



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Manegement
Mata Kuliah Pemograman Mobile

Ketentuan;

Simpanlah file lembar jawaban ini dengan format; **Kelas_Nama Lengkap**

Contoh; **TI 3C_Rifki Fakhruhin**

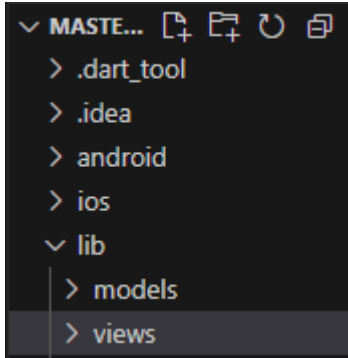
Upload tugas jobsheet ini dengan batas maksimum sesuai jadwal perkuliahan masing masing kelas

Upload file tugas jobsheet di website **lms.polinema**

Nama	:	Rifki Fakhruhin
Nim	:	2241720218
Kelas	:	3C

Tulislah Jawaban Pada Kolom Yang tersedia di bawah ini;

LEMBAR JAWABAN JOBSHEET-11

Soal No	Jawaban
1	Praktikum 1 Langkah 1: Buat Project Baru 
2	Langkah 2: Membuat model task.dart



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
task.dart x plan.dart data_layer.dart
lib\SEMESTER 5\Mobile2024\master_plan\lib\models\task.dart
1 class Task {
2   final String description;
3   final bool complete;
4
5   const Task({
6     this.complete = false,
7     this.description = '',
8   });
9 }
```

3 Langkah 3: Buat file **plan.dart**

```
task.dart x plan.dart x data_layer.dart main.dart
lib > models > plan.dart > Plan
1 import './task.dart';
2
3 class Plan {
4   final String name;
5   final List<Task> tasks;
6
7   const Plan({this.name = '', this.tasks = const []});
8 }
```

4 Langkah 4: Buat file **data_layer.dart**

```
task.dart plan.dart data_layer.dart x
lib > models > data_layer.dart
1 export 'plan.dart';
2 export 'task.dart';
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

5 Langkah 5: Pindah ke file **main.dart**

```
task.dart x plan.dart data_layer.dart main.dart 2 x plan_screen.dart
lib > main.dart > MasterPlanApp
1 import 'package:flutter/material.dart';
2 import './views/plan_screen.dart';
3
Run | Debug | Profile
4 void main() => runApp(MasterPlanApp());
5
6 class MasterPlanApp extends StatelessWidget {
7   const MasterPlanApp({super.key});
8
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      theme: ThemeData(primarySwatch: Colors.purple),
13      home: PlanScreen(),
14    ); // MaterialApp
15  }
16 }
```

6 Langkah 6: buat **plan_screen.dart**

```
task.dart plan.dart data_layer.dart main.dart plan_screen.dart 2 x
lib > views > plan_screen.dart > _PlanScreenState
1 import '../models/data_layer.dart';
2 import 'package:flutter/material.dart';
3
4 class PlanScreen extends StatefulWidget {
5   const PlanScreen({super.key});
6
7   @override
8   State createState() => _PlanScreenState();
9 }
10
11 class _PlanScreenState extends State<PlanScreen> {
12   Plan plan = const Plan();
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       // ganti 'Namaku' dengan Nama panggilan Anda
18       appBar: AppBar(title: const Text('Master Plan Namaku')),
19       body: _buildlist(),
20       floatingActionButton: _buildAddTaskButton(),
21     ); // Scaffold
22   }
23 }
```

7 Langkah 7: buat method **_buildAddTaskButton()**



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
Widget _buildAddTaskButton() {  
  return FloatingActionButton(  
    child: const Icon(Icons.add),  
    onPressed: () {  
      setState(() {  
        plan = Plan(  
          name: plan.name,  
          tasks: List<Task>.from(plan.tasks)..add(const Task(description: 'New Task', complete: false)),  
        ); // Plan  
      });  
    },  
  ); // FloatingActionButton  
}
```

8 Langkah 8: buat widget **_buildList()**

```
Widget _buildList() {  
  return ListView.builder(  
    itemCount: plan.tasks.length,  
    itemBuilder: (context, index) => _buildTaskTile(plan.tasks[index], index),  
  );  
}
```

9 Langkah 9: buat widget **_buildTaskTile**



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
Widget _buildTaskTile(Task task, int index) {  
  return ListTile(  
    leading: Checkbox(  
      value: task.complete,  
      onChanged: (selected) {  
        setState(() {  
          plan = Plan(  
            name: plan.name,  
            tasks: List<Task>.from(plan.tasks)  
              ..[index] = Task(  
                description: task.description,  
                complete: selected ?? false,  
              ), // Task  
          ); // Plan  
        });  
      },  
    ), // Checkbox
```

```
      title: TextFormField(  
        initialValue: task.description,  
        onChanged: (text) {  
          setState(() {  
            plan = Plan(  
              name: plan.name,  
              tasks: List<Task>.from(plan.tasks)  
                ..[index] = Task(  
                  description: text,  
                  complete: task.complete,  
                ), // Task  
            ); // Plan  
          });  
        },  
      ), // TextFormField  
    ); // ListTile  
  }  
}
```

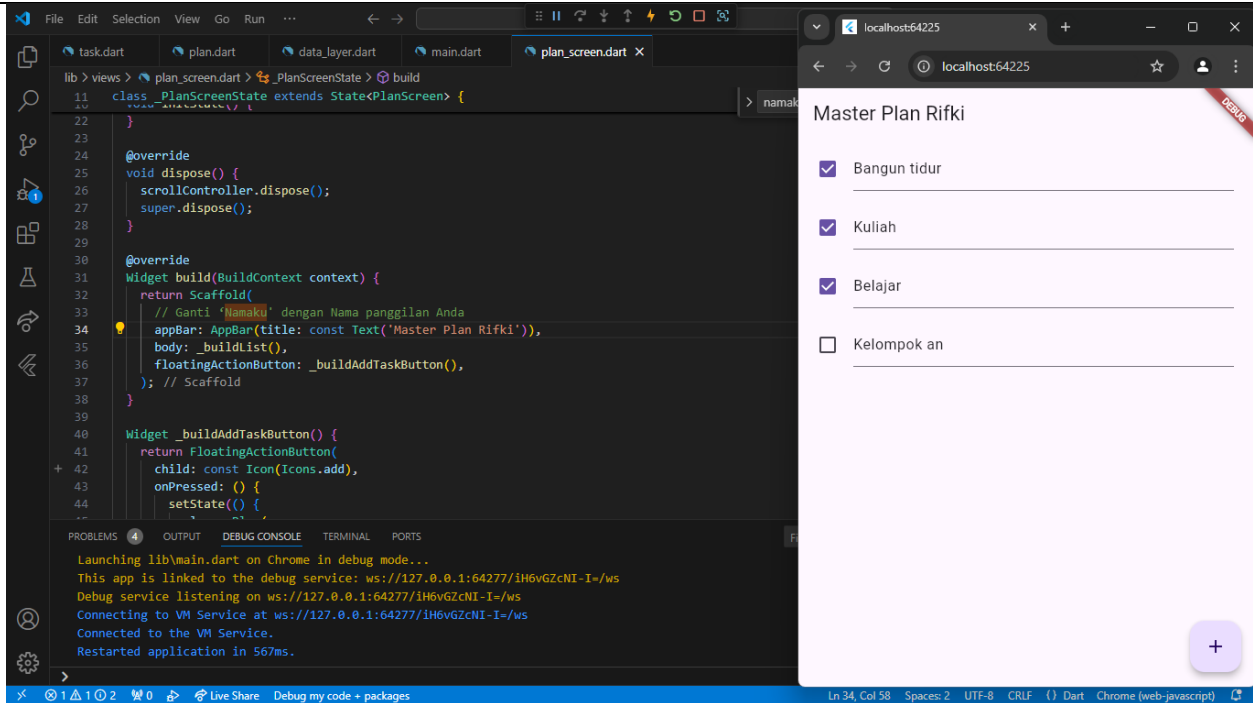


Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Manegement
Mata Kuliah Pemograman Mobile

	Langkah 10: Tambah Scroll Controller <pre>class _PlanScreenState extends State<PlanScreen> { Plan plan = const Plan(); late ScrollController scrollController;</pre>
11	Langkah 11: Tambah Scroll Listener <pre>@override void initState() { super.initState(); scrollController = ScrollController() ..addListener(() { FocusScope.of(context).requestFocus(FocusNode()); }); }</pre>
12	Langkah 12: Tambah controller dan keyboard behavior <pre>Widget _buildList() { return ListView.builder(controller: scrollController, itemCount: plan.tasks.length, itemBuilder: (context, index) => _buildTaskTile(plan.tasks[index], index), keyboardDismissBehavior: Theme.of(context).platform == TargetPlatform.iOS ? ScrollViewKeyboardDismissBehavior.onDrag : ScrollViewKeyboardDismissBehavior.manual,); }</pre>
13	Langkah 13: Terakhir, tambah method dispose() <pre>@override void dispose() { scrollController.dispose(); super.dispose(); }</pre>
14	Langkah 14: Hasil



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile



15 **Tugas Praktikum 1**

1. Selesaikan langkah-langkah praktikum tersebut, lalu dokumentasikan berupa GIF hasil akhir praktikum beserta penjelasannya di file README.md! Jika Anda menemukan ada yang error atau tidak berjalan dengan baik, silakan diperbaiki.
2. Jelaskan maksud dari langkah 4 pada praktikum tersebut! Mengapa dilakukan demikian?
 - Langkah 4 dimaksudkan untuk berfungsi sebagai pusat ekspor untuk model yang relevan, seperti plan.dart dan task.dart. Dilakukan untuk metode yang berfungsi, memproses impor menjadi lebih efisien, bersih, dan lebih mudah untuk mengelola lapisan data, sehingga dapat mengurangi risiko kesalahan impor ganda dan meningkatkan keakuratan kode.

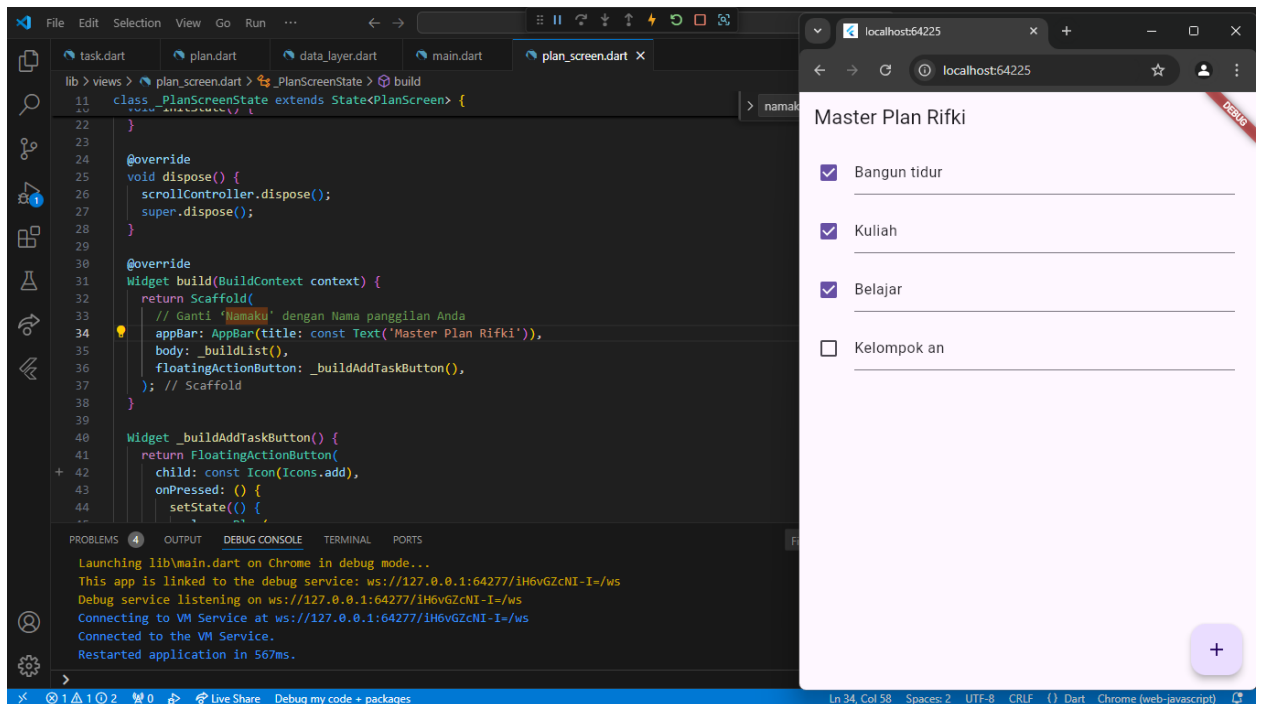


Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

3. Mengapa perlu variabel plan di langkah 6 pada praktikum tersebut? Mengapa dibuat konstanta ?

- Variabel plan menyimpan dan mengelola data yang akan ditampilkan dan diubah pada layar PlanScreen. Variabel ini dibuat sebagai konstanta sehingga nilai awalnya tetap dan tidak berubah, yang membantu menjaga aplikasi stabil sebelum pengguna memasukkan atau mengubah data.

4. Lakukan capture hasil dari Langkah 9 berupa GIF, kemudian jelaskan apa yang telah Anda buat!



5. Apa kegunaan method pada Langkah 11 dan 13 dalam *lifecyle state* ?

- Langkah 11: Mengaktifkan kontrol scroll saat layar dibuka dengan initState()
- Langkah 13: Membersihkan kontrol saat layar ditutup untuk menjaga aplikasi tetap beroperasi dengan baik.

6. Kumpulkan laporan praktikum Anda berupa link commit atau repository GitHub ke spreadsheet yang telah disediakan!



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

16 **Praktikum 2**

Langkah 1: Buat file plan_provider.dart

```
task.dart x plan.dart data_layer.dart main.dart plan_screen.dart plan_provider.dart x
lib > provider > plan_provider.dart > PlanProvider
1 import 'package:flutter/material.dart';
2 import '../models/data_layer.dart';
3
4 class PlanProvider extends InheritedNotifier<ValueNotifier<Plan>> {
5   const PlanProvider({super.key, required Widget child, required
6     ValueNotifier<Plan> notifier})
7     : super(child: child, notifier: notifier);
8
9   static ValueNotifier<Plan> of(BuildContext context) {
10     return context.
11       dependOnInheritedWidgetOfExactType<PlanProvider>()!.notifier!;
12   }
13 }
```

17 **Langkah 2: Edit main.dart**

```
return MaterialApp(
  theme: ThemeData(primarySwatch: Colors.purple),
  home: PlanProvider(
    notifier: ValueNotifier<Plan>(const Plan()),
    child: const PlanScreen(),
  ), // PlanProvider
); // MaterialApp
}
```

18 **Langkah 3: Tambah method pada model plan.dart**

```
int get completedCount => tasks
  .where((task) => task.complete)
  .length;

String get completenessMessage =>
  '$completedCount out of ${tasks.length} tasks';
}
```

19 **Langkah 4: Pindah ke PlanScreen**



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
class _PlanScreenState extends State<PlanScreen> {  
  // Plan plan = const Plan();  
  late ScrollController scrollController;
```

20 **Langkah 5: Edit method _buildAddTaskButton**

```
Widget _buildAddTaskButton(BuildContext context) {  
  ValueNotifier<Plan> planNotifier = PlanProvider.of(context);  
  return FloatingActionButton(  
    child: const Icon(Icons.add),  
    onPressed: () {  
      Plan currentPlan = planNotifier.value;  
      planNotifier.value = Plan(  
        name: currentPlan.name,  
        tasks: List<Task>.from(currentPlan.tasks)..add(const Task()),  
      ); // Plan  
    },  
  ); // FloatingActionButton  
}
```

21 **Langkah 6: Edit method _buildTaskTile**

```
Widget _buildTaskTile(Task task, int index, BuildContext context) {  
  ValueNotifier<Plan> planNotifier = PlanProvider.of(context);  
  return ListTile(  
    leading: Checkbox(  
      value: task.complete,  
      onChanged: (selected) {  
        Plan currentPlan = planNotifier.value;  
        planNotifier.value = Plan(  
          name: currentPlan.name,  
          tasks: List<Task>.from(currentPlan.tasks)  
            ..[index] = Task(  
              description: task.description,  
              complete: selected ?? false,  
            ),  
        );  
      }),  
    title: TextFormField(  
      title: task.description,
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
 Mata Kuliah Pemograman Mobile

	<pre> initialValue: task.description, onChanged: (text) { Plan currentPlan = planNotifier.value; planNotifier.value = Plan(name: currentPlan.name, tasks: List<Task>.from(currentPlan.tasks) ..[index] = Task(description: text, complete: task.complete,),); },),);); } </pre>
22	Langkah 7: Edit _buildList <pre> Widget _buildList(Plan plan) { return ListView.builder(controller: scrollController, itemCount: plan.tasks.length, itemBuilder: (context, index) => _buildTaskTile(plan.tasks[index], index, context),); } </pre>
23	Langkah 8: Tetap di class PlanScreen
24	Langkah 9: Tambah widget SafeArea



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Master Plan Rifki')),
    body: ValueListenableBuilder<Plan>(
      valueListenable: PlanProvider.of(context),
      builder: (context, plan, child) {
        return Column(
          children: [
            Expanded(child: _buildList(plan)),
            SafeArea(child: Text(plan.completenessMessage))
          ],
        ); // Column
      },
    ), // ValueListenableBuilder
    floatingActionButton: _buildAddTaskButton(context),
  ); // Scaffold
}
```

25 **Tugas 2**

1. Selesaikan langkah-langkah praktikum tersebut, lalu dokumentasikan berupa GIF hasil akhir praktikum beserta penjelasannya di file README.md! Jika Anda menemukan ada yang error atau tidak berjalan dengan baik, silakan diperbaiki sesuai dengan tujuan aplikasi tersebut dibuat.
2. Jelaskan mana yang dimaksud InheritedWidget pada langkah 1 tersebut! Mengapa yang digunakan InheritedNotifier?
 - Pada langkah pertama, PlanProvider berfungsi sebagai InheritedWidget, yang memungkinkan widget anak mengakses data tanpa harus mengirim data secara eksplisit ke setiap widget. Salah satu alasan mengapa widget ini dipilih adalah karena, selain mewarisi fungsi InheritedWidget, widget akan secara otomatis memperbarui tampilannya saat nilai notifier berubah, sehingga semua widget anak yang bergantung pada data Plan juga akan diperbarui setiap kali nilai notifier diperbarui.



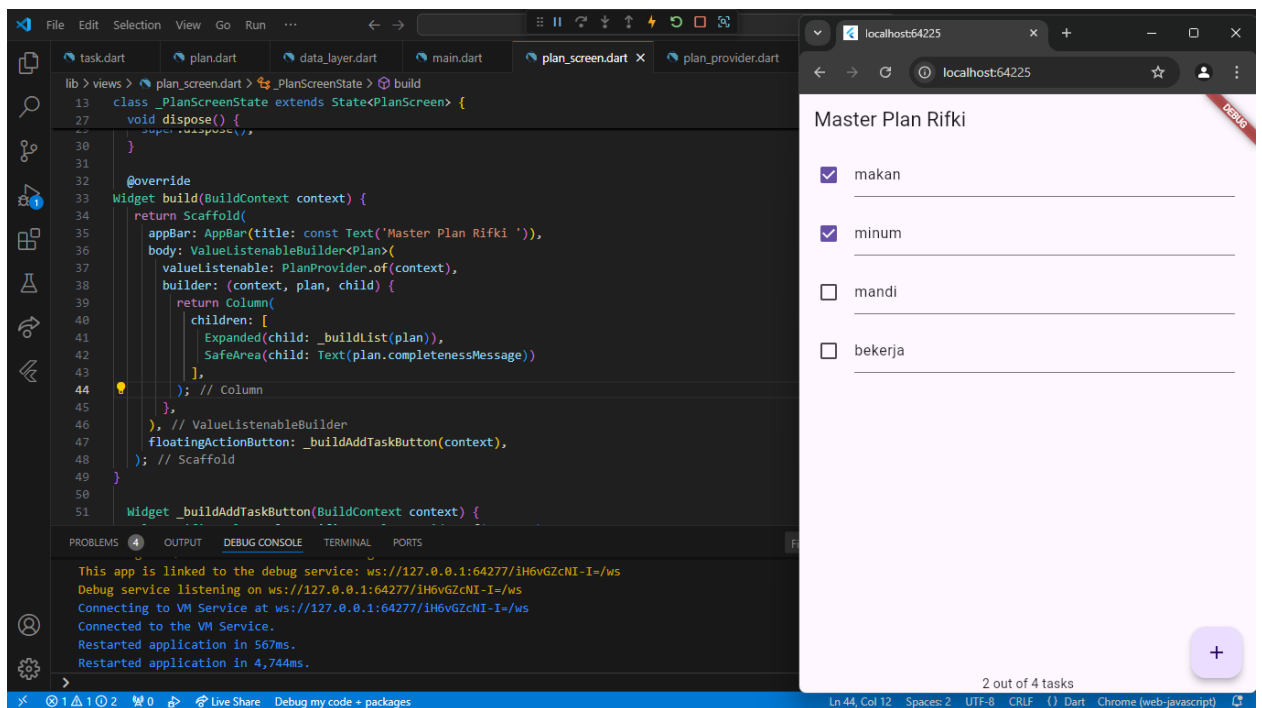
Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

3. Jelaskan maksud dari method di langkah 3 pada praktikum tersebut! Mengapa dilakukan demikian?

- "CompletedCount" menunjukkan jumlah tugas yang telah selesai dalam daftar tugas, dan "CompletenessMessage" menunjukkan jumlah tugas yang telah selesai dibandingkan dengan jumlah total tugas.

untuk memudahkan pemantauan kemajuan rencana secara langsung dan memberikan informasi tentang status penyelesaian tugas yang jelas pada tampilan aplikasi tanpa perlu menghitung lagi

4. Lakukan capture hasil dari Langkah 9 berupa GIF, kemudian jelaskan apa yang telah Anda buat!



5. Kumpulkan laporan praktikum Anda berupa link commit atau repository GitHub ke spreadsheet yang telah disediakan!

26 **Praktikum 3**
Langkah 1: Edit PlanProvider



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
 Mata Kuliah Pemograman Mobile

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

class PlanProvider extends
InheritedNotifier<ValueNotifier<List<Plan>>> {}
  const PlanProvider({super.key, required Widget child, required
ValueNotifier<List<Plan>> notifier})
    : super(child: child, notifier: notifier);

  static ValueNotifier<List<Plan>> of(BuildContext context) {
    return context.
  }
  dependOnInheritedWidgetOfExactType<PlanProvider>()!.notifier!;
}
```

27 Langkah 2: Edit main.dart

```
return PlanProvider(
  notifier: ValueNotifier<List<Plan>>(const []),
  child: MaterialApp(
    title: 'State management app',
    theme: ThemeData(
      primarySwatch: Colors.blue,
    ), // ThemeData
    home: const PlanScreen(),
  ), // MaterialApp
); // PlanProvider
}
```

28 Langkah 3: Edit plan_screen.dart

```
class PlanScreen extends StatefulWidget {
  final Plan plan;
  const PlanScreen({super.key, required this.plan});
```

29 Langkah 4: Error



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
 Mata Kuliah Pemograman Mobile

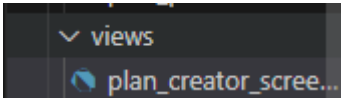
	<p>Itu akan terjadi error setiap kali memanggil <code>PlanProvider.of(context)</code>. Itu terjadi karena screen saat ini hanya menerima tugas-tugas untuk satu kelompok Plan, tapi sekarang <code>PlanProvider</code> menjadi list dari objek plan tersebut.</p>
30	<p>Langkah 5: Tambah getter Plan</p> <pre>class _PlanScreenState extends State<PlanScreen> { late ScrollController scrollController; Plan get plan => widget.plan; }</pre>
31	<p>Langkah 6: Method initState()</p> <pre>@override void initState() { super.initState(); scrollController = ScrollController() ..addListener(() { FocusScope.of(context).requestFocus(FocusNode()); }); }</pre>
32	<p>Langkah 7: Widget build</p> <pre>@override Widget build(BuildContext context) { ValueNotifier<List<Plan>> plansNotifier = PlanProvider.of(context); return Scaffold(appBar: AppBar(title: Text(widget.plan.name)), body: ValueListenableBuilder<List<Plan>>(valueListenable: plansNotifier, builder: (context, plans, child) { Plan currentPlan = plans.firstWhere((p) => p.name == widget.plan.name); return Column(children: [Expanded(child: _buildList(currentPlan)), SafeArea(child: Text(currentPlan.completenessMessage)),],); },),); }</pre>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
    );  
  },  
),  
floatingActionButton: ValueListenableBuilder<List<Plan>>(  
  valueListenable: plansNotifier,  
  builder: (context, plans, child) {  
    Plan currentPlan =  
      plans.firstWhere((p) => p.name == widget.plan.name);  
    return _buildAddTaskButton(context, currentPlan);  
  },  
),  
);  
}  
  
Widget _buildAddTaskButton(BuildContext context, Plan currentPlan) {  
  ValueNotifier<List<Plan>> planNotifier = PlanProvider.of(context);  
  
  return FloatingActionButton(  
    child: const Icon(Icons.add),  
    onPressed: () {  
      int planIndex =  
        planNotifier.value.indexWhere((p) => p.name == currentPlan.name);  
      if (planIndex >= 0) {  
        List<Plan> updatedPlans = List<Plan>.from(planNotifier.value);  
        List<Task> updatedTasks = List<Task>.from(currentPlan.tasks);  
  
        updatedTasks.add(Task(description: '', complete: false));  
  
        updatedPlans[planIndex] = Plan(  
          name: currentPlan.name,  
          tasks: updatedTasks,  
        );  
        planNotifier.value = updatedPlans;  
      }  
    },  
  );  
}
```




33	Langkah 8: Edit _buildTaskTile <pre> Widget _buildTaskTile(Task task, int index, BuildContext context, Plan currentPlan) { ValueNotifier<List<Plan>> planNotifier = PlanProvider.of(context); return ListTile(leading: Checkbox(value: task.complete, onChanged: (selected) { int planIndex = planNotifier.value.indexWhere((p) => p.name == currentPlan.name); if (planIndex >= 0) { List<Plan> updatedPlans = List<Plan>.from(planNotifier.value); List<Task> updatedTasks = List<Task>.from(currentPlan.tasks); updatedTasks[index] = Task(description: task.description, complete: selected ?? false,); updatedPlans[planIndex] = Plan(name: currentPlan.name, tasks: updatedTasks,); planNotifier.value = updatedPlans; } },),); } </pre>
34	Langkah 9: Buat screen baru 
35	Langkah 10: Pindah ke class _PlanCreatorScreenState



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
@override  
void dispose() {  
  textController.dispose();  
  super.dispose();  
}
```

36 Langkah 11: Pindah ke method build

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    // ganti 'Namaku' dengan nama panggilan Anda  
    appBar: AppBar(title: const Text('Master Plans Rifki')),  
    body: Column(children: [  
      _buildListCreator(),  
      Expanded(child: _buildMasterPlans())  
    ]), // Column  
  ); // Scaffold  
}
```

37 Langkah 12: Buat widget _buildListCreator

```
Widget _buildListCreator() {  
  return Padding(  
    padding: const EdgeInsets.all(20.0),  
    child: Material(  
      color: Theme.of(context).cardColor,  
      elevation: 10,  
      child: TextField(  
        controller: textController,  
        decoration: const InputDecoration(  
          labelText: 'Add a plan',  
          contentPadding: EdgeInsets.all(20)), // InputDecoration  
        onEditingComplete: addPlan), // TextField  
      ) // Material  
    ); // Padding  
}
```

38 Langkah 13: Buat void addPlan()



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
void addPlan() {  
    final text = textController.text;  
    if (text.isEmpty) {  
        return;  
    }  
    final plan = Plan(name: text, tasks: []);  
    ValueNotifier<List<Plan>> planNotifier = PlanProvider.of(context);  
    planNotifier.value = List<Plan>.from(planNotifier.value)..add(plan);  
    textController.clear();  
    FocusScope.of(context).requestFocus(FocusNode());  
    setState(() {});  
}
```

39 **Langkah 14: Buat widget _buildMasterPlans()**

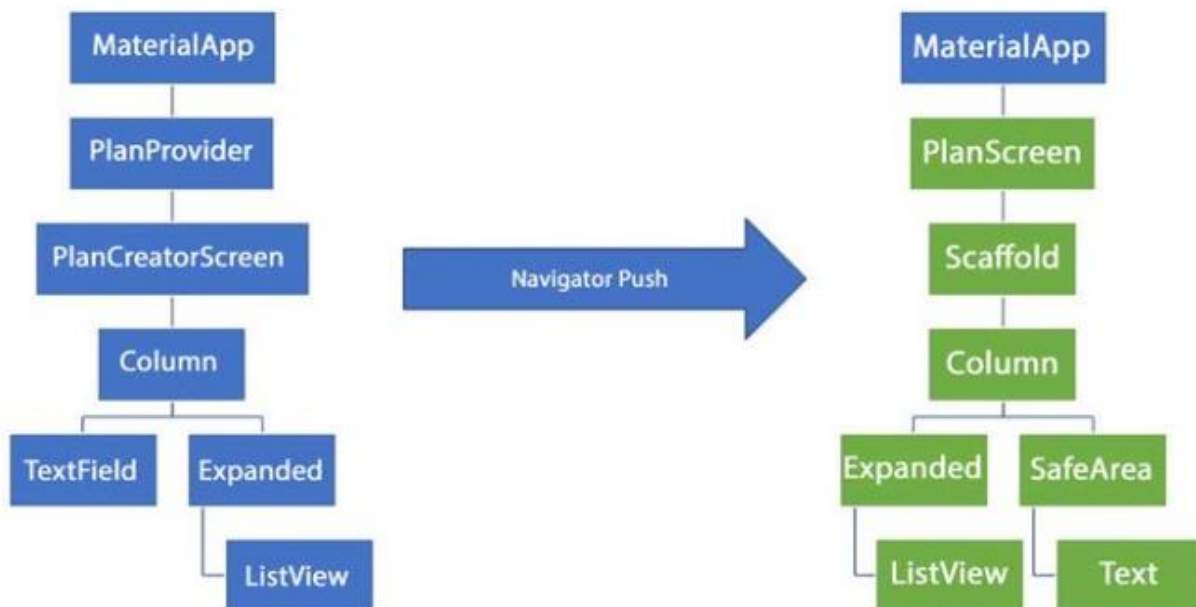
```
Widget _buildMasterPlans() {  
    ValueNotifier<List<Plan>> planNotifier = PlanProvider.of(context);  
    List<Plan> plans = planNotifier.value;  
  
    if (plans.isEmpty) {  
        return Column(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: <Widget>[  
                const Icon(Icons.note, size: 100, color: Colors.grey),  
                Text('Anda belum memiliki rencana apapun.',  
                    style: Theme.of(context).textTheme.headlineSmall)  
            ]);  
    }  
    return ListView.builder(  
        itemCount: plans.length,  
        itemBuilder: (context, index) {  
            final plan = plans[index];  
            return ListTile(  
                title: Text(plan.name),  
                subtitle: Text(plan.completenessMessage),  
                onTap: () {  
                    Navigator.of(context).push(MaterialPageRoute(  
                        builder: (_) => PlanScreen(  
                            plan: plan,  
                        )  
                    )  
                )  
            );  
        }  
    );  
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

```
    });  
  }  
);  
}  
);  
}  
}  
}
```

- 40
1. Selesaikan langkah-langkah praktikum tersebut, lalu dokumentasikan berupa GIF hasil akhir praktikum beserta penjelasannya di file README.md! Jika Anda menemukan ada yang error atau tidak berjalan dengan baik, silakan diperbaiki sesuai dengan tujuan aplikasi tersebut dibuat.
 2. Berdasarkan Praktikum 3 yang telah Anda lakukan, jelaskan maksud dari gambar diagram berikut ini!



Gambaran ini tampak seperti sebuah diagram atau bagan yang mendeskripsikan relasi antar beragam komponen atau unsur dalam suatu aplikasi atau sistem perangkat lunak.

- Komponen utama yang terlihat antara lain:
- MaterialApp: Kemungkinan besar merupakan aplikasi atau kerangka kerja inti.



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile

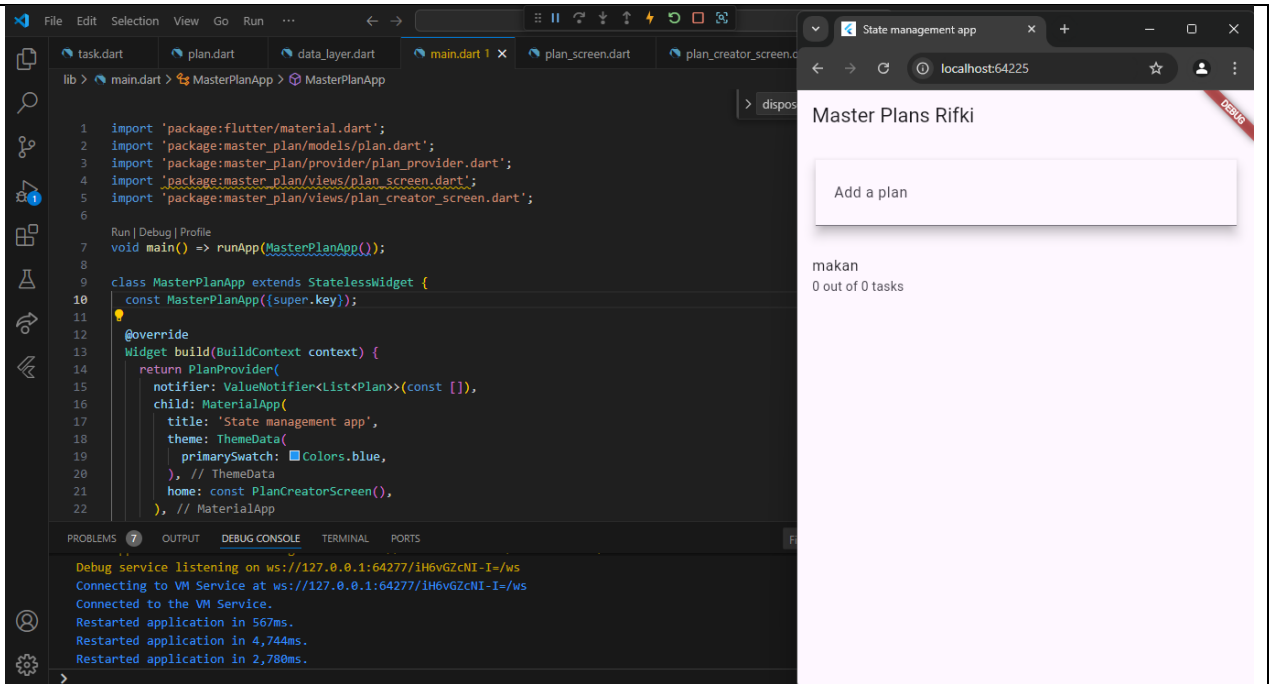
- PlanProvider: Mungkin menjadi komponen yang mengatur atau menyediakan fungsionalitas terkait rencana.
- PlanCreatorScreen: Kemungkinan besar merupakan layar atau antarmuka untuk membuat atau mengelola rencana.
- Column: Komponen tata letak atau antarmuka pengguna yang menyusun elemen secara vertikal.
- TextField dan Expanded: Unsur antarmuka pengguna untuk masukan teks dan perluasan, masing-masing.
- ListView: Komponen yang menyajikan daftar item yang dapat digulir.
- Navigator Push: Sebuah aksi atau transisi navigasi.
- PlanScreen: Berpotensi menjadi layar atau antarmuka untuk menampilkan atau berinteraksi dengan rencana.
- Scaffold: Komponen tata letak dasar.
- SafeArea: Komponen yang memastikan konten ditampilkan dalam area layar yang aman.
- Text: Elemen antarmuka pengguna untuk menampilkan teks.

Diagram ini tampaknya mengilustrasikan relasi dan interaksi antara berbagai komponen tersebut, kemungkinan besar dalam suatu aplikasi atau sistem perangkat lunak yang lebih besar.

3. Lakukan capture hasil dari Langkah 14 berupa GIF, kemudian jelaskan apa yang telah Anda buat!



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Dasar State Management
Mata Kuliah Pemograman Mobile



4. Kumpulkan laporan praktikum Anda berupa link commit atau repository GitHub ke spreadsheet yang telah disediakan!