

Nama : Dias Sugi Rahayu

Kelas : 06TPLE017

Nim : 181011401635

## UAS MOBILE PROGRAMMING

### SOAL

1. Jelaskan apa yang dimaksud dengan Mobile Programming? Point 5
2. Jelaskan apa yang dimaksud dengan User Interface ( UI)? Point 5
3. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya! Point 5
4. Jelaskan perbedaan Native dan Hybrid pada mobile programming? Point 5
6. Jelaskan apa fungsi github! Point 5
7. Apa output dari script berikut ! Point 10:



```
ListView.builder(  
  itemCount: 10,  
  itemBuilder: (context, i){  
    return Text("$i");  
  },  
);
```

8. Apa output dari script berikut ! Point 10:

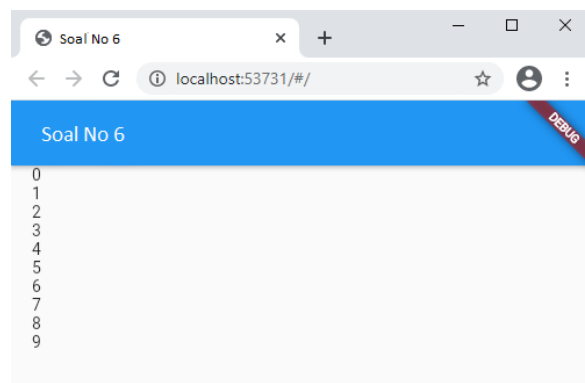


```
int timesTwo(int x) {  
  return x * 2;  
}  
  
int timesFour(int x) => timesTwo(timesTwo(x));  
  
int runTwice(int x, int Function(int) f) {  
  for (var i = 0; i < 2; i++) {  
    x = f(x);  
  }  
  return x;  
}  
  
void main() {  
  print("4 times two is ${timesTwo(4)}");  
  print("4 times four is ${timesFour(4)}");  
  print("2 x 2 x 2 is ${runTwice(2, timesTwo)}");  
}
```

9. Tuliskan sintak cara parsing JSON pada flutter ! Poin 55

JAWABAN :

1. Pemrograman mobile merupakan gabungan antara kata ” **pemrograman** ” dan ” **mobile** “. Istilah lainnya yaitu **mobile programming**. Pemrograman ini secara singkat memiliki makna proses menulis kode-kode program untuk membuat aplikasi yang ditujukan untuk perangkat bergerak (*Mobile*).
2. User Interface adalah tampilan visual sebuah produk yang menjembatani sistem dengan pengguna (*user*). Tampilan UI dapat berupa bentuk, warna, dan tulisan yang didesain semenarik mungkin. Secara sederhana, UI adalah bagaimana tampilan sebuah produk dilihat oleh pengguna.
3. Sebuah API adalah sekumpulan definisi dan protokol untuk membangun dan mengintegrasikan perangkat lunak aplikasi. API adalah singkatan dari antarmuka pemrograman aplikasi (Application Programming Interface). API memungkinkan produk atau layanan Anda berkomunikasi dengan produk dan layanan lain tanpa harus tahu bagaimana penerapannya.
4. - [native](#) adalah aplikasi mobile yang dikembangkan secara khusus untuk satu sistem operasi. Contohnya adalah [pengembangan android](#) hanya untuk android saja, atau Objective-C/Swift hanya untuk ios saja. Yang mana keduanya memiliki IDE (Integrated Development Environment) masing-masing. Untuk membuat aplikasi android, IDE yang digunakan adalah Android Studio. Sedangkan untuk membuat aplikasi ios, IDE yang digunakan adalah XCode. Untuk menggunakan produk aplikasi native, kita tinggal download di playstore untuk android dan appstore untuk ios.
  - [Hybrid](#) adalah aplikasi yang pengembangannya menggabungkan antara aplikasi native dan aplikasi web mobile. Yang awalnya aplikasi web kemudian diubah dalam sebuah tool sehingga menjadi kode native. Beberapa tool untuk mengembangkan aplikasi hybrid antara lain Phonegap, Xamarin, Ionic dan lainnya. Aplikasi hybrid ini dikembangkan menggunakan HTML5 dan JavaScript. Untuk dapat menggunakannya kita dapat download di masing-masing market platform. Inilah perbedaan utama antara aplikasi web dan aplikasi hybrid. Aplikasi web mobile tidak tersedia di market, sebaliknya hybrid sudah tersedia.
5. aplikasi dengan basis website ini berfungsi untuk membantu penyimpanan *repository*. Namun tak hanya itu, dengan aplikasi ini Anda juga dapat melakukan kolaborasi dalam mengerjakan *project* tertentu agar dapat terus melakukan update secara rutin, termasuk melacak dan menyimpan perubahan-perubahan yang terjadi dalam *project* Anda tersebut.
- 6.



Script :

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp( title: 'Soal No.6',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Soal No.6'), centerTitle: false,
        ),
        body: Container(
          child: Center(
            child: Column(
              children: <Widget>[ Container(
                height: 400,
                padding: EdgeInsets.symmetric(horizontal: 18), child: ListView.builder(
                  itemCount: 10,
                  itemBuilder: (context, i) {
                    return Text("$i");
                  },
                ),
              ),
            ),
            Text("NAMA : DIAS SUGI RAHAYU"), Text("NIM : 181011401635"), Text("KELAS : 05TPLE017"),
          ],
        ),
      ),
    /*child: ListView.builder(
      itemCount: 10,
      itemBuilder: (context, i) {
        return Text("$i");
      },
    ),*/
  ),
)
```

```

    }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

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

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
      // so that the display can reflect the updated values. If we changed
      // _counter without calling setState(), then the build method would not be
      // called again, and so nothing would appear to happen.
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    // This method is rerun every time setState is called, for instance as done
    // by the _incrementCounter method above.
    //
    // The Flutter framework has been optimized to make rerunning build methods
    // fast, so that you can just rebuild anything that needs updating rather
    // than having to individually change instances of widgets. return Scaffold(
    appBar: AppBar(

```

```

        // Here we take the value from the MyHomePage object that was created by
        // the App.build method, and use it to set our appbar title. title:
        Text(widget.title),
    ),
    body: Center(
        // Center is a layout widget. It takes a single child and positions it
        // in the middle of the parent. child:
        Column(
            // Column is also a layout widget.
            // It takes a list of children and
            // arranges them vertically. By default, it sizes itself to fit its
            // children horizontally, and tries to be as tall as its parent.
            //
            // Invoke "debug painting" (press "p" in the console, choose the
            // "Toggle Debug Paint" action from the Flutter Inspector in Android
            // Studio, or the "Toggle Debug Paint" command in Visual Studio Code)
            // to see the wireframe for each widget.
            //
            // Column has various properties to control how it sizes itself and
            // how it positions its children. Here we use mainAxisAlignment to
            // center the children vertically; the main axis here is the vertical
            // axis because Columns are vertical (the cross axis would be
            // horizontal).
            mainAxisAlignment: MainAxisAlignment.center, children:
            <Widget>[
                Text(
                    'You have pushed the button this many times:',
                ),
                Text(
                    '$_counter',
                    style: Theme.of(context).textTheme.headline4,
                ),
            ],
        ),
    ),
    floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter, tooltip:
        'Increment',
        child: Icon(Icons.add),
    ), // This trailing comma makes auto-formatting nicer for build methods.
);

```

```
}  
}
```

7.

```
Performing hot restart...  
Waiting for connection from debug service on Chrome...  
Restarted application in 188ms.  
4 times two is 8  
4 times four is 16  
2 x 2 x 2 is 8
```

8. menampilkan sebuah urutan kebawah nama, nim, kelas

```
import  
'dart:convert';  
  
List decodedList = jsonDecode('["DIAS SUGI RAHAYU", "181011401635",  
"06TPLE017"]');
```



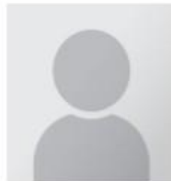
**UNIVERSITAS PAMULANG**  
**KARTU UJIAN AKHIR SEMESTER GENAP 2020/2021**  
**NOMOR UJIAN : 418729624906**

FAK/PROG : TEKNIK / TEKNIK INFORMATIKA  
NAMA : DIAS SUGI RAHAYU  
NIM : 181011401635  
SHIFT : REGULER C

NO	HARI / TANGGAL	WAKTU	RUANG	KELAS	MATA KULIAH	PARAF
1	-			06TPLE017	SISTEM INFORMASI MANAJEMEN	
2	-			06TPLE017	KERJA PRAKTEK	
3	-			06TPLE017	MOBILE PROGRAMMING	

**Peraturan dan Tata Tertib Peserta Ujian**

1. Peserta ujian harus berpakaian rapi, sopan dan memakai jaket Almamater
2. Peserta ujian sudah berada di ruangan sepuluh menit sebelum ujian dimulai
3. Peserta ujian yang terlambat diperkenankan mengikuti ujian setelah mendapat ijin, tanpa perpanjangan waktu
4. Peserta ujian hanya diperkenankan membawa alat-alat yang ditentukan oleh panitia ujian
5. Peserta ujian dilarang membantu teman, mencontoh dari teman dan tindakan-tindakan lainnya yang mengganggu peserta ujian lain
6. Peserta ujian yang melanggar tata tertib ujian dikenakan sanksi akademik



Pamulang, 07 April 2021  
Ketua Panitia Ujian

**Dr. E. NURZAMAN AM, M.M, M. Si**  
**NIDK. 8811520016**