio;
3. Android SDK;
4. Flutter SDK;
5. Teks Editor (atau bisa juga pakai Android Studio atau Visual Studio Code).

e. Prosedur Kerja

1. Buatlah flutter project baru dengan nama listview
2. Kemudian sama seperti prosedur kerja praktikum sebelumnya jalankan MaterialApp di dalam void main(), dengan title "List_Input_Alert_Snack" dan PageOne() sebagai home.
3. Selanjutnya tambahkan widget Scaffold dengan widget AppBar didalamnya seperti biasa pada class PageOne dengan meng-extends StatelessWidget. Warna background merah dan title berisi widget Text ("listview ").
4. Setelah itu tambahkan widget ListView di dalam body seperti pada potongan source code di bawah ini.

```
body: ListView(
  children: const <Widget>[
    ListTile(
      leading: Icon(Icons.home),
      title: Text("Home"),
    ),
    ListTile(
      leading: Icon(Icons.favorite),
      title: Text("Fave"),
```

```
    ),
    ListTile(
      leading: Icon(Icons.place),
      title: Text("Place"),
    ),
  ],
),
```

5. Jalankan dan perhatikan hasilnya. Secara umum widget Listview dengan widget Card hampir mirip. Kedua widget tersebut dapat diimplementasikan dalam class dengan parameter variabel tertentu.
6. Untuk mengimplementasikan widget InputText, AlertDialog, dan Snackbar gantilah class pageOne yang sebelumnya meng-extends StatelessWidget menjadi StatefulWidget.
7. Setelah itu tambahkan children widget TextField dan Text di bagian body dalam child widget Column dan widget Container seperti pada potongan source code di bawah.

```
body: Container(
  child: Column(
    children: <Widget>[
      TextField(
        decoration: const InputDecoration(
          border: OutlineInputBorder(),
          hintText: "Ketik untuk tampilkan Teks di sini",
        ),
        onChanged: (String str) {
          setState((){
            teks=str;
          });
        },
      ),
      Text(
        teks, style: const TextStyle(fontSize: 20.0),
      )
    ],
  ),
),
```

8. Deklarasikan variabel teks dengan variabel String di awal class _PageOneState.
9. Jalankan dan perhatikan hasilnya. Potongan source code di atas menunjukan widget TextField memiliki property onChanged yang berarti setiap ada perubahan di TextField otomatis akan langsung merubah variabel teks yang ada pada widget Text.
10. Gantilah property onChanged tersebut dengan property onSubmitted, jalankan dan perhatikan hasilnya.
11. Widget TextField juga bisa ditambahkn controller, yaitu TextEditingController. Potongan source code berikut adalah deklarasi variabel controller untuk widget TextField.

```
TextEditingController controller = TextEditingController();
```

12. Kemudian di dalam widget TextField tambahkan property controller dan sesuaikan menjadi seperti pada potongan source code berikut ini.

```
TextField(
  controller: controller,
  decoration: const InputDecoration(
    border: OutlineInputBorder(),
    hintText: "Ketik untuk tampilkan Teks di sini",
  ),
  onSubmitted: (String str) {
    setState((){
      teks=str + '\n' + teks;
      controller.text="";
    });
  },
),
```

13. Jalankan dan perhatikan hasilnya.
14. Selanjutnya buatlah kembali sebuah TextField yang sama, kemudian ganti method setState() dengan _alertdialog seperti pada potongan source code di bawah ini.

```
TextField(
  controller: controller,
  decoration: const InputDecoration(
    border: OutlineInputBorder(),
    hintText: "Ketik untuk tampilkan Alert di sini",
  ),
  onSubmitted: (String str) {
    _alertDialog(str);
    controller.text = "";
  },
),
```

15. Lalu, berikut ini adalah source code untuk method _alertDialog.

```
void _alertdialog(String str) {
  if (str.isEmpty) return;
  AlertDialog alertDialog = AlertDialog(
    title: const Text("Alert"),
    content: Text(
      str,
      style: const TextStyle(fontSize: 20.0),
    ),
  );
  showDialog(
    context: context,
    builder: (BuildContext context) => alertDialog,
  );
}
```

16. Jalankan dan perhatikan hasilnya.
17. Pada AlertDialog juga bisa ditambahkan action yang didalamnya berisi widget Button, dan pada saat Button tersebut ditekan akan dilakukan suatu action. Berikut adalah contoh potongan source code modifikasi dari method _alertdialog dengan FloatingActionButton.

```
void _alertdialog(String str) {
  if (str.isEmpty) return;
  AlertDialog alertDialog = AlertDialog(
    title: const Text(
      "Alert",
      style: TextStyle(fontSize: 20.0),
    ),
    content: Text(
      str,
      style: const TextStyle(fontSize: 12.0),
    ),
    actions: <Widget>[
      FloatingActionButton(
        onPressed: () {
          Navigator.pop(context);
        },
        backgroundColor: Colors.red,
        child: const Text("OK"),
      )
    ],
  );
  showDialog(
    context: context,
    builder: (BuildContext context) => alertDialog,
  );
}
```

18. Selanjutnya buatlah kembali sebuah TextField yang sama, kemudian ganti method _alertdialog dengan _snackbar seperti pada potongan source code di bawah ini.

```
TextField(
  controller: controller,
  decoration: const InputDecoration(
    border: OutlineInputBorder(),
    hintText: "Ketik untuk tampilkan Snackbar di sini",
  ),
  onSubmitted: (String str) {
    _snackBar(str);
    controller.text = "";
  },
),
```

19. Lalu, berikut ini adalah source code untuk method _snackBar.

```
void _snackBar(String str) {
  if (str.isEmpty) return;
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(
    duration: const Duration(seconds: 5),
    content: Text(
      str,
      style: const TextStyle(fontSize: 20.0),
    ),
  ));
}
```

20. Jalankan dan perhatikan hasilnya.

f. Hasil dan Pembahasan

1. Dokumentasi langkah prosedur kerja praktikum berupa laporan (penjelasan mengenai potongan source-code ListView, InputText, AlertDialog, dan Snackbar).
2. Dokumentasi langkah prosedur kerja praktikum berupa file pdf.

g. Kesimpulan

Mahasiswa memahami mengenai ListView, InputText, AlertDialog, dan Snackbar dalam flutter.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas ditunjang dengan bukti referensi	1	2	3	(4)
2	Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas	1	2	(3)	4
3	Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas	1	(2)	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas	(1)	2	3	4
Jumlah skor					