Modul Praktikum 7

Interactive Widgets

A. Tujuan

Setelah melakukan kegiatan praktikum ini, mahasiswa dapat memahami konsep intractive widgets sehingga dapat memberikan pengalaman yang lebih bagi pengguna. Widget ini digunakan untuk menerima input atau interaksi dari user terhadap aplikasi. Beberapa jenis interactive widgets yang akan dipelajari dalam praktikum ini yaitu:

- Gesture-based widgets
- Time and date pitcker widgets
- Input and selection widgets
- Dismissable widgets

Serta mengimplementasikannya ke dalam studi kasus aplikasi **Fooderlich.**

B. Peralatan Pendukung

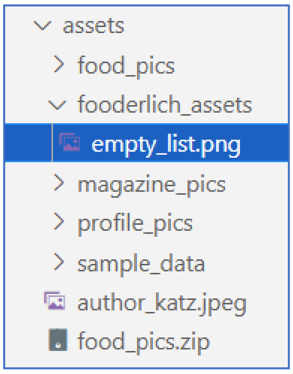
Perangkat keras yang dibutuhkan adalah laptop atau PC yang sudah ter-install SDK Flutter dan IDE VS Code atau Android Studio.

C. Materi

Adapun praktikum kali ini, Anda akan melanjutkan proyek aplikasi Fooderlich yang telah dikembangkan sebelumnya. Kali ini Anda akan melengkapi screen pada item bottom navigation **To Buy**.

D. Langkah Kerja

1. Buka kembali proyek aplikasi **Fooderlich**, kemudian tambahkan beberapa folder berikut **fooderlich_assets** di dalam folder **assets** yang sebelumnya telah dibuat, kemudian tambahkan file **empty_list.png** seperti berikut.



```
assets
  food_pics
  fooderlich_assets
    empty_list.png
  magazine_pics
  profile_pics
  sample_data
  author_katz.jpeg
  food_pics.zip
```

2. Selanjutnya buat file **grocery_item.dart** didalam folder **models** dan sesuaikan kodenya seperti berikut.

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

enum Importance { low, medium, high }

class GroceryItem {
  final String id;
  final String name;
  final Importance importance;
  final Color color;
  final int quantity;
  final DateTime date;
  final bool isComplete;

  GroceryItem(
      {this.id,
      this.name,
      this.importance,
      this.color,
      this.quantity,
      this.date,
      this.isComplete = false});

  GroceryItem copyWith(
      {String id,
      String name,
      Importance importance,
      Color color,
      int quantity,
      DateTime  date,
      bool isComplete}) {
    return GroceryItem(
        id: id ?? this.id,
        name: name ?? this.name,
        importance: importance ?? this.importance,
        color: color ?? this.color,
        quantity: quantity ?? this.quantity,
        date: date ?? this.date,
        isComplete: isComplete ?? this.isComplete);
  }
}
```

3. Jangan lupa tambahkan file **grocery_item.dart** tadi kedalam file **models.dart**, seperti berikut.

```
export 'explore_recipe.dart';
export 'post.dart';
export 'simple_recipe.dart';
export 'explore_data.dart';
export 'grocery_item.dart';
```

4. Kemudian buat screen untuk menampilkan daftar kebutuhan yang akan dibelanjakan, dimana didalam screen tersebut akan terdapat 2 kondisi yaitu empty screen, jika belum ada item yang ditambahkan dan list screen yang akan menampilkan item yang sudah ditambahkan. Buat file **empty_grocery_screen.dart** di dalam folder **screen**, dan sesuaikan kodenya seperti berikut.

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

class EmptyGroceryScreen extends StatelessWidget {
  const EmptyGroceryScreen({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // 1
    return Padding(
      padding: const EdgeInsets.all(30.0),
      // 2
      child: Center(
        // 3
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            AspectRatio(
              aspectRatio: 1 / 1,
              child: Image.asset('assets/fooderlich_assets/empty_list.png')
,
            ),
            const SizedBox(height: 8.0),
            const Text(
              'No Groceries',
              style: TextStyle(fontSize: 21.0),
            ),
            const SizedBox(height: 16.0),
            const Text(
              'Shopping for ingredients?\n'
              'Tap the + button to write them down!',
              textAlign: TextAlign.center,
            ),
            MaterialButton(
              textColor: Colors.white,
              child: const Text('Browse Recipes'),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(30.0),
              ),
              color: Colors.green,
              onPressed: () {},
            ),
          ],
        ),
      ),
    );
  }
}
```

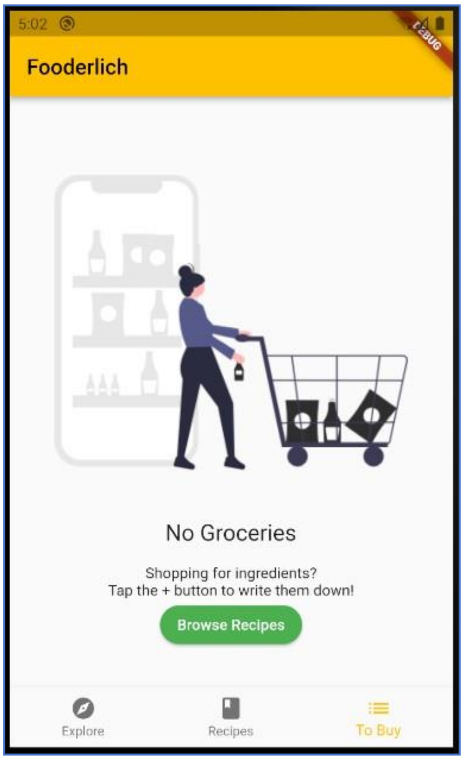
5. Selanjutnya buat file **grocery_screen.dart** didalam folder **screen** seperti berikut.

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

class GroceryScreen extends StatelessWidget {
  const GroceryScreen({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const EmptyGroceryScreen();
  }
}
```

6. Kemudian tambahkan file **grocery_screen.dart** ke dalam file **home.dart**, untuk menampilkan grocery screen pada item botom navigation **To Buy**, berikut potongan kodenya.

```
import 'package:flutter/material.dart';
import 'screens/explore_screen.dart';
import 'screens/recipes_screen.dart';
import 'screens/grocery_screen.dart';

class Home extends StatefulWidget {
  const Home({Key key}) : super(key: key);
  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  int _selectedIndex = 0;

  static List<Widget> pages = <Widget>[
    ExploreScreen(),
    RecipesScreen(),
    const GroceryScreen()
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }
```

7. Simpan dan jalankan aplikasi, sekarang item bottom navigation **To Buy** jika kita tap akan menampilkan empty screen seperti berikut.



No Groceries

Shopping for ingredients?
Tap the + button to write them down!

8. Selanjutnya untuk mengelola **state** di aplikasi seperti yang sudah dibahas pada sesi perkuliahan, kiata akan menggunakan **Provider**, tambahkan package **provider** yang akan digunakan untuk menyimpan **state**, buka file **pubspec.yaml** lalu sesuaikan seperti berikut.

```
dependencies:
  flutter:
    sdk: flutter

  google_fonts: ^2.1.0
  provider: ^5.0.0
```

9. Kemudian buat file **tab_manager.dart** di dalam folder **models** sesuaikan kodenya seperti berikut.

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

class TabManager extends ChangeNotifier {
  int selectedTab = 0;

  void goToTab(index) {
    selectedTab = index;
    notifyListeners();
  }

  void goToRecipes() {
    selectedTab = 1;
    notifyListeners();
  }
}
```

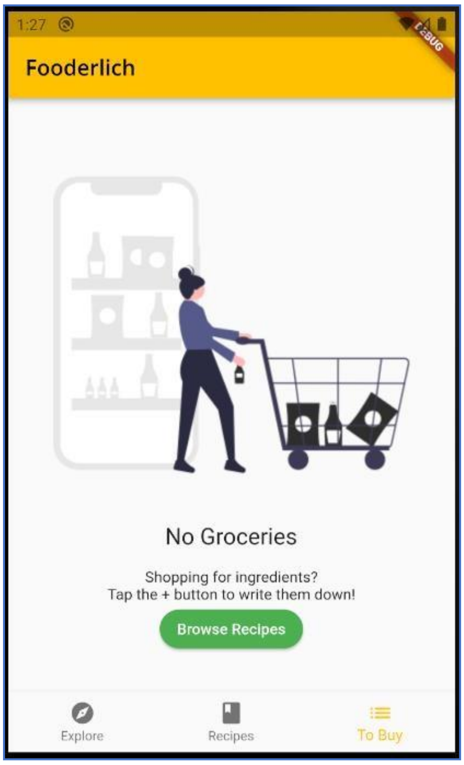
10. Lalu tambahkan file **tab_manager.dart** ke dalam file **models.dart** seperti berikut.

```
export 'explore_recipe.dart';
export 'post.dart';
export 'simple_recipe.dart';
export 'explore_data.dart';
export 'grocery_item.dart';
export 'tab_manager.dart';
```

11. Kemudian buka kembali file **empty_grocery_screen.dart** lalu tambahkan potongan kode aksi pada tombol hijau **Browse Recipes**, seperti berikut (warna kuning).

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

class EmptyGroceryScreen extends StatelessWidget {
  const EmptyGroceryScreen({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // 1
    return Padding(
   padding: const EdgeInsets.all(30.0),
   // 2
   child: Center(
    // 3
    child: Column(
     mainAxisAlignment: MainAxisAlignment.center,
     children: [
      AspectRatio(
       aspectRatio: 1 / 1,
       child: Image.asset('assets/fooderlich_assets/empty_list.png'),
      ),
      const SizedBox(height: 8.0),
      const Text(
       'No Groceries',
       style: TextStyle(fontSize: 21.0),
      ),
      const SizedBox(height: 16.0),
      const Text(
       'Shopping for ingredients?\n'
       'Tap the + button to write them down!',
       textAlign: TextAlign.center,
      ),
      MaterialButton(
       textColor: Colors.white,
       child: const Text('Browse Recipes'),
       shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30.0),
       ),
       color: Colors.green,
       onPressed: () {
        Provider.of<TabManager>(context, listen: false).goToRecipes();
       },
      ),
     ],
    ),
   ),
  );
 }
}
```

12. Simpan dan jalankan aplikasi, kini tombol **Browse Recipes** pada empty screen jika ditap akan menuju recipes screen.

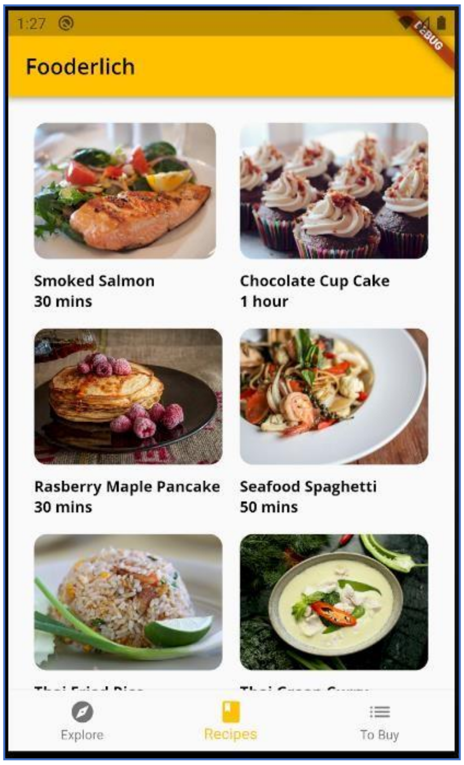


1:27

Fooderlich

No Groceries

Shopping for ingredients?
Tap the + button to write them down!



1:27

Fooderlich

Smoked Salmon
30 mins

Chocolate Cup Cake
1 hour

Rasberry Maple Pancake
30 mins

Seafood Spaghetti
50 mins

13. Kemudian selanjutnya adalah membuat tampilan grocery items, namun sebelumnya Anda perlu menambahkan kelas untuk mengelolanya (insert, update, delete). Buat file **grocery_manager.dart** di dalam folder **models** sesuaikan kodenya seperti berikut.

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

class GroceryManager extends ChangeNotifier {
  final _groceryItems = <GroceryItem>[];

  List<GroceryItem> get groceryItems => List.unmodifiable(_groceryItems);

  void deleteItem(int index) {
    _groceryItems.removeAt(index);
    notifyListeners();
  }

  void addItem(GroceryItem item) {
    _groceryItems.add(item);
    notifyListeners();
  }

  void updateItem(GroceryItem item, int index) {
    _groceryItems[index] = item;
    notifyListeners();
  }

  void completeItem(int index, bool change) {
    final item = _groceryItems[index];
    _groceryItems[index] = item.copyWith(isComplete: change);
    notifyListeners();
  }
}
```

14. Lalu tambahkan file **grocery_manager.dart** ke dalam file **models.dart** seperti berikut.

```
export 'explore_recipe.dart';
export 'post.dart';
export 'simple_recipe.dart';
export 'explore_data.dart';
export 'grocery_item.dart';
export 'tab_manager.dart';
export 'grocery_manager.dart';
```

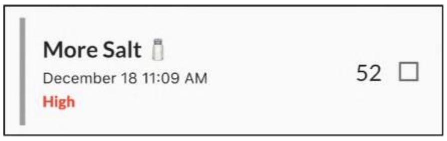
15. Sama sepeti pada TabManager untuk menyimpan state perubahan pada grocery list, maka **GroceryManager** ditambahkan sebagai provider pada file **main.dart**, berikut adalah kodenya.

```
import 'package:flutter/material.dart';
import 'fooderlich_theme.dart';
import 'home.dart';
import 'package:provider/provider.dart';
import 'models/models.dart';

void main() {
  runApp(const Fooderlich());
}

class Fooderlich extends StatelessWidget {
  const Fooderlich({Key key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    final theme = FooderlichTheme.light();

    return MaterialApp(
      title: 'Fooderlich',
      theme: theme,
      home: MultiProvider(
        providers: [
          ChangeNotifierProvider(
            create: (context) => TabManager(),
          ),
          ChangeNotifierProvider(
            create: (context) => GroceryManager(),
          ),
        ],
        child: const Home(),
      ),
    );
  }
}
```

16. Kemudian selanjutnya kita siapkan screen untuk menambahkan item grocery. Namun sebelumnya perlu menambahkan beberapa package pada file **pubscpec.yaml** seperti berikut.

```
dependencies:
  flutter:
    sdk: flutter

  google_fonts: ^2.1.0
  provider: ^5.0.0
  flutter_colorpicker: ^0.4.0
  intl: ^0.17.0
  uuid: ^3.0.4
```

17. Sebelum membuat tampilan item grocery kita perlu membuat komponen GroceryTile untuk membuat tampilan seperti berikut.



More Salt
December 18 11:09 AM
High
52 □

18. Buat file **grocery_tile.dart** di dalam folder **lib/component**, sesuaikan kodenya seperti berikut.

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:intl/intl.dart';
import '../models/grocery_item.dart';

class GroceryTile extends StatelessWidget {
  final GroceryItem item;

  final Function(bool) onComplete;

  final TextDecoration textDecoration;

  GroceryTile({Key key, this.item, this.onComplete})
      : textDecoration =
            item.isComplete ? TextDecoration.lineThrough : TextDecoration.n
one,
        super(key: key);

  @override
  Widget build(BuildContext context) {
    return SizedBox(
      height: 100,
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Row(
            children: [
              Container(width: 5.0, color: item.color),
              const SizedBox(width: 16.0),
              Column(
                mainAxisAlignment: MainAxisAlignment.center,
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    item.name,
                    style: GoogleFonts.lato(
                        decoration: textDecoration,
                        fontSize: 21.0,
                        fontWeight: FontWeight.bold),
                  ),
                  const SizedBox(height: 4.0),
                  buildDate(),
                  const SizedBox(height: 4.0),
                  buildImportance(),
                ],
              ),
            ],
          ),
          Row(
            children: [
              Text(item.quantity.toString(),
                  style: GoogleFonts.lato(
                      decoration: textDecoration, fontSize: 21)),
              buildCheckbox()
            ],
          )
        ],
      ),
    );
  }

  Widget buildImportance() {
    if (item.importance == Importance.low) {
      return Text('Low', style: GoogleFonts.lato(decoration: textDecoration
));
    } else if (item.importance == Importance.medium) {
      return Text('Medium',
          style: GoogleFonts.lato(
       fontWeight: FontWeight.w800, decoration: textDecoration));
  } else if (item.importance == Importance.high) {
   return Text(
    'High',
    style: GoogleFonts.lato(
     color: Colors.red,
     fontWeight: FontWeight.w900,
     decoration: textDecoration,
    ),
   );
  } else {
   throw Exception('This importance type does not exist');
  }
 }

 Widget buildDate() {
  final dateFormatter = DateFormat('MMMM dd h:mm a');
  final dateString = dateFormatter.format(item.date);
  return Text(
   dateString,
   style: TextStyle(decoration: textDecoration),
  );
 }

 Widget buildCheckbox() {
  return Checkbox(
   // 1
   value: item.isComplete,
   // 2
   onChanged: onComplete,
  );
 }
}
```

19. Lalu buat file **grocey_item_screen.dart** di dalam folder **lib/screens**, sesuaikan kode seperti berikut.

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:flutter_colorpicker/flutter_colorpicker.dart';
import 'package:intl/intl.dart';
import 'package:uuid/uuid.dart';
import '../models/models.dart';
import '../components/grocery_tile.dart';

class GroceryItemScreen extends StatefulWidget {
 final Function(GroceryItem) onCreate;

 final Function(GroceryItem) onUpdate;
```

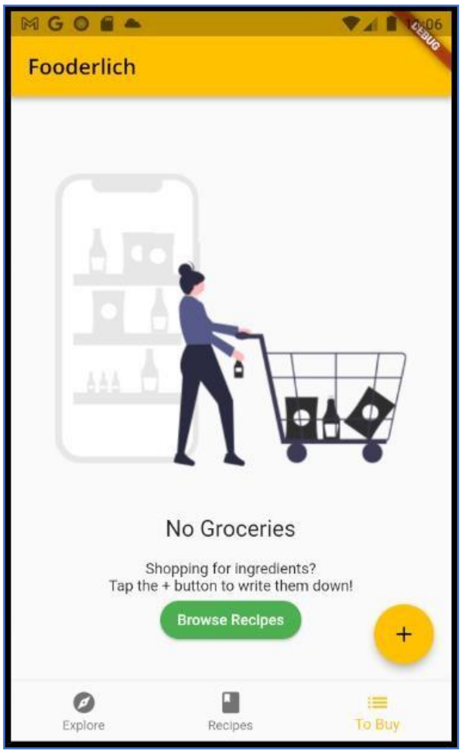
```
  final GroceryItem originalItem;

  final bool isUpdating;

  const GroceryItemScreen({
    Key  key,
    this.onCreate,
    this.onUpdate,
    this.originalItem,
  }) : isUpdating = (originalItem != null),
        super(key: key);

  @override
  _GroceryItemScreenState createState() => _GroceryItemScreenState();
}

class _GroceryItemScreenState extends State<GroceryItemScreen> {
  final _nameController = TextEditingController();
  String _name = '';
  Importance _importance = Importance.low;
  DateTime _dueDate = DateTime.now();
  TimeOfDay _timeOfDay = TimeOfDay.now();
  Color _currentColor = Colors.green;
  int _currentSliderValue = 0;

  @override
  void initState() {
    if (widget.originalItem != null) {
      _nameController.text = widget.originalItem.name;
      _name = widget.originalItem.name;
      _currentSliderValue = widget.originalItem.quantity;
      _importance = widget.originalItem.importance;
      _currentColor = widget.originalItem.color;
      final date = widget.originalItem.date;
      _timeOfDay = TimeOfDay(hour: date.hour, minute: date.minute);
      _dueDate = date;
    }

    _nameController.addListener(() {
      setState(() {
        _name = _nameController.text;
      });
    });

    super.initState();
  }

  @override
```

```
void dispose() {
  _nameController.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      actions: [
        IconButton(
          icon: const Icon(Icons.check),
          onPressed: () {
            final groceryItem = GroceryItem(
                id: widget.originalItem?.id ?? const Uuid().v1(),
                name: _nameController.text,
                importance: _importance,
                color: _currentColor,
                quantity: _currentSliderValue,
                date: DateTime(
                  _dueDate.year,
                  _dueDate.month,
                  _dueDate.day,
                  _timeOfDay.hour,
                  _timeOfDay.minute,
                ));

            if (widget.isUpdating) {
              widget.onUpdate(groceryItem);
            } else {
              widget.onCreate(groceryItem);
            }
          },
        )
      ],
      elevation: 0.0,
      title: Text(
        'Grocery Item',
        style: GoogleFonts.lato(fontWeight: FontWeight.w600),
      ),
    ),
    body: Container(
      padding: const EdgeInsets.all(16),
      child: ListView(
        children: [
          buildNameField(),
          buildImportanceField(),
          buildDateField(context),
          buildTimeField(context),
          const SizedBox(height: 10.0),
          buildColorPicker(context),
          const SizedBox(height: 10.0),
          buildQuantityField(),
          const SizedBox(height: 16.0),
          GroceryTile(
            item: GroceryItem(
              name: _name,
              importance: _importance,
              color: _currentColor,
              quantity: _currentSliderValue,
              date: DateTime(
                _dueDate.year,
                _dueDate.month,
                _dueDate.day,
                _timeOfDay.hour,
                _timeOfDay.minute,
              ),
            ),
          ),
        ],
      ),
    ),
  );
}

Widget buildNameField() {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Item Name',
        style: GoogleFonts.lato(fontSize: 28.0),
      ),
      TextField(
        controller: _nameController,
        cursorColor: _currentColor,
        decoration: InputDecoration(
          enabledBorder: const UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.white),
          ),
          focusedBorder: UnderlineInputBorder(
            borderSide: BorderSide(color: _currentColor),
          ),
          border: UnderlineInputBorder(
            borderSide: BorderSide(color: _currentColor),
          ),
        ),
      ),
    ],
  );
}

Widget buildImportanceField() {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Importance',
        style: GoogleFonts.lato(fontSize: 28.0),
      ),
      Wrap(
        spacing: 10.0,
        children: [
          ChoiceChip(
            selectedColor: Colors.black,
            selected: _importance == Importance.low,
            label: const Text(
              'low',
              style: TextStyle(color: Colors.white),
            ),
            onSelected: (selected) {
              setState(() => _importance = Importance.low);
            },
          ),
          ChoiceChip(
            selectedColor: Colors.black,
            selected: _importance == Importance.medium,
            label: const Text(
              'medium',
              style: TextStyle(color: Colors.white),
            ),
            onSelected: (selected) {
              setState(() => _importance = Importance.medium);
            },
          ),
          ChoiceChip(
            selectedColor: Colors.black,
            selected: _importance == Importance.high,
            label: const Text(
              'high',
              style: TextStyle(color: Colors.white),
            ),
            onSelected: (selected) {
              setState(() => _importance = Importance.high);
            },
          ),
        ],
      )
    ],
  );
}

Widget buildDateField(BuildContext context) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Text(
            'Date',
            style: GoogleFonts.lato(fontSize: 28.0),
          ),
          TextButton(
            child: const Text('Select'),
            onPressed: () async {
              final currentDate = DateTime.now();

              final selectedDate = await showDatePicker(
                context: context,
                initialDate: currentDate,
                firstDate: currentDate,
                lastDate: DateTime(currentDate.year + 5),
              );

              setState(
                () {
                  if (selectedDate != null) {
                    _dueDate = selectedDate;
                  }
                },
              );
            },
          ),
        ],
      ),
      if (_dueDate != null)
        Text('${DateFormat('yyyy-MM-dd').format(_dueDate)}'),
    ],
  );
}

Widget buildTimeField(BuildContext context) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Text(
            'Time of Day',
            style: GoogleFonts.lato(fontSize: 28.0),
          ),
          TextButton(
            child: const Text('Select'),
            onPressed: () async {
              final timeOfDay = await showTimePicker(
                initialTime: TimeOfDay.now(),
                context: context,
              );

              setState(() {
                if (timeOfDay != null) {
                  _timeOfDay = timeOfDay;
                }
              });
            },
          ),
        ],
      ),
      if (_timeOfDay != null) Text('${_timeOfDay.format(context)}'),
    ],
  );
}

Widget buildColorPicker(BuildContext context) {
  return Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Row(
        children: [
          Container(height: 50, width: 10, color: _currentColor),
          const SizedBox(width: 8),
          Text('Color', style: GoogleFonts.lato(fontSize: 28)),
        ],
      ),
      TextButton(
        child: const Text('Select'),
        onPressed: () {
          showDialog(
            context: context,
            builder: (context) {
              return AlertDialog(
                content: BlockPicker(
                    pickerColor: Colors.white,
                    onColorChanged: (color) {
                      setState(() => _currentColor = color);
                    }),
                actions: [
                  TextButton(
                    child: const Text('Save'),
                    onPressed: () {
                      Navigator.of(context).pop();
                    },
                  ),
                ],
              );
            },
          );
        },
      ),
    ],
  );
}

Widget buildQuantityField() {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Row(
        crossAxisAlignment: CrossAxisAlignment.baseline,
        textBaseline: TextBaseline.alphabetic,
        children: [
          Text(
            'Quantity',
            style: GoogleFonts.lato(fontSize: 28.0),
          ),
          const SizedBox(width: 16.0),
          Text(
            _currentSliderValue.toInt().toString(),
            style: GoogleFonts.lato(fontSize: 18.0),
          ),
        ],
      ),
      Slider(
        inactiveColor: _currentColor.withOpacity(0.5),
        activeColor: _currentColor,
        value: _currentSliderValue.toDouble(),
        min: 0.0,
        max: 100.0,
        divisions: 100,
        label: _currentSliderValue.toInt().toString(),
        onChanged: (double value) {
          setState(() {
              _currentSliderValue = value.toInt();
            });
          },
        ),
      ],
    );
  }
}
```

20. Selanjutnya tambahkan FloatingActionButton pada GroceryScreen dimana aksinya jika tombol tersebut ditap akan menuju tampilan GroceryItemScreen. Buka kembali file **grocery_screen.dart** yang ada di dalam folder **lib/screens** tambahkan kode berikut.

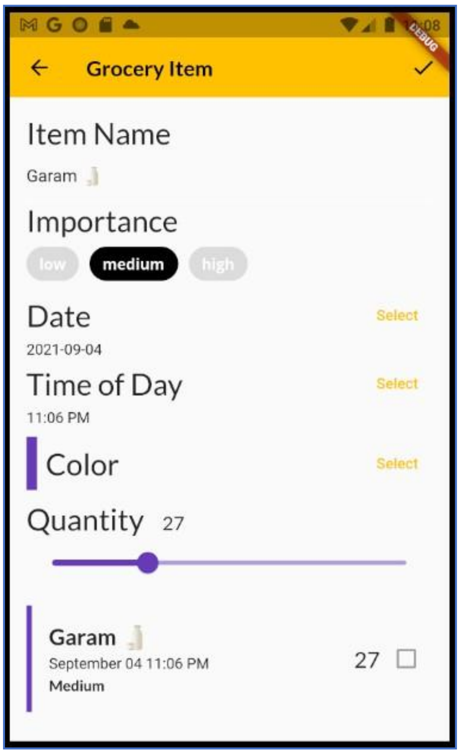
```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'empty_grocery_screen.dart';
import 'grocery_item_screen.dart';
import '../models/models.dart';

class GroceryScreen extends StatelessWidget {
  const GroceryScreen({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(
        child: const Icon(Icons.add),
        onPressed: () {
          final manager = Provider.of<GroceryManager>(context, listen: fals
e);
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => GroceryItemScreen(
                onCreate: (item) {
                  manager.addItem(item);
                  Navigator.pop(context);
                },
              ),
            ),
          );
        },
      ),
      body: buildGroceryScreen(),
    );
  }

  Widget  buildGroceryScreen()  {
    return const EmptyGroceryScreen();
  }
}
```

21. Simpan kemudian jalankan aplikasi, pada screen **To Buy** saat ini sudah terdapat tombol "+" untuk menambahkan grocery item.



No Groceries

Shopping for ingredients?
Tap the + button to write them down!



Item Name
Garam

Importance: medium

Date
2021-09-04

Time of Day
11:06 PM

Color

Quantity 27

Garam
September 04 11:06 PM
Medium
27 □

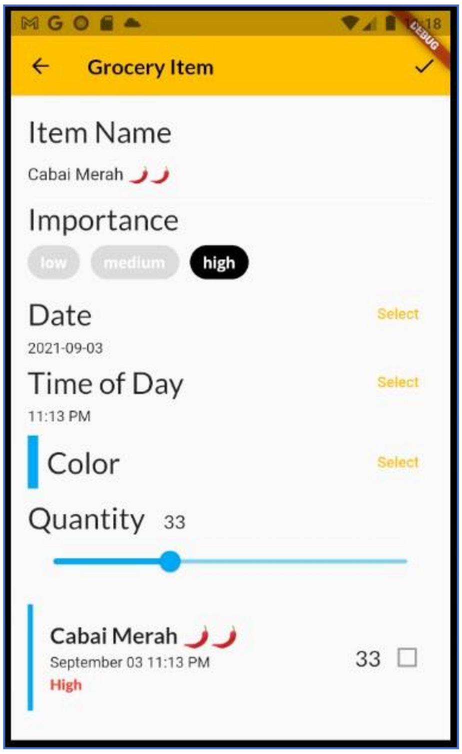
22. Selanjutnya untuk menampilkan list grocery item yang sudah ditambahkan buat file **grocery_list_screen.dart** di dalam folder **lib/screens** sesuaikan kodenya seperti berikut.

```
import 'package:flutter/material.dart';
import '../components/grocery_tile.dart';
import '../models/models.dart';
import 'grocery_item_screen.dart';

class GroceryListScreen extends StatelessWidget {
  final GroceryManager manager;

  const GroceryListScreen({Key key, this.manager}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final groceryItems = manager.groceryItems;

    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: ListView.separated(
          itemCount: groceryItems.length,
          itemBuilder: (context, index) {
            final item = groceryItems[index];

            return Dismissible(
              key: Key(item.id),
              direction: DismissDirection.endToStart,
              background: Container(
                color: Colors.red,
                alignment: Alignment.centerRight,
                child: const Icon(
                  Icons.delete_forever,
                  color: Colors.white,
                  size: 50.0,
                ),
              ),
              onDismissed: (direction) {
                manager.deleteItem(index);

                ScaffoldMessenger.of(context).showSnackBar(
                  SnackBar(
                    content: Text('${item.name} dismissed'),
                  ),
                );
              },
              child: InkWell(
                child: GroceryTile(
                  key: Key(item.id),
                  item: item,
                  onComplete: (change) {
                    manager.completeItem(index, change);
                  },
                ),
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) => GroceryItemScreen(
                        originalItem: item,
                        onUpdate: (item) {
                          manager.updateItem(item, index);

                          Navigator.pop(context);
                        },
                      ),
                    ),
                  );
                },
              ),
            );
          },
          separatorBuilder: (context, index) {
            return const SizedBox(height: 16.0);
          }),
    );
  }
}
```

23. Selanjutnya pada GroceryScreen tambahkan kode untuk menampilkan GroceryListScreen. Buka kembali file **grocery_screen.dart** di dalam folder **lib/screens** kemudian sesuaikan kodenya seperti berikut (warna kuning).

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'empty_grocery_screen.dart';
import 'grocery_item_screen.dart';
import 'grocery_list_screen.dart';
import '../models/models.dart';

class GroceryScreen extends StatelessWidget {
  const GroceryScreen({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(
        child: const Icon(Icons.add),
        onPressed: () {
          final manager = Provider.of<GroceryManager>(context, listen: fals
e);
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => GroceryItemScreen(
                onCreate: (item) {
                  manager.addItem(item);
                  Navigator.pop(context);
                },
              ),
            ),
          );
        },
      ),
      body: buildGroceryScreen(),
    );
  }

  Widget buildGroceryScreen() {
    return Consumer<GroceryManager>(
      builder: (context, manager, child) {
        if (manager.groceryItems.isNotEmpty) {
          return GroceryListScreen(manager: manager);
        } else {
          return const EmptyGroceryScreen();
        }
      },
    );
  }
}
```

24. Simpan kemudian jalankan aplikasi, apabila kita telah menambahkan grocery item maka hasilnya akan ditampilkan pada grocery list.



Grocery Item

Item Name

Cabai Merah 🌶️🌶️

Importance

low medium [✓ high]

Date Select

2021-09-03

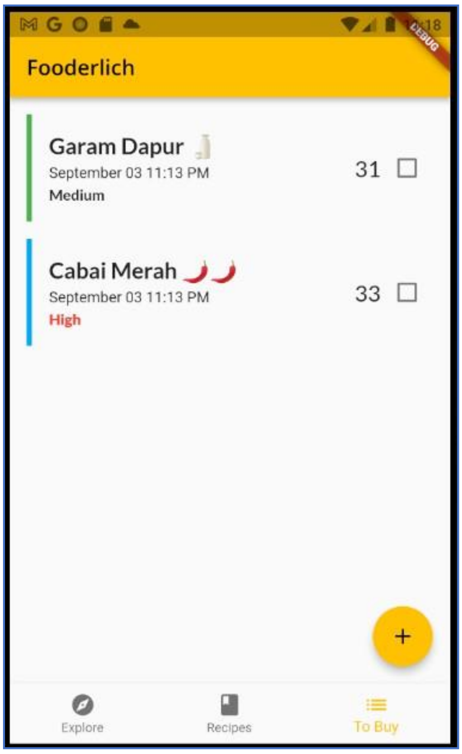
Time of Day Select

11:13 PM

Color Select

Quantity 33

Cabai Merah 🌶️🌶️
September 03 11:13 PM
High
33 □

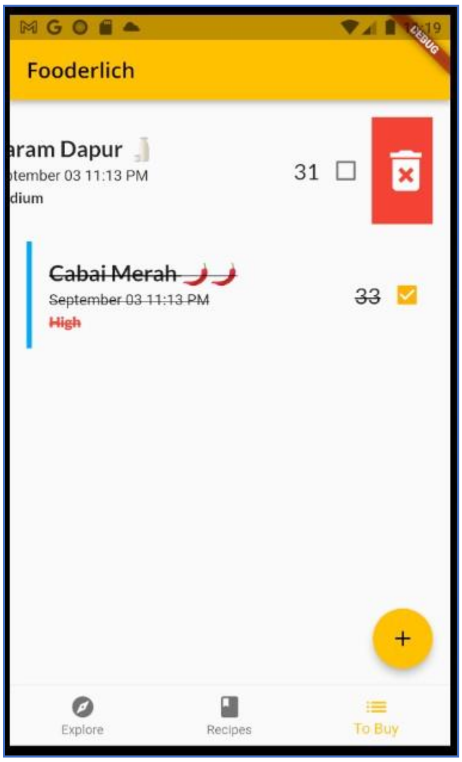


Fooderlich

Garam Dapur 🧂
September 03 11:13 PM
Medium
31 □

Cabai Merah 🌶️🌶️
September 03 11:13 PM
High
33 □

25. Pada list grocery juga telah ditambahkan **gesture** jika kita melakukan slide ke kiri, maka item grocery akan terhapus. Selain itu kita juga dapat melakukan ceklist serta melakukan edit pada item grocery dengan menekan GroceryTile.



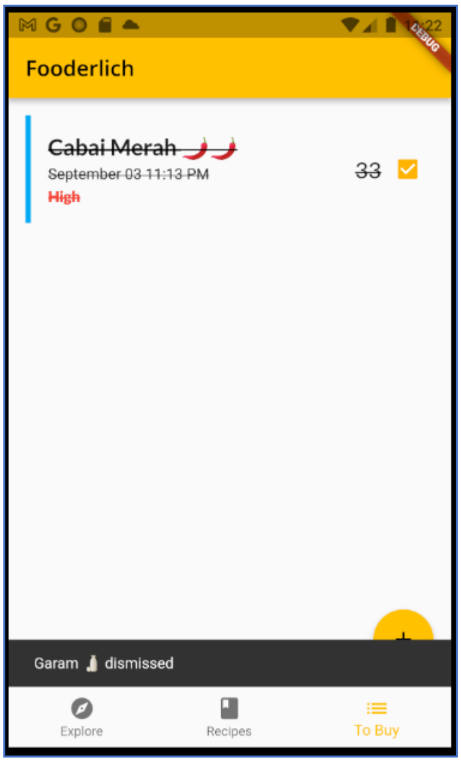
Fooderlich

aram Dapur
ptember 03 11:13 PM
dium

31 □

~~Cabai Merah~~
~~September 03 11:13 PM~~
~~High~~

~~33~~ ☑



Fooderlich

~~Cabai Merah~~
~~September 03 11:13 PM~~
~~High~~

~~33~~ ☑

Garam dismissed

26. Selamat sampai tahap ini Anda telah meningkatkan pemahaman dengan mengimplementasikan interactive widgets pada aplikasi **Fooderlich**.

LEMBAR KERJA PRAKTIKUM (LKP)

Nama	
NPM/NIM	
Asal Perguruan Tinggi	

Untuk lembar kerja praktikum kali ini, silakan buat video implementasi hasil praktikum Anda, durasi video maksimal 2 menit dengan menampilkan seluruh interkasi dan gesture pada aplikasi, kemudian video pada **vclass.unila.ac.id.**