

**WORKSHOP MOBILE APPLICATION  
PRAKTIKUM STATE MANAGEMENT  
(TIFNJK140707 )**

**SEMESTER IV**



**Fitri Meydayani  
(E41231686)**

**PROGRAM STUDI TEKNIK INFORMATIKA KAMPUS 3 NGANJUK  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI JEMBER  
TAHUN 2025**

## 1. Kode & Penjelasan counter\_provider.dart

```
1 import 'package:flutter/material.dart';
2
3 // CounterProvider menerapkan mixin ChangeNotifier untuk memberitahu listener ketika data berubah
4 class CounterProvider with ChangeNotifier {
5   // Variabel private untuk menyimpan nilai counter
6   int _count = 0;
7
8   // Getter untuk mengakses nilai counter dari luar class
9   int get count => _count;
10
11   // Fungsi untuk menambah nilai counter
12   void increment() {
13     _count++; // Menambah nilai counter
14     notifyListeners(); // Memberitahu semua widget yang listen provider ini bahwa data telah berubah
15   }
16
17   // Fungsi untuk mengurangi nilai counter
18   void decrement() {
19     if (_count > 0) { // Memeriksa nilai tidak kurang dari 0
20       _count--; // Mengurangi nilai counter
21       notifyListeners(); // Memberitahu semua widget yang listen provider ini bahwa data telah berubah
22     }
23   }
24 }
```

## 2. Kode & Penjelasan main.dart

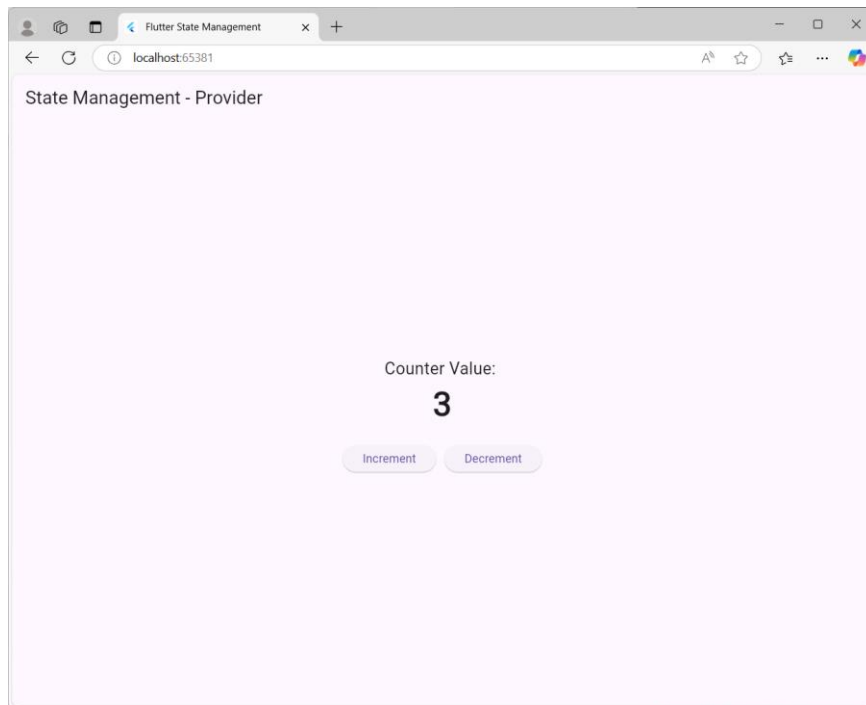
```
1 import 'package:flutter/material.dart';
2 import 'package:provider/provider.dart';
3 import 'providers/counter_provider.dart';
4 import 'counter_screen.dart';
5
6 // Fungsi utama yang menjadi entry point aplikasi
7 void main() {
8   runApp(
9     // MultiProvider digunakan untuk menyediakan beberapa provider dalam aplikasi
10     MultiProvider(
11       providers: [
12         // Mendaftarkan CounterProvider sebagai ChangeNotifier di aplikasi
13         // create: (_) membuat instance baru dari CounterProvider
14         ChangeNotifierProvider(create: (_) => CounterProvider()),
15       ],
16       // MyApp menjadi child dari MultiProvider sehingga dapat mengakses semua provider
17       child: const MyApp(),
18     ),
19   );
20 }
21
22 // Widget StatelessWidget sebagai root aplikasi
23 class MyApp extends StatelessWidget {
24   const MyApp({super.key});
25
26   @override
27   Widget build(BuildContext context) {
28     // MaterialApp sebagai wrapper utama yang mengatur tema dan navigasi aplikasi
29     return MaterialApp(
30       debugShowCheckedModeBanner: false, // Menyembunyikan banner debug di pojok kanan atas
31       title: 'Flutter State Management', // Judul aplikasi
32       home: const CounterScreen(), // Menentukan CounterScreen sebagai layar utama (home screen)
33     );
34   }
35 }
```

### 3. Kode & Penjelasan counter\_screen.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:provider/provider.dart';
3 import 'providers/counter_provider.dart';
4
5 // Widget StatelessWidget untuk menampilkan UI counter
6 class CounterScreen extends StatelessWidget {
7   const CounterScreen({super.key});
8
9   @override
10  Widget build(BuildContext context) {
11    // Mengambil instance CounterProvider dari context
12    // Ini akan otomatis rebuild widget ketika nilai di provider berubah
13    final counterProvider = Provider.of<CounterProvider>(context);
14
15    return Scaffold(
16      appBar: AppBar(
17        title:
18          const Text('State Management - Provider'), // AppBar dengan judul
19      body: Center(
20        child: Column(
21          mainAxisAlignment: MainAxisAlignment
22            .center, // Meletakkan children di tengah vertikal
23          children: [
24            // Teks statis untuk label counter
25            const Text(
26              'Counter Value: ',
27              style: TextStyle(fontSize: 20),
28            ),
29            // Teks yang menampilkan nilai counter dari provider
30            Text(
31              '${counterProvider.count}',
32              style: const TextStyle(fontSize: 40, fontWeight: FontWeight.bold),
33            ),
34            const SizedBox(height: 20), // Spacer vertikal
35            Row(
36              mainAxisAlignment: MainAxisAlignment
37                .center, // Meletakkan tombol di tengah horizontal
38              children: [
39                // Tombol untuk increment
40                // MASALAH: Seharusnya memanggil counterProvider.increment, bukan decrement
41                ElevatedButton(
42                  onPressed: counterProvider
43                    .decrement, // Fungsi yang salah - seharusnya increment
44                  child: const Text('Increment'),
45                ),
46                const SizedBox(width: 10), // Spacer horizontal
47                // Tombol untuk decrement
48                ElevatedButton(
49                  onPressed: counterProvider
50                    .decrement, // Fungsi yang benar untuk tombol decrement
51                  child: const Text('Decrement'),
52                ),
53              ],
54            ),
55          ],
56        ),
57      ),
58    );
59  }
60 }
61
```

#### 4. Tampilan

- Jika button increment ditekan nilai akan bertambah seperti pada gambar:



- Jika button decrement ditekan nilai akan berkurang seperti pada gambar:

