

NAMA: ANDRIAN FAKHRIZAL

NIM : 181011401677

KELAS: 06TPLE017

UAS

MOBILE PROGRAMMING

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

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
5. Jelaskan apa fungsi github! Point 5
6. Apa output dari script berikut ! Point 10:



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

7. 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)}");  
}
```

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

Jawaban

1. Mobile programming adalah proses pembuatan aplikasi untuk perangkat mobile baik aplikasi yang bersifat offline maupun online
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. API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. Jadi, API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform.

Manfaat API

1. Memudahkan Membangun Aplikasi yang Fungsional

Dengan menggunakan API, akan lebih mudah untuk membuat aplikasi yang fungsional dan kompleks. Tanpa perlu menambahkan data secara manual, aplikasi yang dikembangkan akan memiliki fitur dari aplikasi tujuan.

Sebagai contoh, pada aplikasi Gojek. Sebagai sebuah platform layanan transportasi, peran peta sangatlah penting. Namun, Gojek tidak perlu mengembangkan aplikasi peta sendiri. Dengan API, aplikasi tersebut cukup mengambil data dari Google Maps.

Penggunaan API ini cukup membantu membuat platform Gojek semakin besar. Alasannya, developer cukup mengembangkan layanan lain karena penggunaan peta sebagai elemen utama dipastikan berjalan dengan baik.

2. Pengembangan Aplikasi Menjadi Lebih Efisien

Dengan adanya API, Anda tidak perlu melakukan komunikasi langsung dengan aplikasi lain yang ingin dihubungkan. Cukup dengan komunikasi melalui API. Hal ini sangat membantu, terutama jika Anda ingin membangun aplikasi lintas platform dengan berbagai layanan sekaligus.

Sebagai contoh, Anda membangun website pemesanan tiket online untuk berbagai maskapai di dunia. Dengan bantuan API, Anda cukup melakukan integrasi untuk masing-masing layanan maskapai tersebut. Jadi, tidak perlu lagi melakukan komunikasi manual berupa update harga atau tersedianya tempat duduk.

Selain itu, Anda bisa dengan mudah menambahkan atau mengurangi integrasi layanan sesuai perkembangan bisnis Anda.

3. Meringankan Beban Server

Dengan menggunakan API, Anda tidak perlu menyimpan semua data yang dibutuhkan di server Anda sendiri. Cukup meminta API untuk mendapatkan data terbaru dari server aplikasi asal. Dengan kondisi ini, server Anda tidak akan

terbebani. Pada akhirnya, mengurangi resiko website tidak dapat diakses karena server down.

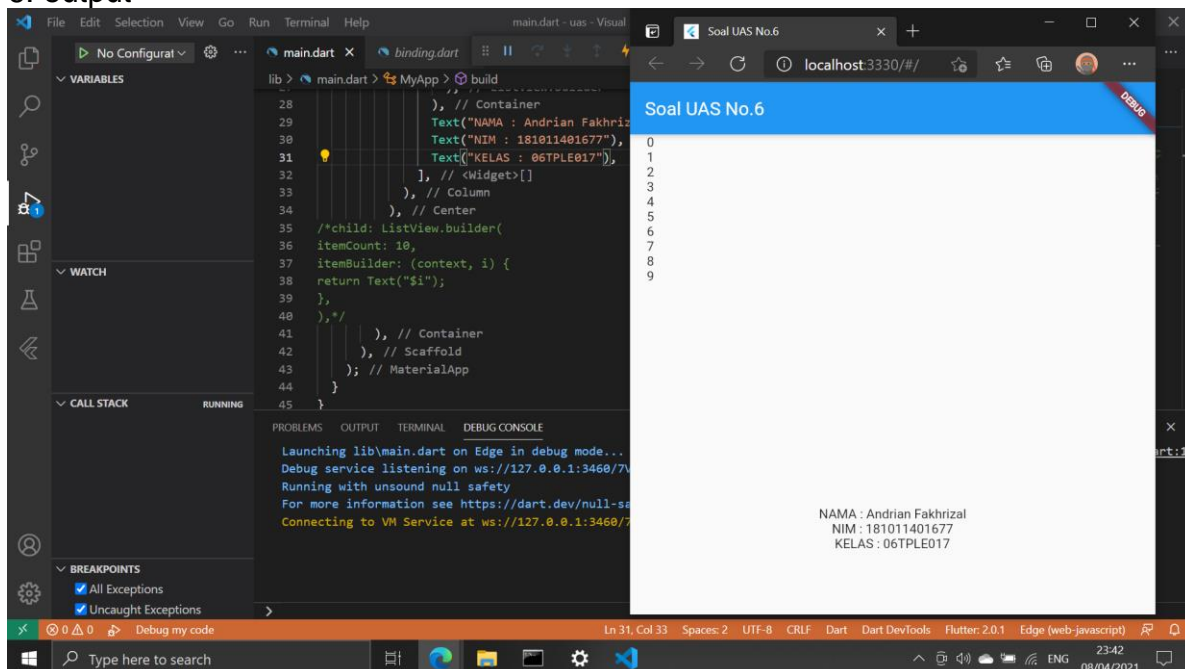
4.1. Aplikasi native adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk platform tertentu. Contoh populernya yakni penggunaan bahasa pemrograman Objective-C atau Swift untuk platform iOS (Apple). Adapun platform Android yang menggunakan bahasa pemrograman Java.

2. Aplikasi hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android. Aplikasi hybrid biasanya menggunakan browser untuk memungkinkan 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, Rubymotion dan lain-lain

5. Beberapa fungsi github adalah:

- Memungkinkan Anda untuk berkolaborasi dengan orang lain;
- Menyimpan dan mengawasi repository;
- Merencanakan, menyimpan dan melacak proses kerja dari proyek;
- Berkomunikasi dengan sesama programmer;
- Melacak bug dan manajemen tugas. hingga;
- Menampilkan profil dan update dari Anda ke khalayak banyak.

6. output



Script:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Soal UAS No.6',
      home: Scaffold(
        appBar: AppBar(
```

```

        title: Text('Soal UAS 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("$i");
                            },
                        ),
                    Text("NAMA : Andrian Fakhrizal"),
                    Text("NIM : 181011401677"),
                    Text("KELAS : 06TPLE017"),
                ],
            ),
        ),
    ),
    /*child: ListView.builder(
    itemCount: 10,
    itemBuilder: (context, i) {
    return Text("$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:',
            ),
          ],
        ),
      ),
    );
  }
}

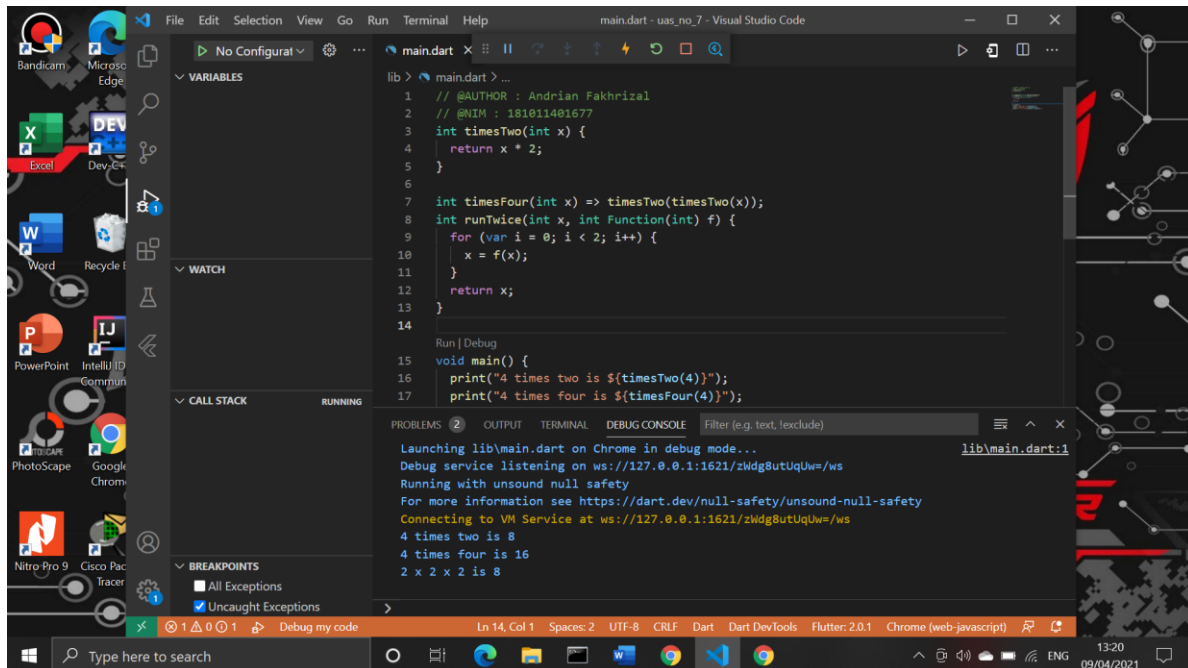
```

```

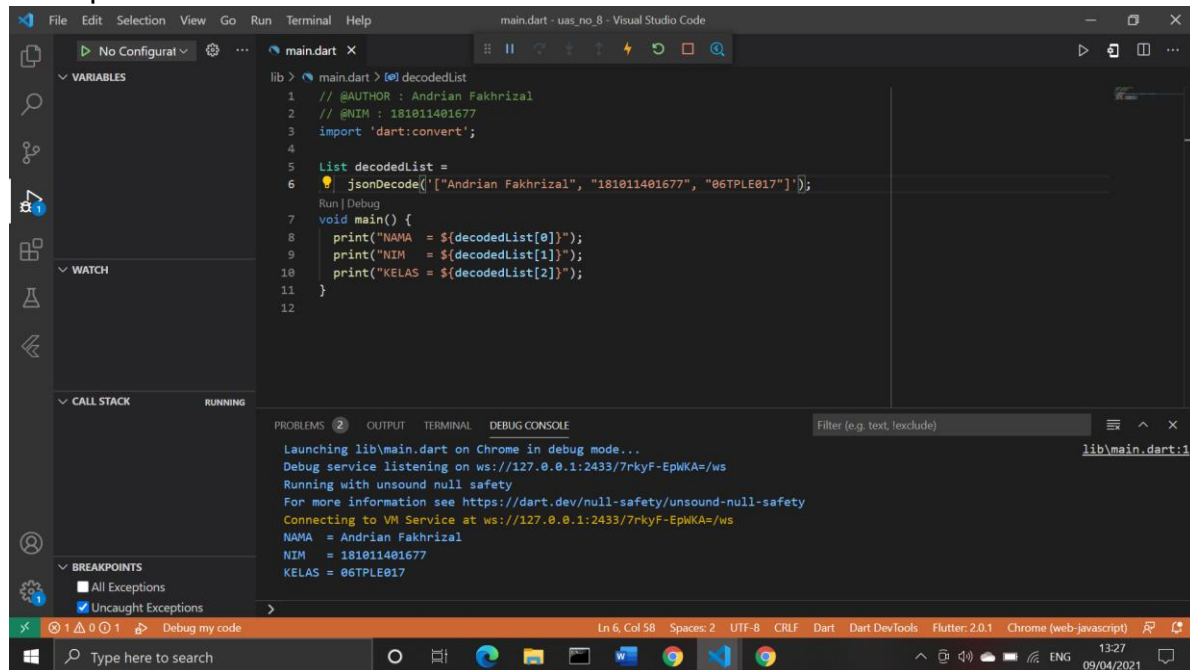
        Text(
          '$_counter',
          style: Theme.of(context).textTheme.headline4,
        ),
      ],
    ),
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: _incrementCounter,
    tooltip: 'Increment',
    child: Icon(Icons.add),
  ), // This trailing comma makes auto-
formatting nicer for build methods.
);
}
}

```

7.OUTPUT



8. output



The screenshot shows the Visual Studio Code editor with a Dart file named `main.dart` open. The file contains the following code:

```
lib > main.dart > decodedList
1 // @AUTHOR : Andrian Fakhrihal
2 // @NIM : 181011401677
3 import 'dart:convert';
4
5 List decodedList =
6   jsonDecode(['Andrian Fakhrihal', '181011401677', '06TPLE017']);
7
8 void main() {
9   print("NAMA = ${decodedList[0]}");
10  print("NIM = ${decodedList[1]}");
11  print("KELAS = ${decodedList[2]}");
12 }
```

The debug console at the bottom shows the output of the program:

```
Launching lib/main.dart on Chrome in debug mode...
Debug service listening on ws://127.0.0.1:2433/7rkyF-EpWKA=/ws
Running with unsound null safety
For more information see https://dart.dev/null-safety/unsound-null-safety
Connecting to VM Service at ws://127.0.0.1:2433/7rkyF-EpWKA=/ws
NAMA = Andrian Fakhrihal
NIM = 181011401677
KELAS = 06TPLE017
```

import

'dart:convert';

List decodedList = jsonDecode(['Andrian Fakhrihal', '181011401677', '06TPLE017']);

void main() {

print("NAMA = \${decodedList[0]}");

print("NIM = \${decodedList[1]}");

print("KELAS = \${decodedList[2]}");


}

Kartu Ujian.pdf

File | C:/Users/Andrian/Downloads/Kartu%20Ujian.pdf

1 of 1

Page view | Read aloud | Draw | Highlight | Erase





UNIVERSITAS PAMULANG
KARTU UJIAN AKHIR SEMESTER GENAP 2020/2021
NOMOR UJIAN : 473514209722

FAK/PROG : TEKNIK / TEKNIK INFORMATIKA
NAMA : ANDRIAN FAKHRIZAL
NIM : 181011401677
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, mencoretlah dari teman dan tindakan lainnya yang mengganggu peserta ujian lain
6. Peserta ujian yang melanggar tata tertib ujian dikenakan sanksi akademik



Pamulang, 08 April 2021
Ketua Panitia Ujian

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

Type here to search

13:29
09/04/2021