

Nama: Ni'mas Subang Abh.  
NIM: 4337 8592 0123 0083  
Prodi: Informatika

## Mobile Programming

### 1. Perbedaan StatefulWidget dan StatelessWidget.

- a. StatefulWidget adalah widget yang memiliki state atau data yang berubah cocok untuk UI Statis.

Contoh:

```
dart (class MyStatelessWidget extend StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Text('Hello, StatelessWidget!');  
  }  
})
```

- b. StatefulWidget adalah widget yang dapat berubah karena adanya state. Cocok untuk UI dinamis.

Contoh:

```
dart (class MyStatefulWidget extend StatefulWidget {  
  @override  
  - MyStatefulWidgetState createState() => - MyStatefulWidgetState();  
})
```

```
class - MyStatefulWidgetState extends State<MyStatefulWidget>  
{
```

```
  String text = "Hello!";
```

```
  void updateText() { set state (c) {  
    text = "Hello, StatefulWidget!";  
  }  
};
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return Column(children: [Text(text),
```

```
    ElevatedButton( onPressed: updateText, child: Text("update")  
    ], );  
  }  
};
```

### 2. Apa itu Hot Reload dalam flutter, dan apa itu kelebihannya?

Jawab:

Hot Reload adalah fitur flutter yang memungkinkan pengembang melihat perubahan kode langsung tanpa kehilangan state aplikasi yang sedang berjalan.

Kelebihan:

- Waktu pengembangan lebih cepat.

- Tidak perlu memulai ulang aplikasi

- Cocok untuk eksperimen dan debugging UI.

- Lebih Efisien



### 3. Halaman flutter dengan AppBar, TextField, dan ElevatedButton.

Halaman ini menunjukkan interaksi sederhana, misalnya, pengguna memasukkan textfield, kemudian menekan tombol untuk memproses input tersebut.

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

```
Void main() {  
  run App (MyApp());
```

```
Class MyApp extends StatelessWidget {  
  @override
```

```
Widget build (BuildContext context) {  
  return MaterialApp (home: MyHomePage (),); } }
```

```
Class MyHomePage extends StatelessWidget {  
  final TextEditingController ();
```

```
@override
```

```
Widget build (BuildContext context) {
```

```
< Scaffold ( appBar: AppBar (title: Text ('Example Page'),
```

```
body: Padding (padding: const EdgeInsets.all (16.0),  
  child: Column (children: [TextField (controller:
```

```
- controller,  
  decoration: InputDecoration (labelText: 'Enter  
  Something'), ),
```

```
SizeBox (height: 16),
```

```
ElevatedButton (onPressed: () { print ('Text: $ {  
  - controller.text }'); },
```

```
child: Text ('Submit'), ), ], ), ); } }
```

### 4. Cara Kerja Navigasi Antar Halaman di flutter. Navigator bekerja seperti stack;

a. Setiap kali pengguna berpindah ke halaman baru, halaman tersebut ditambahkan ke tumpukan menggunakan `Navigator.push()`.

b. Untuk kembali ke halaman sebelumnya, halaman diatas dihapus dari tumpukan menggunakan `Navigator.pop()`.

kelebihan:

a. Fleksibel untuk mendukung navigasi sederhana atau kompleks.

b. Mendukung animasi transisi bawaan.

c. Dapat digunakan untuk implementasi navigasi maju dan mundur dengan kontrol penuh.



Nama: Ni'mas Subang Aih.  
Nim: 4377 8552 023 0083  
Prodi: Informatika.

Lanjutan No. 4. kode

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

```
void main() {
```

```
  runApp(MyApp());
```

```
class MyApp extends StatelessWidget {
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return MaterialApp(home: pageA());
```

```
  };
```

```
class pageA extends StatelessWidget {
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return Scaffold(appBar: AppBar(title: Text('page A')),
```

```
      body: Center(child: ElevatedButton(
```

```
        onPressed: () {
```

```
          Navigator.push(context, MaterialPageRoute(
```

```
            builder: context => pageB(),
```

```
          ),
```

```
        ),
```

```
      ),
```

```
    );
```

```
  };
```

```
class pageB extends StatelessWidget {
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return Scaffold(appBar: AppBar(title: Text('page B')),
```

```
      body: Center(child: ElevatedButton(
```

```
        onPressed: () {
```

```
          Navigator.pop(context),
```

```
        ),
```

```
      ),
```

```
    );
```

```
  };
```

```
};
```



5. Peran Pubspec. yam dalam proyek flutter  
Pubspec. yam adalah file konfigurasi utama untuk proyek flutter.  
fungsi :

- Mendeklarasikan dependensi (library).
- Menentukan informasi proyek (nama, versi).
- Menambahkan asset seperti gambar, font, atau file lain.

Elemen penting :

- name : Nama aplikasi
- Version : Versi aplikasi
- Dependencies : Daftar library yang dibutuhkan.
- dev\_dependencies : Library untuk pengembangan.
- assets : File seperti gambar atau json yang digunakan aplikasi.