

Nama : Rizki Gustianto Sidi
NIM : 181011401641
Kelas : 06TPLE017

UTS MOBILE PROGRAMMING

Dosen : Ade Putra Prima Suhendri, S.Kom, M.Kom
Email : dosen02555@unpam.ac.id

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. Buatlah Design UI menggunakan Framework Flutter sesuai dengan Keterangan Aplikasi ! Screenshot Script dan Hasil UI nya di simulator atau device ! **Point 40**
6. Hubungkan UI dengan API menggunakan Framework Flutter sesuai dengan Keterangan Aplikasi ! Screenshot Script dan Hasil UI nya di simulator atau device ! **Point 40**

Keterangan Aplikasi :

Saya seorang manajer disebuah perusahaan keuangan yang membutuhkan sebuah aplikasi sederhana yang bisa digunakan untuk melihat KURS mata uang secara realtime, anda bisa menggunakan API dari <https://fixer.io/documentation> atau API lain yang anda ketahui.

JAWABAN

1.

Mobile Programming adalah pemrograman yang ditujukan untuk pembuatan aplikasi diperangkat mobile. Banyak sekali platform mobile yang dapat kita coba, diantaranya iOS, BB RIM, J2ME, QT Mobile, Symbian, dan Android. Salah satunya adalah Android Mobile. Android merupakan sistem operasi berbasis linux yang bahasa pemrograman aplikasinya dapat kita buat menggunakan java. 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.

Fungsi :

- Memudahkan membangun aplikasi yang fungsional
- Pengembangan aplikasi menjadi lebih efisien
- Meringankan Beban Server.

4.

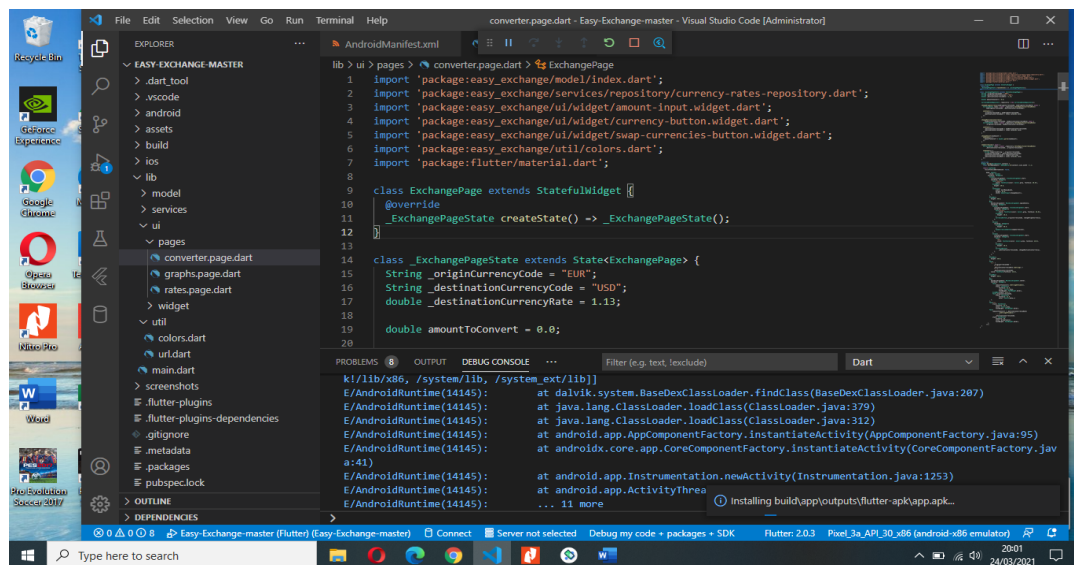
NATIVE

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.

HYBRID

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



```

import 'package:easy_exchange/model/index.dart';
import 'package:easy_exchange/services/repository/currency-rates-repository.dart';
import 'package:easy_exchange/ui/widget/amount-input/widget.dart';
import 'package:easy_exchange/ui/widget/currency-button/widget.dart';
import 'package:easy_exchange/ui/widget/swap-currencies-button/widget.dart';
import 'package:easy_exchange/util/colors.dart';
import 'package:flutter/material.dart';

class ExchangePage extends StatefulWidget {
  @override
  _ExchangePageState createState() => _ExchangePageState();
}

class _ExchangePageState extends State<ExchangePage> {
  String _originCurrencyCode = "EUR";
  String _destinationCurrencyCode = "USD";
  double _destinationCurrencyRate = 1.13;

  double amountToConvert = 0.0;

  CurrencyRatesRepository _repository = new CurrencyRatesRepository();

  changeOrigenCurrency(newOrigenCurrencyCode, newOrigenCurrencyRate) async {
    CurrencyRates rates = await _repository.fetchSpecificCurrencyRates(
      newOrigenCurrencyCode, _destinationCurrencyCode);

    setState(() {
      _originCurrencyCode = newOrigenCurrencyCode;
      _destinationCurrencyRate = rates.rates[0].rate;
    });
  }

  changeDestinationCurrency(
    newDestinationCurrencyCode, newDestinationCurrencyRate) async {
    CurrencyRates rates = await _repository.fetchSpecificCurrencyRates(
      _originCurrencyCode, newDestinationCurrencyCode);

    setState(() {
      _destinationCurrencyCode = newDestinationCurrencyCode;
      _destinationCurrencyRate = rates.rates[0].rate;
    });
  }

  changeAmount(newAmount) {
    setState(() {
      amountToConvert = double.parse(newAmount);
    });
  }

  swapCurrencies() async {
    CurrencyRates rates = await _repository.fetchSpecificCurrencyRates(
      _destinationCurrencyCode, _originCurrencyCode);

    setState(() {
      String temporaryString = _originCurrencyCode;
      _originCurrencyCode = _destinationCurrencyCode;
      _destinationCurrencyCode = temporaryString;
      _destinationCurrencyRate = rates.rates[0].rate;
    });
  }
}

```

```

    });
}

@override
Widget build(BuildContext context) {
  final halfMediaWidth = MediaQuery.of(context).size.width / 1.1;

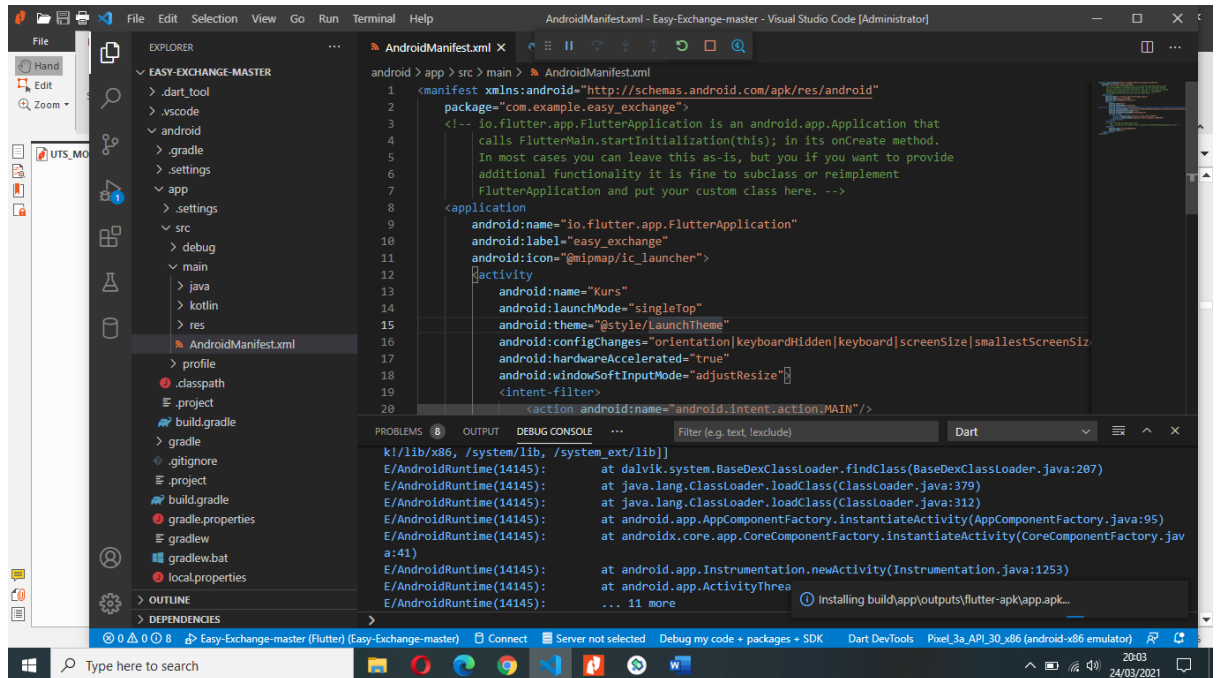
  return Scaffold(
    resizeToAvoidBottomInset: false,
    body: Center(
      child: Column(
        children: <Widget>[
          Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: <Widget>[
              Text("Amount",
                style: TextStyle(color: Colors.grey, fontSize: 20.0)),
              SizedBox(
                height: 10.0,
              ),
              Container(
                width: halfMediaWidth,
                height: 55.0,
                child: AmountInput(changeAmount)),
            ],
          ),
          SizedBox(
            height: 40.0,
          ),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: <Widget>[
              Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                  Text("From",
                    style: TextStyle(color: Colors.grey, fontSize: 20.0)),
                  SizedBox(
                    height: 10.0,
                  ),
                  CurrencyButton(_originCurrencyCode, changeOrigenCurrency),
                ],
              ),
              Column(
                children: <Widget>[
                  SizedBox(
                    height: 40.0,
                  ),
                  SwapCurrenciesButton(swapCurrencies),
                ],
              ),
              Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                  Text(
                    "To",
                    style: TextStyle(color: Colors.grey, fontSize: 20.0),
                  ),

```

```

        SizedBox(
          height: 10.0,
        ),
        CurrencyButton(
          _destinationCurrencyCode, changeDestinationCurrency),
      ],
    ),
    ],
  ),
  SizedBox(
    height: 30.0,
  ),
  Text(
    "1 " +
      _originCurrencyCode +
      " = " +
      _destinationCurrencyRate.toString() +
      " " +
      _destinationCurrencyCode,
    style: TextStyle(fontSize: 16.0),
  ),
  SizedBox(
    height: 40.0,
  ),
  Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Text(
        (amountToConvert).toStringAsFixed(2),
        style: TextStyle(
          fontSize: 21.0,
          color: Colors.grey,
          fontWeight: FontWeight.bold)),
      SizedBox(width: 10.0,),
      Text(_originCurrencyCode,
        style: TextStyle(
          fontSize: 21.0,
          color: Colors.grey,))
    ],
  ),
  Text("=",
    style: TextStyle(
      fontSize: 24.0,
      color: Colors.grey,
      fontWeight: FontWeight.bold)),
  Text(
    (amountToConvert * _destinationCurrencyRate)
      .toStringAsFixed(2) +
      " " +
      _destinationCurrencyCode,
    style: TextStyle(
      fontSize: 45,
      color: primaryColor,
      fontWeight: FontWeight.bold)),
  ],
  ));
}
}

```



```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.easy_exchange">
    <!-- io.flutter.app.FlutterApplication is an android.app.Application that
         calls FlutterMain.startInitialization(this); in its onCreate method.
         In most cases you can leave this as-is, but you if you want to provide
         additional functionality it is fine to subclass or reimplement
         FlutterApplication and put your custom class here. -->
    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="easy_exchange"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name="Kurs"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|l
            ayoutDirection|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    <!-- Don't delete the meta-data below.
         This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
    <meta-data
        android:name="flutterEmbedding"
        android:value="2" />
    </application>
</manifest>

```

Tampilan Aplikasi

The image displays two screenshots of the 'Easy Exchange' web application running in a browser on a Windows 10 desktop.

Top Screenshot: Rates Page

The browser address bar shows `localhost:56721/#/`. The page title is 'Rates'. It lists the following exchange rates:

Country	Flag	Rate
Philippine Peso	PHP	57.77900
Danish Krone	DKK	7.43600
Forint	HUF	366.51000
Czech Koruna	CZK	26.19800
Australian Dollar	AUD	1.54910
New Romanian Leu	RON	4.88930
Swedish Krona	SEK	10.18130

Bottom Screenshot: Converter Page

The browser address bar shows `localhost:56721/#/`. The page title is 'Converter'. It shows a conversion from HKD to CZK.

Amount: 10.00

From: HKD

To: CZK

1 HKD = 2.8386607433 CZK

10.00 HKD =

28.39 CZK



Converter

Amount

From



To

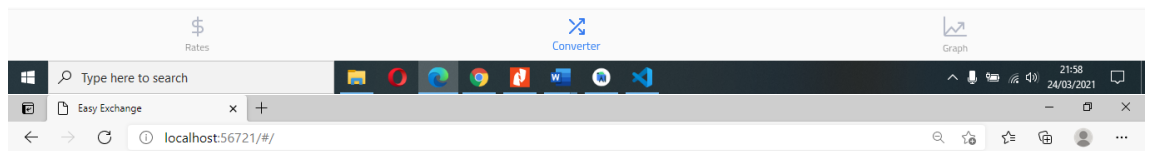


1 HKD = 1853.4597464514 IDR

10.00 HKD

=

18534.60 IDR



Graph



1D

15

1M

3M

1A

5A

EUR/USD



