

## UAS MOBILE PROGRAMMING

NAMA : Redo Ramadhan Caniago

NIM : 181011401701

KELAS : 06TPLE017

DOSEN : Ade Putra Prima Suhendri, S.Kom, M.Kom

### **1. Jelaskan apa yang dimaksud dengan Mobile Programming?**

Jawab : Mobile programming adalah proses pembuatan aplikasi untuk perangkat mobile baik aplikasi yang bersifat offline maupun online. mendukung sepenuhnya pengeditan file proyek yang menggunakan bahasa C/C++ sehingga teman-teman dapat dengan cepat membuat komponen-komponen JNI dalam aplikasi.

### **2. Jelaskan apa yang dimaksud dengan User Interface (UI)?**

Jawab : User interface adalah bagian visual dari aplikasi software atau device hardware yang memastikan bagaimana seorang user berinteraksi dengan aplikasi atau website tersebut serta bagaimana informasi ditampilkan di layarnya. User interface sendiri menggabungkan konsep desain visual, desain interaksi, dan infrastruktur informasi. Tujuan dari user interface adalah untuk meningkatkan usability dan tentunya user experience.

### **3. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya!**

Jawab : API adalah sebuah antarmuka yang digunakan untuk menghubungkan antara satu aplikasi dengan aplikasi yang lain. Peran dari API adalah untuk sebagai perantara yang menghubungkan aplikasi berbeda, baik dari platform yang sama maupun lintas platform. API juga digunakan sebagai alat untuk melakukan kegiatan komunikasi dengan developer menggunakan berbagai jenis bahasa pemrograman. Developer tidak perlu untuk menyediakan data sendiri, namun cukup dengan mengambil data dan informasi dari platform melalui API.

### **4. Jelaskan perbedaan Native dan Hybrid pada mobile programming?**

Jawab : Native adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk platform tertentu. Contoh populernya yakni penggunaan bahasa pemrograman Objective-C atau Swift untuk platform iOS (Apple). Adapun platform Android yang menggunakan bahasa pemrograman Java. Membangun aplikasi native harus menyediakan pengalaman produk yang optimal pada perangkat mobile. Meskipun begitu, budget yang tinggi dibutuhkan untuk membangun aplikasi cross platform yang mampu mempertahankan aplikasi native tetap update.

Hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android. Aplikasi hybrid biasanya menggunakan browser untuk mengizinkan aplikasi web mengakses berbagai fitur di device mobile seperti Push Notification, Contacts, atau Offline Data Storage. Beberapa tools untuk mengembangkan aplikasi hybrid antara lain Phonegap, RhoMobile dan lain-lain.

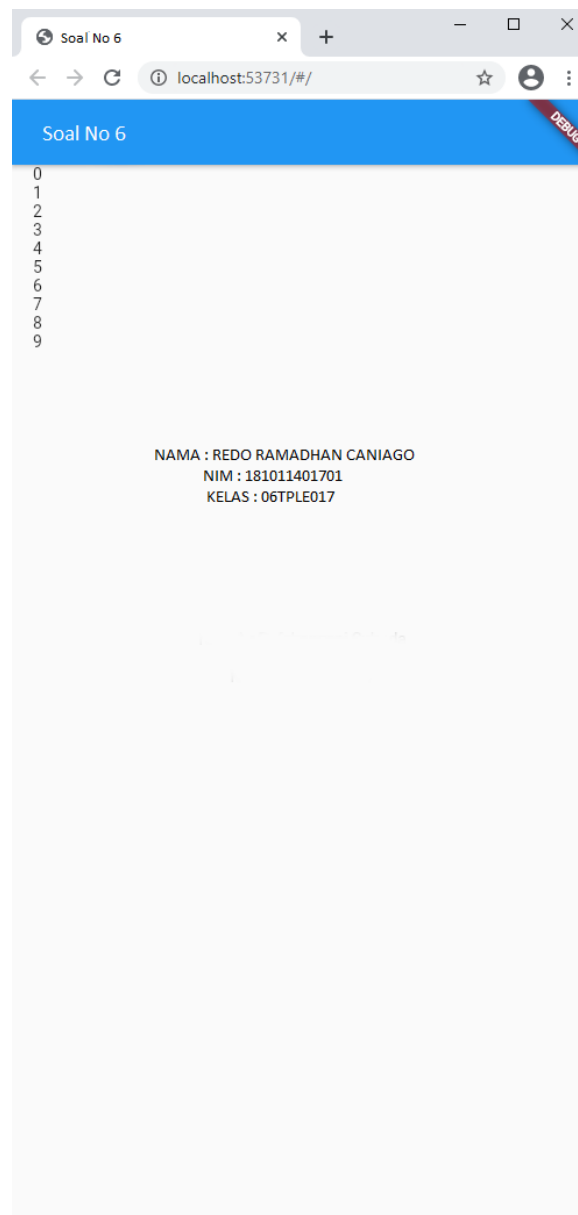
### **5. Jelaskan apa fungsi github!**

Jawab : Aplikasi dengan basis website ini berfungsi untuk membantu penyimpanan repository. 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. Aplikasi ini tidak hanya dapat digunakan oleh para programmer atau orang-orang yang sedang memiliki project saja. Aplikasi ini dapat digunakan secara luas oleh siapa saja, termasuk oleh Anda yang saat ini sedang mencari pekerjaan. Aplikasi ini memiliki laman profil yang akan memuat data pribadi Anda, seperti foto, email, bahkan followers yang Anda miliki. Informasi-informasi yang dimuat dalam profil Anda ini akan menarik pihak lain untuk mengajak Anda bekerjasama dalam sebuah project.

6. Apa output dari script berikut !

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

Jawab :



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("$i");
                    },
                  ),
                Text("NAMA : REDO RAMADHAN CANIAGO"),
                Text("NIM : 181011401701"),
                Text("KELAS : 05TPLE017"),
              ],
            ),
          ),
        ),
      /*child: ListView.builder(
        itemCount: 10,
        itemBuilder: (context, i) {
          return Text("$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. Apa output dari script berikut !

```
Jawab: 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)}");  
}
```

Jawab:

```
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. Tuliskan sintak cara parsing JSON pada flutter !

Jawab : akan menampilkan sebuah urutan dibawah nama, nim, kelas

```
import  
'dart:convert';  
  
List decodedList = jsonDecode('["REDO RAMADHAN CANIAGO", "181011401701",  
"06TPLE017"]');  
void main() {  
    print("NAMA = ${decodedList[0]}");  
    print("NIM = ${decodedList[1]}");  
    print("KELAS = ${decodedList[2]}");  
}
```