Nama : Aji Abdul Majid Dosen : Ade Putra Prima Suhendri, S.Kom, M.Kom

Nim : 181011400184 Matkul : Mobile Programming

Kelas : 06TPLE010 Shift : Reguler C Sabtu

1. Jelaskan apa yang dimaksud dengan Mobile Programming?

* Mobile Programming adalah pemrograman yang digunakan untuk perangkat mobile

1. Jelaskan apa yang dimaksud dengan User Interface (UI)?

* User Interface merupakan tampilan visual dari sebuah produk yang berfungsi untuk menjembatani sistem dengan user atau pengguna. Dimana tampilan UI bisa berupa warna, bentuk serta tulisan yang didesain dengan semenarik mungkin. Namun secara sederhana, UI dapat diartikan sebagai bagaiaman tampilan suatu produk dilihat oleh pengguna atau user.

1. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya!

* API kependekan dari Application Programming Interface merupakan sebuah perangkat lunak yang dapat menerima panggilan atau permintaan dari perangkat lunak lainnya seperti aplikasi dan website yang memberikan pelayanan. Dengan kata lain API merupakan sebuah perangkat lunak yang mengijinkan atau menghubungkan dua aplikasi.
* Berikut merupakan beberapa fungsi utama dari penggunaannya.
  1. Membantu beban kerja dari server

Fungsi pertama dari API sendiri adalah untuk membantu tugas dari server. Dimana, dengan menggunakan sebuah antarmuka khusus ini, maka server tidak perlu mencari dan menyimpan semua data. Cukup dengan memanggil atau meminta API untuk mendapatkan data dari server asal.  Dengan kondisi tersebut, server yang anda gunakan tidak akan terbebani tugas terlalu berat.

### Mengembangkan aplikasi lebih cepat dan efektif

### API memberikan kemudahan dan manfaat dari sisi pengembangan aplikasi. Anda tidak perlu melakukan menghubungkan dua aplikasi untuk melakukan komunikasi. Cukup dengan menggunakan bantuan API, maka komunikasi dapat terjalin dengan baik.

### Menciptakan aplikasi yang bersifat fungsional

Manfaat lain dari penggunaan API ini adalah menciptakan aplikasi yang lebih fungsional dan memiliki struktur yang kompleks. Maksudnya, dalam menambahkan informasi tidak perlu melakukan *input* secara manual. Cukup dengan menggunakan bantuan API dapat menampilkan fitur yang sama dengan aplikasi tujuan.

1. Jelaskan perbedaan Native dan Hybrid pada mobile programming?

* 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. 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.
* Aplikasi hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau [Android](https://www.codepolitan.com/tag/android/). Aplikasi hybrid biasanya menggunakan browser untuk mengijinkan 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. Keuntungan membangun aplikasi hybrid diantaranya pemeliharaan project menjadi semakin mudah jika dibandingkan dengan aplikasi native. Aplikasi hybrid juga, bisa dibangun secara cepat untuk keperluan cross platform dan dana yang bisa menjadi lebih hemat jika dibandingkan dengan native.

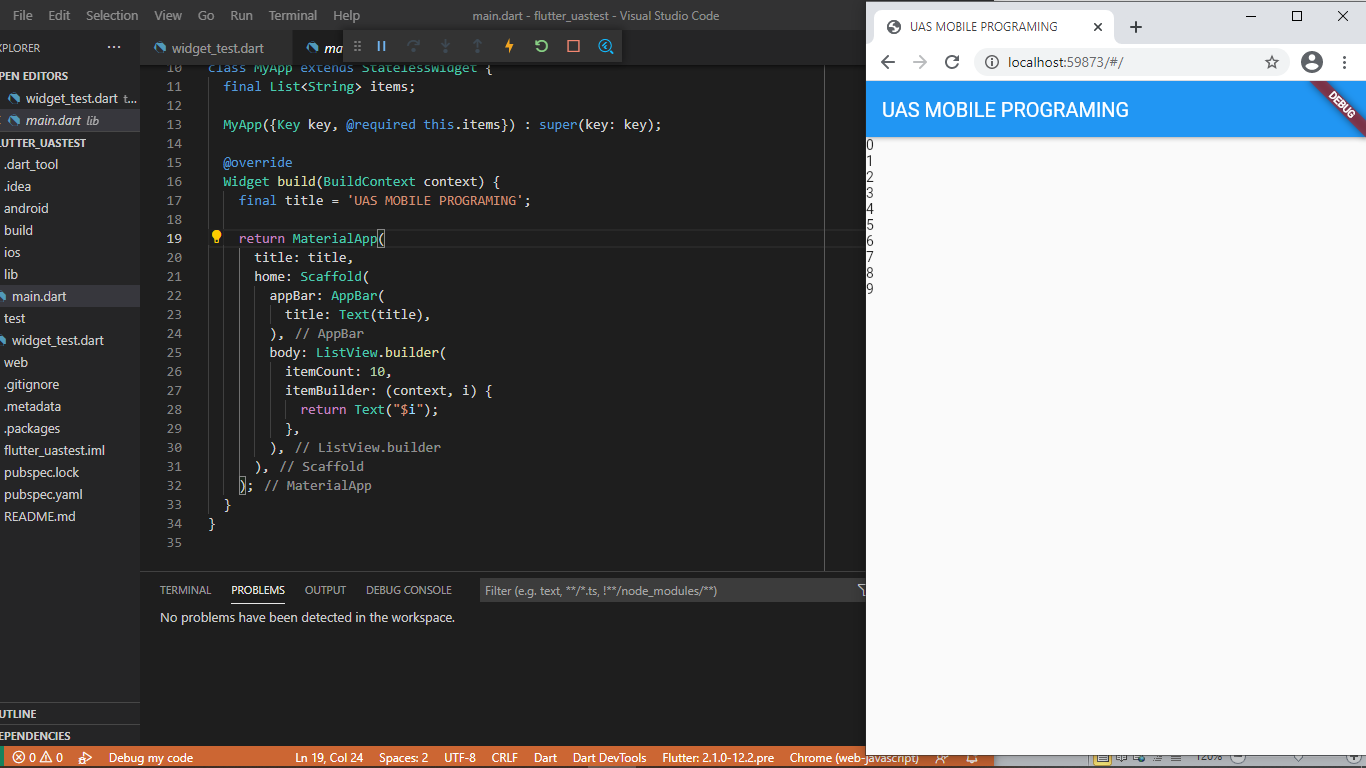
1. Jelaskan apa fungsi github!

* Github memiliki sejumlah fungsi yang menjadikan peranannya dibutuhkan dalam beragam aktivitas programming. Fungsi utama github adalah membantu penyimpanan repository. Namun tak hanya sebatas itu saja, masih ada lebih banyak fungsi dari github untuk mendukung *project* yang Anda garap.

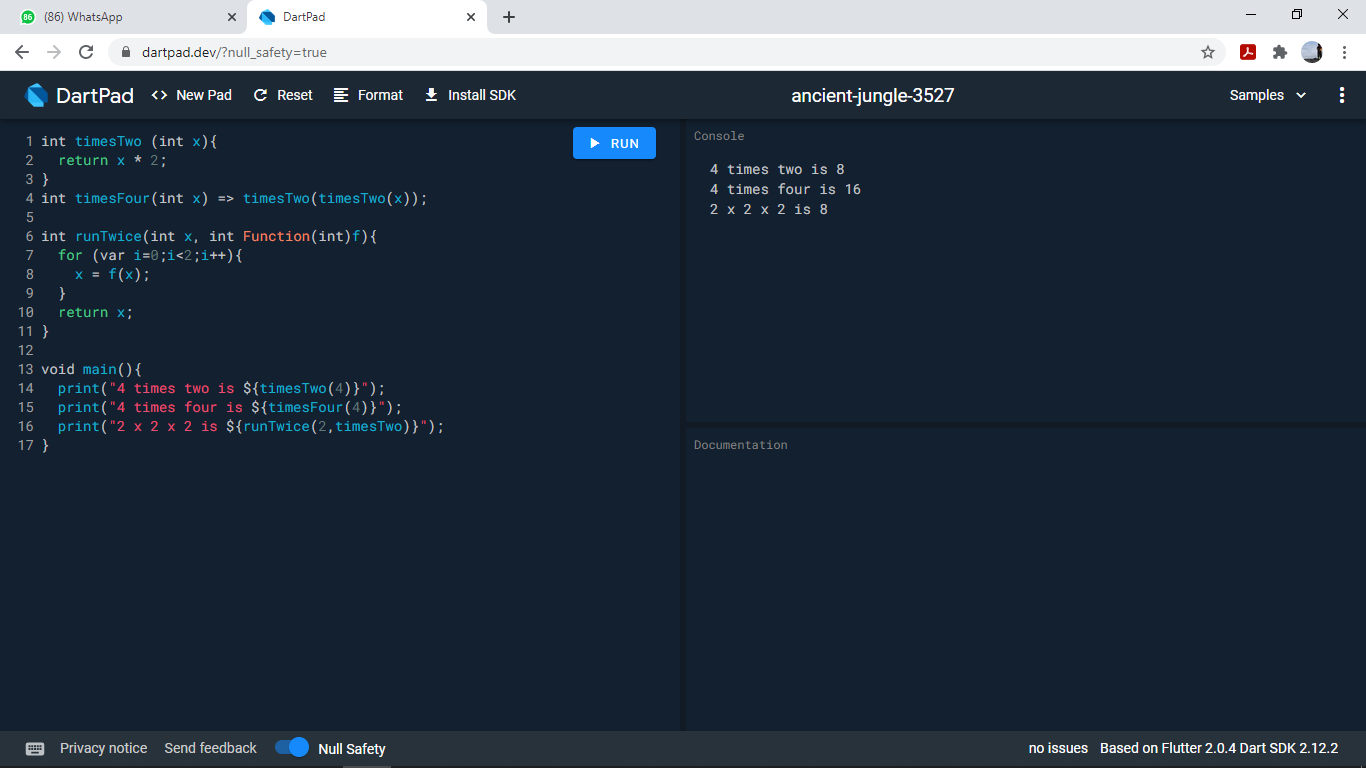
Beberapa fungsi github adalah:

* 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 memanajemen tugas. hingga;
* Menampilkan profil dan update dari Anda ke khalayak banyak.

1. Apa output dari script berikut !



1. Apa output dari script berikut !



1. Tuliskan sintak cara parsing JSON pada Flutter !

kita membutuhkan beberapa package diantaranya sebagai berikut. Tambahkan http dependencies dalam file pubspecs.yaml

dependencies:

flutter:

sdk: flutter

http: ^0.12.0

Selanjutnya isi di main.dart

import 'dart:async';

import 'dart:convert';

import 'package:flutter/foundation.dart';

import 'package:flutter/material.dart';

import 'package:http/http.dart' as http;

Future<List<Photo>> fetchPhotos(http.Client client) async {

final response = await client

.get(Uri.parse('https://jsonplaceholder.typicode.com/photos'));

// Use the compute function to run parsePhotos in a separate isolate.

return compute(parsePhotos, response.body);

}

// A function that converts a response body into a List<Photo>.

List<Photo> parsePhotos(String responseBody) {

final parsed = jsonDecode(responseBody).cast<Map<String, dynamic>>();

return parsed.map<Photo>((json) => Photo.fromJson(json)).toList();

}

class Photo {

final int albumId;

final int id;

final String title;

final String url;

final String thumbnailUrl;

Photo({this.albumId, this.id, this.title, this.url, this.thumbnailUrl});

factory Photo.fromJson(Map<String, dynamic> json) {

return Photo(

albumId: json['albumId'] as int,

id: json['id'] as int,

title: json['title'] as String,

url: json['url'] as String,

thumbnailUrl: json['thumbnailUrl'] as String,

);

}

}

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

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

final appTitle = 'Isolate Demo';

return MaterialApp(

title: appTitle,

home: MyHomePage(title: appTitle),

);

}

}

class MyHomePage extends StatelessWidget {

final String title;

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

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(title),

),

body: FutureBuilder<List<Photo>>(

future: fetchPhotos(http.Client()),

builder: (context, snapshot) {

if (snapshot.hasError) print(snapshot.error);

return snapshot.hasData

? PhotosList(photos: snapshot.data)

: Center(child: CircularProgressIndicator());

},

),

);

}

}

class PhotosList extends StatelessWidget {

final List<Photo> photos;

PhotosList({Key key, this.photos}) : super(key: key);

@override

Widget build(BuildContext context) {

return GridView.builder(

gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(

crossAxisCount: 2,

),

itemCount: photos.length,

itemBuilder: (context, index) {

return Image.network(photos[index].thumbnailUrl);

},

);

}

}