



SALFORD & CO.

PROGRAMMING

Visual Programming pada sesi ke - 4, dengan project yang dibuat yaitu flutter provider

Next Page

https://github.com/KingPublic/Flutter_provider_Andreyhs



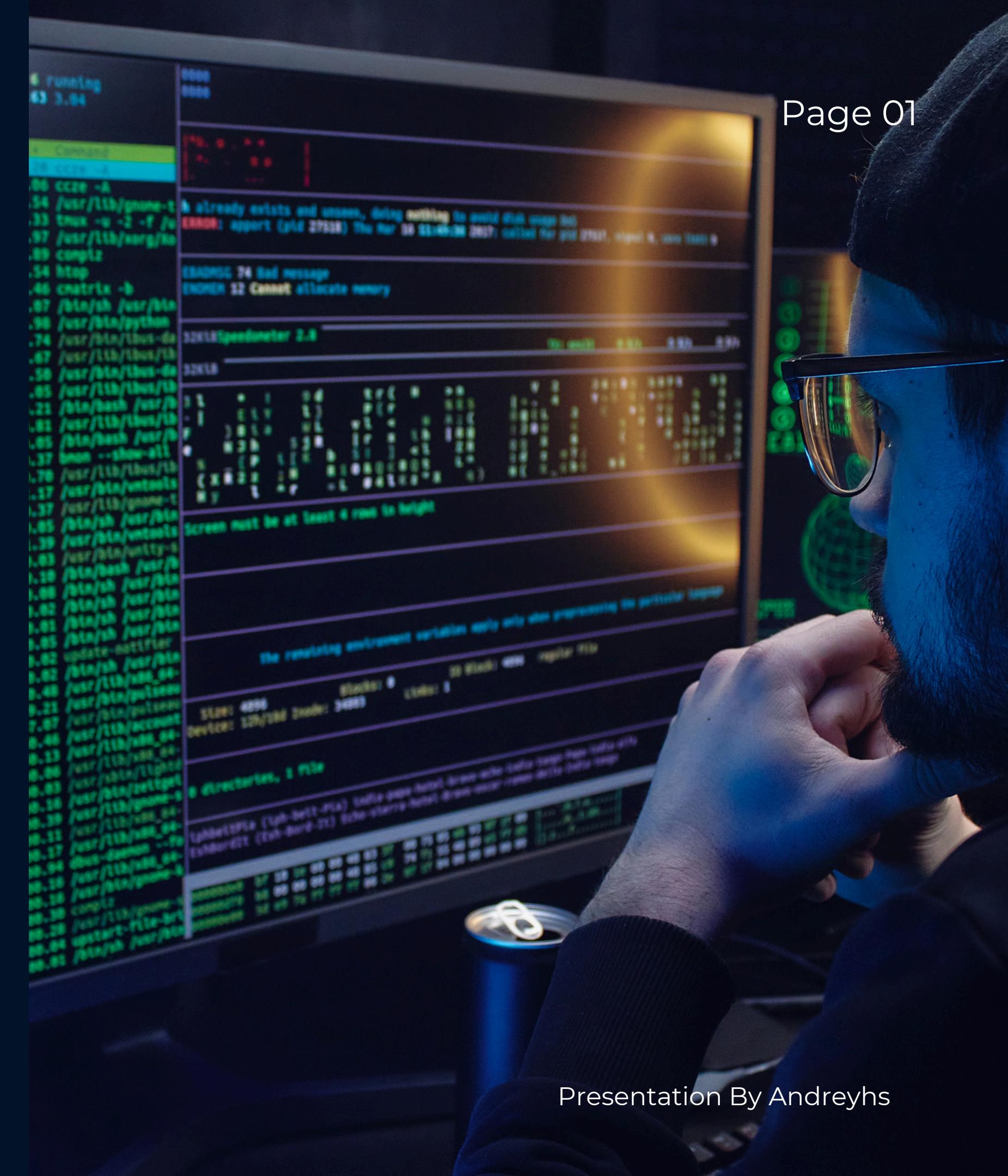
Presentation By Andreyhs



TUJUAN PROJEK

- Mengimplementasikan teknik manajemen state lanjutan.
- Memperkenalkan penggunaan StatefulWidget dan Provider untuk manajemen state.
- Membuat aplikasi counter yang dapat menangani state lokal dan global dengan fitur tambahan.

Next Page





PEMBAGIAN

- *Manajemen State Lokal*
- *Manajemen State Global*
- *UI dan Interaktivitas Lanjutan*



Next Page



PART 1 CODING

Fitur:

- Counter yang dapat diubah dengan tombol Increment dan Decrement.
- Nilai counter tidak boleh kurang dari nol.

Tujuan:

- Menggunakan StatefulWidget untuk mengelola state lokal aplikasi.

Next Page

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

class LocalCounterWidget extends StatefulWidget {
  @override
  _LocalCounterWidgetState createState() => _LocalCounterWidgetState();
}

class _LocalCounterWidgetState extends State<LocalCounterWidget> {
  int _counter = 0;

  void _increment() {
    setState(() {
      _counter++;
    });
  }

  void _decrement() {
    setState(() {
      if (_counter > 0) _counter--;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text('Local Counter: $_counter', style: TextStyle(fontSize: 24)),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            ElevatedButton(
              onPressed: _decrement,
              child: Text('-'),
            ),
            ElevatedButton(
              onPressed: _increment,
              child: Text('+'),
            ),
          ],
        ),
      ],
    );
  }
}
```



PART 2

Fitur:

- Menggunakan Provider untuk mengelola state global aplikasi.
- Setiap counter dapat ditambahkan, dihapus, atau dimodifikasi secara global.

Tujuan:

- Mengelola daftar counter secara global.
- Mengatur nilai counter, dan label menggunakan GlobalState.

Next Page

```
class GlobalState extends ChangeNotifier {  
  final List<Counter> _counters = [];  
  
  List<Counter> get counters => List.unmodifiable(_counters);  
  
  void addCounter() {  
    _counters.add(Counter());  
    notifyListeners();  
  }  
  
  void removeCounter(int index) {  
    _counters.removeAt(index);  
    notifyListeners();  
  }  
  
  void increment(int index) {  
    _counters[index].value++;  
    notifyListeners();  
  }  
  
  void decrement(int index) {  
    if (_counters[index].value > 0) {  
      _counters[index].value--;  
    }  
  }  
}  
  
class Counter extends ChangeNotifier {  
  int value = 0;  
  String label = 'Counter';  
  Color color = Colors.blue;  
  
  void increment() {  
    value++;  
    notifyListeners();  
  }  
  
  void decrement() {  
    if (value > 0) {  
      value--;  
    }  
    notifyListeners();  
  }  
}
```



PART 3

Fitur:

- Animasi pada tombol Increment dan Decrement.
- Drag-and-drop untuk mengatur ulang urutan counter.
- Setiap counter memiliki warna dan label yang dapat disesuaikan.

Tujuan:

- Meningkatkan interaktivitas dan UI dengan fitur dinamis seperti animasi dan drag-and-drop.

```
class AdvancedCounterApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Advanced Counter App'),
      ), // AppBar
      body: Consumer<GlobalState>(
        builder: (context, globalState, child) {
          return ListView.builder(
            itemCount: globalState.counters.length,
            itemBuilder: (context, index) {
              final counter = globalState.counters[index];
              return ListTile(
                leading: CircleAvatar(
                  backgroundColor: counter.color,
                ), // CircleAvatar
                title: Text('${counter.label}: ${counter.value}'),
                trailing: Row(
                  mainAxisAlignment: MainAxisAlignment.end,
                  children: [
                    IconButton(
                      icon: const Icon(Icons.add),
                      onPressed: () => globalState.increment(index),
                    ), // IconButton
                    IconButton(
                      icon: const Icon(Icons.remove),
                      onPressed: () => globalState.decrement(index),
                    ), // IconButton
                  ],
                ),
              );
            },
          );
        },
      ),
    );
}
```

Next Page

THANK YOU

[HTTPS://GITHUB.COM/KINGPUBLI/C-FLUTTER_PROVIDER_ANDREYHS](https://github.com/KINGPUBLI/C-FLUTTER_PROVIDER_ANDREYHS)

Presentation By Andreyhs