

LAPORAN AKHIR
RANGKUMAN MATERI
MSIB 2023 COMPLETE FRONT-END ENGINEER CAREER WITH FLUTTER



NAMA: MOHAMMAD NUR HOLIS
KELAS: FLUTTER (D)



PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR
SURABAYA
2023

DAFTAR ISI

COVER.....	1
DAFTAR ISI.....	2
BAB 1.....	4
MOBILE APP DEVELOPMENT	4
BAB 2.....	6
BASIC VERSION AND BRANCH MANAGEMENT (GIT)	6
BAB 3.....	9
BAHASA PEMROGRAMAN FUNDAMENTAL DART.....	9
BAB 4.....	11
BRANCH, LOOP, FUNCTION DART	11
BAB 5.....	13
ADVANCE FUNCTION, ASYNC AWAIT, COLLECTION DART	13
BAB 6.....	14
FUNDAMENTAL OBJECT ORIENTED PROGRAMMING	14
BAB 7.....	16
DEEP DIVE INTO DART OBJECT ORIENTED PROGRAMMING	16
BAB 8.....	19
INTRODUCTION FLUTTER AND FLUTTER WIDGET.....	19
BAB 9.....	23
INSTALLATION COMMAND LINE INTERFACE & FLUTTER PACKAGE MANAGEMENT	23
BAB 10.....	25
INSTALL FLUTTER AND TOOLS PART 1	25
BAB 11	27
INSTALL FLUTTER AND TOOLS PART 2	27
BAB 12.....	28
FLUTTER PLATFORM WIDGET	28
BAB 13.....	29
MENINGKATKAN KEMAMPUAN KOMUNIKASI EFEKTIF	29
BAB 14.....	30
FLUTTER LAYOUT	30
BAB 15.....	32

FLUTTER FORM	32
BAB 16.....	33
ADVANCE FORM INPUT.....	33
BAB 17.....	34
ASSETS.....	34
BAB 18.....	36
DIALOG BOTTOM SHEET	36
BAB 19.....	37
FLUTTER NAVIGATION	37
BAB 20.....	38
FLUTTER GLOBAL STATE MANAGEMENT.....	38
BAB 21.....	39
FLUTTER GLOBAL STATE MANAGEMENT (BLoC)	39
BAB 22.....	41
FLUTTER ANIMATION	41
BAB 23.....	42
STORAGE	42
BAB 24.....	43
MEMBUAT CV ATAU RESUME YANG EFEKTIF	43
BAB 25.....	44
REST API CLIENT	44
BAB 26.....	45
MVVM ARCHITECTURE	45
BAB 27.....	46
FINITE STATE MACHINE.....	46
BAB 28.....	47
UNIT TEST.....	47
BAB 29.....	48
UI TESTING.....	48
BAB 30.....	49
OUTPUT BUILD FLUTTER.....	49

BAB 1

MOBILE APP DEVELOPMENT

Mobile application development adalah istilah yang dipakai untuk menyebut proses pengembangan aplikasi yang beroperasi di perangkat mobile

A. pengertian software

software disebut juga perangkat lunak. atau merupakan sebuah data yang diprogram, disimpan, dan diformat secara digital dengan tujuan tertentu dan berbentuk program yang cara penggunaannya yaitu dengan dieksekusi oleh prosesor.

B. Fungsi software

- sebagai kebutuhan komputer
- mengatur perangkat keras pada komputer
- sebagai perantara atau penghubung antar software
- sebagai penerjemah perintah pada software yang lain yang masih ada di dalam bahasa mesin
- sebagai pengidentifikasi antar software

C. Jenis software

- web -- digunakan pada web browser -- perlu akses internet untuk membukanya
- desktop -- berjalan pada perangkat desktop dalam pengaplikasiannya perlu instalasi (umum) optional (tanpa akses internet masih dapat membukanya) umumnya tidak menggunakan akses internet
- mobile → berjalan pada smartphone → perlu langkah instalasi → dapat diunduh melalui store penyedia aplikasi → sebagian besar tanpa akses internet

D. Pengembangan software

merupakan serangkaian proses yang digunakan dalam mengembangkan perangkat lunak atau software. dalam pengembangannya terdapat banyak macam cara yang digunakan dalam mengembangkan perangkat lunak berdasarkan pendekatan-pendekatan yang ada didalam perangkat lunak tersebut.

1. waterfall pendekatan jenis ini memberikan pernyataan bahwa semuanya berjalan dengan lancar. misalnya apabila satu fase telah selesai maka tidak perlu ke fase sebelumnya
2. spiral merupakan pendekatan yang memungkinkan pengembangan secara incremental. Incremental yang dimaksud adalah pada setiap fasenya dapat dilalui lebih dari 1 kali dalam rangka pengembangan maupun perbaikan perangkat lunak yang dikembangkan.
3. agile pengembangan model ini mengharuskan tugas-tugas pengembangan dipisahkan ke dalam sebuah box. pendekatan ini berfokus pada adaptability dan kepuasan customer dengan waktu yang cepat

4. prototype yaitu pendekatan yang mengacu pada pembuatan prototipe perangkat lunak yang memiliki fungsionalitas terbatas[3]. Pendekatan ini memungkinkan untuk mendapat feedback dari customer dan membantu pengembang untuk memahami apa yang diinginkan oleh customer secara jelas.

E. Mobile apps development

mobile apps development adalah istilah yang digunakan untuk merujuk pada proses pengembangan aplikasi yang berjalan di perangkat seluler. Jadi, jika kita ingin tetap sederhana, mobile apps development adalah proses pembuatan aplikasi ponsel cerdas. Nantinya, pengembang harus memastikan bahwa aplikasi yang mereka buat dapat berjalan di jaringan dengan berbagai jenis sumber daya komputasi. Saat mengembangkan aplikasi seluler, programmer diminta untuk mengembangkan aplikasi yang mudah diakses. Berkat aksesibilitasnya yang baik, perangkat lunak ini dapat diakses oleh semua orang, termasuk penyandang disabilitas. berikut beberapa pendekatan dalam mengembangkan mobile app development

1. pengembangan secara hybrid pengembangan ini paling sering dipilih oleh seorang development karena menggunakan single code yang dapat di implementasikan di banyak platform seperti android dan ios.
2. pengembangan secara native pengembangan secara native juga menjadi favorite para web app developer, bedanya pengembangan secara hybrid dengan native ini basis code yang digunakan ditulis langsung pada bahasa pemrograman asli dari platform tujuan. dengan kata lain kode native hanya bisa di satu platform.
3. Cross-Platform Native Mobile Applications pengembangan jenis ini lebih kompleks dari 2 sebelumnya karena pengembangan model dapat di pakai atau digunakan di berbagai sistem operasi (Microsoft Windows, Linux, Mac OS, BSD dan lain sebagainya).

BAB 2

BASIC VERSION AND BRANCH MANAGEMENT (GIT)

Salah satu version kontrol sistem populer yang digunakan para developer untuk mengembangkan software secara bersama-sama. Git memungkinkan proses kolaborasi menjadi lebih mudah karena menyediakan bantuan untuk menggabungkan kode dan memastikan penggabungan kode berjalan dengan lancar. Git juga memungkinkan untuk kembali ke versi-versi sebelumnya sesuai yang diinginkan.

A. Version Control System

Sebuah kategori software yang membantu sebuah tim software mengelola perubahan pada source code. Version control system mencatat semua modifikasi yang dilakukan ke kode di sebuah database khusus. Jika terjadi kesalahan, pengembang dapat mempratinjau versi kode sebelumnya untuk memperbaiki kesalahan dengan gangguan seminimal mungkin.

Untuk proyek perangkat lunak apa pun, kode sumber merupakan aset penting. Bagi kebanyakan orang, kode sumber adalah gudang pengetahuan dan pemahaman yang tak ternilai tentang berbagai masalah yang telah diperhatikan dan diperbaiki oleh pengembang.

Kontrol versi melindungi kode sumber dari gangguan atau kesalahan manusia dan konsekuensi yang tidak diinginkan. Pengembang perangkat lunak terus-menerus menulis kode sumber baru dan memodifikasi kode sumber yang ada. Kode untuk proyek, aplikasi, atau komponen perangkat lunak biasanya diatur dalam folder atau hierarki file.

Pengembang dalam tim mungkin mengerjakan fitur baru sementara pengembang lain memperbaiki bug dengan memodifikasi kode. Jadi, dengan pohon file, setiap pengembang dapat membuat perubahan pada pohon file apa pun.

- Perbedaan git dan github

GIT dan GitHub sebenarnya adalah dua platform yang didirikan oleh perusahaan yang melayani tujuan yang sama tetapi berbeda fungsinya. Kedua platform sangat membantu programmer bekerja sebagai tim untuk menyusun kode skrip. Semua pekerjaan juga dapat dengan mudah dipantau dan dievaluasi menggunakan sistem kontrol

GIT	GITHUB
Dikelola oleh The Linux Foundation	Diakuisisi oleh Microsoft
Diakses secara offline	Diakses secara online
Tidak memakai fitur user management	Menggunakan fitur user managemen
Install software di penyimpanan lokal	Di-host melalui layanan cloud

GIT	GITHUB
Fokus pada version control dan code sharing	Fokus pada source code hosting yang terpusat

B. Essential Git Commands

Dalam fungsi utamanya sebagai sistem kontrol versi, git menyediakan berbagai perintah yang dapat digunakan pengguna. Berikut adalah beberapa perintah dasar yang perlu diketahui saat menggunakan git:

- `git status`. ini akan menampilkan kondisi berkas saat ini. Pengguna dapat melihat apakah berkas telah diubah sejak terakhir kali disimpan dan melihat berkas mana yang siap untuk di commit
- `git add <nama_file>`. Git add memungkinkan pengguna untuk melacak file dan folder yang ada dan memindahkan status file dan folder tersebut ke tahap pementasan. Setelah itu, Anda dapat menggunakan perintah komit untuk menyimpan riwayat file yang dimaksud secara permanen.
- `git commit -m "commit message"`. Setelah menjalankan perintah `git add` terlebih dahulu, `git commit` baru dapat dibuat. Jika `git add` memindahkan status file ke pementasan, `git commit` akan menyimpan file dalam riwayat git
- `git branch <nama_branch_baru>`. digunakan jika ingin membuat sebuah percabangan baru untuk pengerjaan program. Dengan adanya cabang baru ini, maka cabang utama dari program tidak akan terganggu dan jika terjadi error pada cabang baru, maka cabang utama tidak akan mengalami error
- `git checkout <nama_branch>`. Git checkout memungkinkan pengguna untuk berpindah antar branch agar dapat menulis kode di branch tersebut. Selain itu, pengguna juga dapat pindah ke commit lampau menggunakan commit hash untuk melihat kode yang pernah ditulis saat itu.
- `git clone`. memungkinkan pengguna untuk menyalin kode yang ada di remote repository (seperti github dan gitlab) ke local repository. Proses clone dapat dilakukan dengan mendapatkan https dari remote repository terlebih dahulu.
- `git pull <nama_branch>`. Git pull memungkinkan pengguna untuk mendapatkan kode terbaru dari remote repository. Dengan menggunakan perintah ini, maka kode yang ada di remote repository akan di merge dengan kode yang ada di local.
- `git push <nama_branch>`. Git push memungkinkan pengguna untuk mengirimkan kode yang ada di local repository ke remote repository agar dapat dilihat dan di pull kembali oleh rekan kerja yang lain
- `git log`. Git log digunakan untuk melihat commit history dari program yang kita buat. Dengan `git log` kita akan memperoleh commit hash dan dapat berpindah ke salah satu commit yang telah dibuat.
- `git merge <nama_branch>`. Git merge digunakan jika ingin menggabungkan dua branch yang berbeda. Contohnya jika kita ingin menggabungkan branch utama dengan branch baru yang telah kita buat. Proses penggabungan branch menggunakan merge akan menghasilkan satu commit baru

- git rebase <nama_branch>. Git rebase juga bisa digunakan jika ingin menggabungkan dua branch yang berbeda. Namun, berbeda dengan merge, rebase tidak akan menghasilkan commit baru dalam penggunaannya.

C. fitur dan kegunaan

1) performance

Algoritme yang diimplementasikan di dalamnya mendapat manfaat dari pengetahuan mendalam tentang fitur umum pohon file kode sumber dan bagaimana mereka biasanya berubah seiring waktu dan pola aksesnya. Tidak seperti beberapa perangkat lunak kontrol versi, Git tidak tertipu oleh nama file saat menentukan penyimpanan dan riwayat versi apa yang seharusnya menjadi pohon file.

2) Security

Git dirancang dengan integritas kode sumber terkelolanya sebagai prioritas utama. Konten file serta hubungan direktori file yang benar, versi, tag, dan komit terdapat dalam repositori dan diamankan dengan algoritme hashing yang aman secara kriptografis (SHA1). Algoritme ini melindungi kode dan riwayat perubahan dari perubahan berbahaya. Selain itu, ini memastikan ketertelusuran penuh sejarah. Dengan Git, Anda memiliki riwayat konten kode sumber asli. Beberapa sistem kontrol versi lain tidak memiliki perlindungan terhadap perubahan rahasia di kemudian hari. Ini bisa menjadi masalah keamanan utama bagi organisasi mana pun yang bergantung pada pengembangan perangkat lunak

3) Flexibility

Git sangat fleksibel dalam beberapa hal: mendukung beragam alur kerja non-linear, kinerja dalam proyek kecil dan besar, dan kompatibilitas dengan banyak sistem dan protokol yang ada. Git dirancang untuk menangani cabang dan operasi yang memengaruhi cabang, dan tag (seperti penggabungan atau perubahan) juga disimpan sebagai bagian dari riwayat perubahan. Tidak semua sistem kontrol versi memiliki tingkat pelacakan ini.

BAB 3

BAHASA PEMROGRAMAN FUNDAMENTAL DART

A. Dart

Dart merupakan sebuah bahasa pemrograman yang di rancang untuk membuat aplikasi dengan cepat dan mudah. dart adalah aplikasi yang bekerja di bagian front end. seperti web, desktop dan mobile.

Kelebihan dart:

- a. type safety menjamin konsistensi tipe data
- b. null menjaga keamanan dari data bernilai hampa (null)
- c. rich standard library hadir dengan banyak dukungan library internal
- d. multiplatform mampu berjalan pada semua jenis perangkat

B. Dasar pemrograman dart

- menampilkan teks
yaitu hal sederhana yang dapat dilakukan dalam dart
- program sederhana
 1. fungsi main
 - bagian yang dijalankan pertama kali dalam dart
 - memiliki tipe data void atau int
 - memiliki nama main
 - print (adalah perintah yang menjalankan tampilan di layar)
 2. komentar
 - yaitu teks atau baris kata yang tidak dijalankan dengan tujuan:
 - memberi catatan kepada setiap kode
 - mencegah perintah untuk dijalankan

jenis-jenis komentar

 - garis miring 2 kali (//) yaitu jenis komentar yang memberikan catatan untuk satu baris saja
 - awal garis miring bintang (/) *akhir bintang garis miring (/)* yaitu untuk memberikan catatan lebih dari satu baris
 3. Variabel
 - digunakan untuk menyimpan data
 - memiliki nama
 - memiliki tipe data
 4. konstanta
 - digunakan untuk menyimpan data
 - memiliki nama memiliki tipe data
 - nilai tetap (tidak dapat di rubah) contoh:

```
void main(){ final usia = 18; }
```
 5. Tipe data
 - jenis data yang dapat dikelola
 - tipe data sederhana disebut primitiv data type

jenis-jenis tipe data

- int: untuk bilangan bulat
- double: bilangan pecahan penulisan pecahan harus dengan tanda titik(.)
- bool: true/false
- string: teks penulisan string diawali dengan tanda kutip("")

6. Operator

- digunakan untuk operasi data
- data yang dikelola disebut dengan operand
- arithmetic: untuk perhitungan aritmatika(*,+,/,-)
- assignment: memberikan nilai pada variabel
- comparison: membandingkan kesetaraan nilai
- logical: membandingkan beberapa kondisi

BAB 4

BRANCH, LOOP, FUNCTION DART

A. Pengambilan keputusan

yaitu menentukan alur program pada kondisi tertentu

1. IF

- memerlukan nilai bool (dari operator logical atau comparison)
- menjalankan bagian proses jika nilai bool bernilai true
- contoh if (nilai_bool) { //proses jika nilai_bool adalah true }

2. IF-ELSE

- berjalan dengan is
- menambah alternatif jika nilai_bool adalah false
- contoh if (nilai_bool) { //proses jika nilai_bool adalah true } else { //proses jika nilai_bool adalah false }

b. sisipan IF-ELSE

- berjalan dengan is
- menambah alternatif jika nilai_bool adalah false
- menambah pengujian nilaibool lain
- contoh if (nilai_bool) { //proses jika nilai_bool adalah true } else { //proses jika nilai_bool adalah false //dan nilai_bool adalah true }

B. Pperulangan

yaitu menjalankan perintah proses berulang kali

1. For

- diketahui berapa kali perulangan terjadi
- memerlukan nilai awal
- memerlukan nilai bool, jika true maka perulangan dilanjutkan
- memerlukan pengubah nilai

2. While

- tidak diketahui berapa kali perulangan terjadi
- memerlukan nilai bool, jika trus maka peulangan dilanjutkan
- contoh: while (nilai_bool) { //proses berulang jika nilai_bool adalah true }

c. Do While

- mengubah bentuk while
- proses dijalankan minimum sekali, akan diteruskan jika nilai bool adalah true
- contoh: do { //proses berulang jika nilai_bool adalah true } while (nilai_bool);

C. Break dan continue

break dan continue adalah cara lain untuk menghentikan atau melanjutkan perulangan

deskripsi

- perulangan menggunakan nilai bool untuk lanjut atau berhenti

- break dan continue dapat menghentikan perulangan dengan mengabaikan nilai bool
- continue dapat menghentikan satu kali proses

perbedaan

- break: menghentikan seluruh proses perulangan
- continue: menghentikan satu kali proses perulangan

D. Fungsi

yaitu kumpulan kode yang dapat digunakan ulang

deskripsi

- kumpulan perintah
- dapat digunakan berkali-kali cukup mengubah fungsi sekali, pengguna lainnya akan ikut berubah

BAB 5

ADVANCE FUNCTION, ASYNC AWAIT, COLLECTION DART

A. Fungsi (lanjutan)

1. anonymous function

- yaitu fungsi yang tidak memiliki nama dan fungsi tersebut di jadikan sebagai data
- contoh () {`//perintah yang dijalankan saat fungsi dipanggil`}

2. arrow function

- dapat memiliki nama atau tidak
- berisi 1 data (dari proses maupun data statis)
- nilai return fungsi diambil dari data
- contoh () => `proses_yang_dijalankan_saat_proses_dioanggil()`;

B. Async-await

Fitur ini mempermudah kita dalam menangani proses asynchronous.Async/Await merupakan sebuah syntax khusus yang digunakan untuk menangani Promise agar penulisan code lebih efisien dan rapih.

C. Tipe data future

Future Builder adalah widget yang disediakan oleh flutter untuk mendukung proses asinkron.

1. tipe data dynamic

Tipe data ini merupakan tipe data yang akan menyesuaikan dengan jenis data yang dibawahnya, cukup menarik karena mirip seperti penggunaan any pada Typescript

D. Collection

yaitu objek yang menyimpan banyak data seperti List, Set, dan Map

1. List

yaitu tempat untuk menyimpan data (elemen) secara berbaris. List pada Dart berguna untuk menyimpan data secara berurutan, karena mengandalkan index yang ada di List dan index yang ada di Dart sendiri dimulai dari angka 0. Kita bisa menyimpan tipe data apapun ke dalam List.

contoh: `List nama = ['nama1', 'nama2', 'nama3'];`

2. Set

merupakan syntax yang mirip dengan list, namun set ini berbeda dengan list, karena list dapat menampung data yang sama, sedangkan set ini tidak boleh menampung nilai yang sama.

3. Map

yaitu tempat untuk menyimpan data secara key-value. key berguna selayaknya index pada list. Map bisa menampung tipe data apapun layaknya List dan Set. Perlu dicata bahwa Map tidak digunakan untuk menyimpan informasi secara berurutan. contoh:

`Map namaVariabel = {}; //sekilas mirip Set`

`Map kota = { 'Jakarta': 'Indonesia', 'London': 'England', 'New York': 'USA', 'Tokyo': 'Japan' }; print(kota['Jakarta']); //Indonesia`

BAB 6

FUNDAMENTAL OBJECT ORIENTED PROGRAMMING

A. Perkenalan object oriented programming

merupakan salah satu metode yang digunakan dalam bahasa pemrograman yang didasari pada sebuah class. Tujuan dari dirancangnya OOP yaitu membantu para developer dalam mengembangkan model yang sudah ada.

- a. Deskripsi
 - disebut OOP
 - program disebut dalam bentuk abstraksi objek
 - data akan di proses melalui abstraksi tersebut
 - o Keuntungan
 - mudah di-troubleshoot
 - mudah digunakan
 - Digunakan dalam berbagai bahasa pemrograman lainnya seperti:
 - C++
 - java
 - javascript
 - python
- 2) Komponen
- class
 - object
 - property
 - methode
 - inheritance
 - generics

B. Class

class adalah rancangan yang mendefinisikan variabel dan methode-metode pada seluruh object tertentu. class juga dapat menampung isi dari program yang akan dijalankan.

Deskripsi

- abstraksi dari sebuah program (object)
- memiliki ciri-ciri yang disebut property
- memiliki sifat dan kemampuan yang disebut methode

Membuat class

- menggunakan kata kunci *class*
- memiliki nama
- detail class diletakkan di dalam kurung kurawal ({})
- contoh: `class Hewan { //property //method }`

Membuat object

- bentuk sebenarnya dari *class* disebut sebagai instance of class

- diperlukan seperti data
- contoh: `void main() { var n1 = Hewan(); var n2 = Hewan(); var n3 = Hewan(); }`

C. Property

property merupakan sekumpulan data yang terdapat dalam sebuah class.

Deskripsi

- ciri-ciri suatu class
- hal-hal yang dimiliki suatu class
- memiliki sifat seperti variabel

Membuat property

- membuat property seperti membuat variabel, tetapi terletak didalam class
- contoh: `class Hewan { var mata = 0; var kaki = 0; }`

Mengakses property

- mengakses property seperti menggunakan variabel, tetapi melalui sebuah object
- contoh: `void main() { var h1 = Hewan(); print(h1.mata); }`

D. Methode

methode merupakan sekumpulan program yang akan mempunyai nama, methode juga menjadi sarana bagi programmer untuk memecah program menjadi bagian-bagian kecil agar lebih lebih kompleks sehingga dapat digunakan berulang-ulang. Method merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object.

Deskripsi

- sifat suatu class
- aktivitas yang dapat dikerjakan suatu class
- memiliki sifat seperti fungsi

Membuat method

- membuat method seperti fungsi, tetapi terletak didalam class
- contoh: `class Hewan { void bersuara () { print(""); } }`

Menjalankan method

- menjalankan method seperti menggunakan fungsi, tetapi melalui sebuah object
- contoh: `void main() { var h1 = Hewan(); h1.bersuara(); }`

BAB 7

DEEP DIVE INTO DART OBJECT ORIENTED PROGRAMMING

A. Constructor

merupakan fungsi khusus dari class yang bertanggung jawab dalam menginisialisasi nilai awal dari properti yang berada di dalam class. class terdiri dari tiga jenis, yaitu default constructor dan named constructor, dan factory constructor.

Deskripsi

- method yang dijalankan saat pembuatan object
- dapat menerima parameter
- tidak memiliki return
- nama sama dengan class **contoh:** class hewan { var mata; var kaki; hewan(){
//constructor mata=0; kaki=0; } }

-

B. Inheritance

inheritance merupakan konsep oop yang sering digunakan dalam membuat program. inheritance sendiri biasanya disebut keturunan atau pewarisan. pada sebuah code inheritance adalah penulisan sebuah kelas baru berdasarkan kelas yang sudah lama. ciri-cirinya terdapat kata "extends"

Deskripsi

- membuat class baru dengan memanfaatkan class yang sudah ada
- bertujuan agar kemampuan class yang ada dapat dimiliki oleh class baru **contoh:** class kambing extends hewan { kambing(){ mata=2; kaki=4; } }

-

C. Method overriding

method overriding adalah sebuah situasi dimana sebuah method class turunan menimpa method class parent. terjadi karena terdapat method yang sama baik di child class maupun parent class

Deskripsi

- menulis ulang method yang ada pada super-class
- bertujuan agar class memiliki method yang sama dengan proses yang berbeda
- menggunakan overriding
- dilakukan pada class yang melakukan inheritance
- method sudah ada pada class induk
- method ditulis ulang seperti membuat method baru pada class anak
- ditambahkan tanda @override di baris sebelum method di buat.

-

D. Interface

interface merupakan sintaks yang harus dipatuhi oleh setiap entitas. interface ini menentukan sekumpulan metode yang tersedia pada suatu objek. dalam bahasa dart interface tidak memiliki sintaks. deklarasi class sendiri adalah interface di dart.

Deskripsi

- berupa class
- menunjukkan method apa saja yang ada pada suatu class
- seluruh method wajib di override
- digunakan dengan menggunakan implements
- menggunakan interface
- sekilas mirip inheritance
- pada class yang melakukan implements, wajib melakukan overrode semua method yang ada pada class induk

E. Abstract class

abstract class adalah class yang memang tidak secara rinci mendefinisikan objek tertentu, sehingga class abstrak biasa dijadikan sebagai gambaran umum.

Poin penting abstract class di Dart sendiri antara lain:

- Abstract class biasa ditandai dengan keyword abstract sebelum keyword class.
- Dengan class menjadi abstract, maka kita bisa membuat method abstract dengan menghilangkan kurung kurawal atau body {} pada method tersebut.
- Penggunaan abstract class mewarisi ke child class dengan keyword extends.
- Hanya method yang wajib dioverride atau ditulis ulang ke child class

Deskripsi

- berupa class abstract
- menunjukkan method apa saja yang ada pada suatu class
- digunakan dengan menggunakan extends
- tidak dapat dibuat object
- tidak semua method harus di-override

F. Polymorphism

Polymorphism merupakan penggunaan suatu item baik interface, fungsi, dan lain-lain pada berbagai macam jenis objek maupun entity yang berbeda-beda dengan syarat suatu objek atau entity tersebut memiliki relasi yang menjembatani agar akses ke item tersebut dapat didapatkan.

Deskripsi

- kemampuan berubah menjadi bentuk lain
- tipe data yang dapat digunakan adalah super class
- dapat dilakukan pada class dengan extends atau implement

G. Generics

Generics merupakan konsep dalam membuat sebuah function, method, atau class supaya bisa digunakan ulang untuk beberapa tipe data. Tipe data tersebut nantinya akan ditentukan ketika pemanggilan function, method, atau class generic ini, sehingga tetap memastikan Type-Safety

pada kode kita selagi memberi keleluasaan untuk menggunakan ulang functions atau method2 yang ada pada suatu class untuk berbagai tipe data.

Deskripsi

- dapat digunakan pada class atau fungsi
- memberi kemampuan agar dapat menerima data dengan tipe yang berbeda
- tipe data yang diinginkan, ditentukan saat membuat class atau menjalankan fungsi
- membuat class dengan generics
- class hadiah dapat dimasukkan data dengan tipe T
- tipe T dapat digunakan diseluruh class hadiah
- contoh: `class hadiah {var isi; hadiah(T i){ isi=i; } }`

BAB 8

INTRODUCTION FLUTTER AND FLUTTER WIDGET

A. Flutter

flutter merupakan platform yang digunakan para developer untuk membuat aplikasi multiplatform hanya dengan satu basis coding (codebase). aplikasi yang dihasilkan bisa dipakai di berbagai platform, mobile, iOS, web dan desktop. Sebelum resmi diluncurkan pada Desember 2018, Flutter sebenarnya telah dikembangkan oleh Google sejak 2015. Lalu, di tahun 2019, popularitas Flutter mulai meroket dan banyak developer berbondong-bondong menggunakan Flutter. terdapat dua komponen penting yang dimiliki flutter:

1. software development kit (SDK) yaitu sekumpulan tools yang dapat membuat aplikasi bisa dijalankan di berbagai platform
2. framework UI merupakan komponen UI, seperti teks, tombol, navigasi, dan lainnya yang dapat digunakan di berbagai kebutuhan.

kelebihan framework flutter

- 1) pengembangan aplikasi yang lebih mudah dan cepat karena tersedianya fitur-fitur yang membantu proses pengembangan jadi lebih mudah dan cepat
- 2) custom user interface yang menarik dalam framework flutter ini kita bisa memanfaatkan koleksi widget untuk tampilan aplikasi, seperti layout, navigasi, style tema, hingga pengaturan tanpa scrolling.
- 3) performa seperti aplikasi native flutter mengkombinasikan kode aplikasi native dan mesin render yang canggih. sehingga, aplikasi tetap bisa ditampilkan dengan baik di berbagai platform.
- 4) biaya pengembangan lebih hemat adanya fitur Hot Reload, UI widget, data backend, platform integration, aksesibilitas dan internasionalisasi, packages dan plugin yang memudahkan developer membuat aplikasi. Jadi, mereka bisa menghemat tenaga, waktu, dan biaya lebih banyak.

i. kekurangan flutter

- 1) library pihak ketiga yang belum banyak dalam framework flutter library yang disediakan flutter masih belum banyak, sehingga kita harus lebih kreatif dan melakukan banyak uji coba.
- 2) aplikasi yang dihasilkan berukuran besar hal ini akan memakan banyak ruang penyimpanan pada perangkat, jika penyimpanan pengguna terbatas, maka performa aplikasi terpengaruh.
- 3) perlu belajar bahasa pemrograman baru untuk menggunakan flutter kita perlu mempelajari bahasa pemrograman baru, yaitu bahasa dart.

B. Stateless widget

Stateless widget adalah widget yang hanya bertugas untuk menampilkan sesuatu secara statis. Tanpa melakukan tracking perubahan data dari waktu ke waktu

Deskripsi

- tidak bergantung pada perubahan data
- hanya berfokus pada tampilan saja
- dibuat dengan extends pada class StatelessWidget

membuat stateless widget

```
class MyWidget extends StatelessWidget {
  const MyWidget({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: Scaffold(
        body: Text('Selamat datang di Flutter.'),
      ),
    );
  }
}
```

C. Statefull widget

Stateful Widget merupakan widget yang dinamis dan dapat berubah. Berbanding terbalik dengan stateless, stateful widget dapat mengupdate tampilan, merubah warna, menambah jumlah baris dll. widget yang dapat berubah maka itulah stateful widget.

Deskripsi

- mementingkan pada perubahan data
- dibuat dengan extends pada class StatefulWidget
- 1 widget menggunakan 2 class (widget dan state)

membuat stateful widget

```
class MyWidget extends StatefulWidget {
  const MyWidget({Key? key}) : super(key: key);

  @override
  createState() => MyWidgetState();
}
```



D. Built in widget

yaitu membangun tampilan pada sebuah aplikasi menggunakan flutter seperti tombol, gambar, teks, list, ikon, bahkan satu layar pada handphone tersebut merupakan sekumpulan dari banyak widget.

Deskripsi

- widget yang dapat digunakan langsung digunakan
- sudah terinstall bersama flutter

general widget

- 1) scaffold Scaffold adalah widget utama untuk membuat sebuah halaman pada flutter
- 2) stateless Stateless Widget adalah widget yang berfungsi untuk menampilkan hal-hal yang sifatnya statis mudahnya setelah data ditampilkan maka kita tidak akan merubahnya lagi
- 3) statefull Statefull Widget adalah widget yang digunakan untuk menampilkan data-data yang dinamis atau data yang kita telah tampilkan sewaktu-waktu dapat mengalami perubahan.
- 4) layout Untuk mengatur tata letak atau posisi widget, maka kita akan mengenal dua buah widget yaitu Row dan Column center yaitu mengatur teks ke tengah seperti halnya di ms. word.
- 5) text Text berfungsi untuk menampilkan sebuah teks biasa, atau bisa kita berikan style dengan menambahkan property style.
- 6) Icon Widget ini untuk menggunakan icon yang telah disediakan.
- 7) Container Container merupakan widget yang fungsinya untuk membungkus widget lain sehingga dapat diberikan nilai seperti margin, padding, warna background, atau dekorasi.
- 8) SizedBox yaitu untuk membuat box, widget ini biasanya digunakan untuk menambahkan jarak baik secara vertikal atau horizontal tergantung property yang kita atur.
- 9) Button Terdapat 3 widget Button yang umumnya dipakai yaitu RaisedButton, FlatButton, dan FlatButton:

- raised button dan Material tombol akan sedikit menonjol.
- flat button tombolnya akan datar tanpa adanya efek-efek seperti bayangan dan lain-lain.

- 10) TextFormField TextFormField merupakan widget yang berguna untuk membuat form untuk diisi user.
- 11) InkWell InkWell digunakan untuk menambahkan action pada widget seperti action onTap.
- 12) Simple ListView ListView digunakan untuk menampilkan daftar item dalam jumlah yang banyak.
- 13) Navigator Push Untuk membuka halaman baru, kita bisa menggunakan Navigator.push.
- 14) Navigator Pop Sedangkan pop, kita menghilangkan halaman lapisan teratas atau yang sedang tampil, sehingga akan kembali ke halaman sebelumnya.

BAB 9

INSTALLATION COMMAND LINE INTERFACE & FLUTTER PACKAGE MANAGEMENT

A. Flutter CLI

Command line Interface adalah perintah yang digunakan programmer untuk berinteraksi langsung dengan komputer. Dalam flutter, command Line digunakan sebagai perintah untuk mengembangkan aplikasi.

Deskripsi

- alat yang digunakan untuk berinteraksi dengan flutter SDK
- perintah dijalankan dalam terminal

perintah yang wajib diketahui di dalam flutter CLI

- 1) Create Flutter App MacOS: `$ flutter create flutter_app` Windows: `flutter create flutter_app` digunakan untuk membuat project aplikasi Flutter yang baru.
- 2) Flutter Run MacOS: `$ flutter run` Windows: `flutter run` digunakan untuk menjalankan aplikasi proyek kita.
- 3) Flutter Doctor MacOS: `$ flutter doctor` Windows: `flutter doctor` perintah untuk menampilkan informasi software yang dibutuhkan
- 4) Flutter Upgrade MacOS: `$ flutter upgrade` Windows: `flutter upgrade` digunakan untuk melakukan pembaruan Flutter SDK kita dengan versi yang stabil.
- 5) Flutter Build Apk MacOS: `$ flutter build apk` Windows: `flutter build apk` Perintah ini digunakan untuk membangun aplikasi Android apabila sudah mencapai tahap penyelesaian.
- 6) Flutter Install MacOS: `$ flutter install` Windows: `flutter install` Perintah ini digunakan untuk menginstall aplikasi ke emulator atau gawai kita.
- 7) Install Packages MacOS: `$ flutter packages get` dan `$ flutter packages upgrade` Windows: `flutter packages get` dan `flutter packages upgrade` Pub merupakan file yang mengelola library/package. Untuk menambahkan library, digunakan untuk flutter packages get sedangkan, jika ingin pembaruan gunakan flutter packages upgrade.
- 8) Test Flutter MacOS: `$ flutter test testExample.dart` Windows: `flutter test testExample.dart` Perintah ini digunakan untuk menjalankan file yang kita tes pada aplikasi.
- 9) Analyze Code MacOS: `$ flutter analyze` Windows: `flutter analyze` Perintah ini digunakan untuk menganalisis kode yang dijalankan.
- 10) Connected Device MacOS: `$ flutter devices` Windows: `flutter devices` digunakan untuk menampilkan semua perangkat yang tersambung dengan Flutter.
- 11) Cleaning Build Files MacOS: `$ flutter clean` Windows: `flutter clean` digunakan untuk membersihkan file-file yang tidak dibutuhkan FLutter. bertujuan agar menghemat kapasitas data proyek sehingga ukuran data menjadi lebih kecil.

B. Packages management

Stateless widget adalah widget yang hanya bertugas untuk menampilkan sesuatu secara statis. Tanpa melakukan tracking perubahan data dari waktu ke waktu

Deskripsi

- flutter mendukung sharing packages
- packages dibuat developers lain
- mempercepat pengembangan aplikasi karena tidak perlu membuat semuanya dari awal atau from scratch
- mendapatkan packages di website pub.dev

cara menambahkan packages

- 1) cari package di pub.dev
- 2) copy baris dependencies yang ada di bagian instalasi
- 3) buka `pubspec.yaml`
- 4) paste barisnya di bawah dependencies `pubspec.yaml`
- 5) run flutter di pub get terminal
- 6) import package di file dart agar bisa digunakan
- 7) stop atau restart aplikasi jika dibutuhkan

BAB 10

INSTALL FLUTTER AND TOOLS PART 1

A. Sistem requirements

1. sistem requirements

- sistem operasi windows 7 spi or later (64-bit)
- space kosong 1.64GB (tidak termasuk space tool lainnya)
- tools pendukung window powershell 5.0 dan git for windows
- ram 8 GB (rekomendasi 12 GB)
- SSD 120 GB (rekomendasi untuk menjalankan emulator di android studio)

B. Install flutter SDK

- langkah-langkah

- step 1: download flutter SDK
- step 2: ekstrak zip file hasil download di C:\src\
- step 3: update path
- step 4: pergi ke environment variabels
- step 5: pilih path yang di user variabel
- step 6: tambahkan lokasi flutter\bin
- step 7: tekan OK
- step 8: jalankan perintah flutter di terminal
- step 9: jalankan perintah di flutter doctor di terminal

C. Install android studio

- langkah-langkah

- step 1: download android studio
- step 2: double klik file installer android studio
- step 3: klik next
- step 4: install
- step 5: finish
- step 6: do not import seting
- step 7: next
- step 8: pilih ui theme
- step 9: klik next
- step 10: pilih accept di persetujuan licences
- step 11: tunggu proses download selesai
- step 12: klik finish
- step 13: jalankan perintah flutter doctor

D. Install android toolchain

- langkah-langkah

- step 1: klik more actions di android studio
- step 2: klik SDK kits
- step 3: pastikan beberapa SDK tool sudah terinstall atau belum

- step 4: centang tool diperlukan, klik apply dan OK
- step 5: pilih accept untuk permintaan persetujuan dan klik next untuk menginstall tool nya
- step 6: klik next emulator settings biarkan secara default
- step 7: klik finish
- step 8: apply dan klik OK
- step 9: jalankan perintah flutter doctor
- step 10: jalankan perintah flutter doctor--android-licences
- step 11: jalankan kembali perintah flutter doctor

BAB 11

INSTALL FLUTTER AND TOOLS PART 2

A. Membuat project dart

langkah-langkah

- step 1: pada bagian menu, pilih view-->command paltte
- step 2: pilih console aplication
- step 3: ketikan nama project dart nya, kemudian tekan enter
- step 4: menjalankan project dart

B. Membuat project flutter

langkah-langkah

- step 1: pada bagian menu bar, pilih view-->command palette
- step 2: aplication
- step 3: ketikan nama project flutter nya, kemudian tekan enter

C. Membuat android emulator

langkah-langkah

- step 1: buka android studio, klik more actions-->virtual device manager
- step 2: klik create device
- step 3: download sistem image
- step 4: klik finish-->klik next
- step 5: biarkan pengaturannya secara default, lalu klik finish
- step 6: menjalankan emulator dari visual studio code
- step 7: emulator berhasil dijalankan
- step 8: menjalankan project flutter pada emulator

BAB 12

FLUTTER PLATFORM WIDGET

A. MaterialApp

sebuah parent dimana didalamnya tersebut menerapkan material design, dan juga memiliki kontrol untuk mengatur route, theme, supported locales, dan lain.lain

1. Scaffold

berperan untuk mengatur struktur layout dengan mengimplementasikan material design, dan memiliki kemampuan untuk membuat appbar, drawers, snack bars, bottom sheet dan sebagainya.

2. AppBar

yaitu salah satu properti yang dimiliki scaffold widget untuk membuat bar pada aplikasi. contoh: menampilkan teks.

B. CupertinoApp

Secara default ketika kita pertama kali buat proyek di Flutter maka, defaultnya adalah akan menggunakan material.dart yang mana didalam package tersebut terdapat beberapa widget dengan style Android atau Material Design.

BAB 13

MENINGKATKAN KEMAMPUAN KOMUNIKASI EFEKTIF

A. Tujuan sesi pelatihan

1. mampu berkomunikasi yang efektif ditempat kerja
2. mengetahui kelebihan dan kekurangan dalam berkomunikasi
3. mengetahui prosedur menjadi komunikator dan komunikasi yang baik

B. Apa itu komunikasi efektif

yaitu pertukaran informasi yang baik, verbal maupun nonverbal dalam sebuah organisasi (perusahaan) dengan tujuan menyampaikan, menerima dan memahami sebuah pesan dengan baik.

C. 5W komunikasi

- who ==> dari siapa
- (says) what ==> apa pesannya
- (in which) channel ==> dari kanal mana
- (to) whom ==> kepada siapa
- (with what) effect ==> apa efeknya

D. 7C komunikasi efektif

- conciseness (keringkasan)
- completeness (kelengkapan)
- consideration (pertimbangan)
- correctness (ketepatan)
- clarity (kejelasan)
- concreteness (konkret)
- courtesy (sopan santun)

E. Sync vs async communication

1. synchronous (komunikasi real time)

komunikasi ini menggunakan alat bantu seperti hp atau pun laptop, karena komunikasi ini dilakukan secara tidak langsung. contoh: pesan whatsapp

2. asynchronous (komunikasi tidak real time)

komunikasi ini dilakukan secara langsung atau bertatap muka satu sama lain, baik itu menggunakan alat bantu seperti hp atau pun laptop. contoh. video call ataupun berkomunikasi tanpa alat bantu contoh: komunikasi tatap muka

BAB 14

FLUTTER LAYOUT

A. Layout

suatu penentuan tata letak desain pada elemen tertentu sehingga menghasilkan visual yang menarik. layout berbentuk widget yang mengatur eidget di dalamnya.

1. single-child layout

a. container

- membuat sebuah box
- membungkus widget lain
- box memiliki margin, padding dan border

- penggunaan container



```
Container(  
  margin: const EdgeInsets.all(10),  
  padding: const EdgeInsets.all(10),  
  decoration: BoxDecoration(  
    border: Border.all(),  
  ),  
  child: const Text('HALO'),  
),  
);
```

b. center

- membuat sebuah box
- membungkus widget lain
- memenuhi lebar dan tinggi ruang di luarnya
- meletakkan widget berada di bagian tengah

c. sizedBox

- membuat sebuah box
- membungkus widget lain
- box dapat diatur lebar dan tingginya
- lebih sederhana dari container

2. multi-child layout

▪ column

- mengatur widgets secara vertical

```
Column(
  children: const [
    Text(teks),
    Text(teks),
    Text(teks),
    Text(teks),
  ],
);
```

- rows
- kebalikan dari column yaitu mengatur widgets secara horizontal
- ListView
- mengatur widgets dalam bentuk list
- memiliki kemampuan scroll

```
ListView(
  children: [
    Container(
      margin: const EdgeInsets.all(10),
      decoration: BoxDecoration(border: Border.all()),
      child: const Text('H'),
    ),
  ],
);
```

- GridView
- mengatur widgets dalam bentuk gallery

```
GridView.count(
  crossAxisCount: angka,
  children: [],
);
```

BAB 15

FLUTTER FORM

A. Form

- menerima isian data dari pengguna
- isian dapat lebih dari satu
- menggunakan statefulwidget
- keadaan form disimpan menggunakan globalkey

B. Input

1. textfield

- menerima isian data dari pengguna
- isian dapat lebih dari satu
- data diambil menggunakan text editing controller
- isian data berupa text

2. radio

- memberi opsi pada pengguna
- hanya dapat memilih satu
- data diambil menggunakan property dengan tipe data sesuai value pada radio

3. checkbox

- memberi opsi pada pengguna
- dapat memilih beberapa opsi
- data diambil menggunakan property dengan tipe bool

4. dropdown button

- memberi opsi pada pengguna
- hanya dapat memilih satu
- opsi tidak ditampilkan di awal, hanya tampil jika di tekan
- data diambil menggunakan property dengan tipe data sesuai value pada drop down menu item.

C. Button

- bersifat seperti tombol
- dapat melakukan sesuatu saat ditekan

1. elevated button

- tombol yang timbul
- jika di tekan, akan menjalankan on proses contoh : submit.

2. icon button

- tombol yang hanya menampilkan icon
- jika di tekan, akan menjalankan on proses contoh : icon home dll.

BAB 16

ADVANCE FORM INPUT

Interactive widgets merupakan interaksi antara aplikasi dan user, kemudian aplikasi mengumpulkan input dan feedback dari user

A. Date picker

- widget dimana user bisa memasukkan tanggal
- tanggal kelahiran, waktu pemesanan tiket, waktu reservasi restoran dan lain-lain
- menggunakan fungsi bawaan flutter `showDatePicker`
- fungsi `showDatePicker` memiliki tipe data `Date`
- menampilkan dialog material design date picker

B. Color picker

- widget dimana user bisa memilih color
- penggunaan color picker ini bisa digunakan untuk berbagai macam kondisi

membuat color picker yaitu dengan cara menambahkan packages terlebih dahulu, packages tersebut bisa di tambahkan dengan banyak cara salah satunya dengan flutter CLI

C. File picker

- kemampuan widgets untuk mengakses storage
- memilih dan membuka file

membuat file picker yaitu dengan cara menambahkan packages terlebih dahulu, packages tersebut bisa di tambahkan dengan banyak cara salah satunya dengan flutter CLI

BAB 17

ASSETS

A. Assets

Deskripsi

- file yang di bundled dan di deployed bersamaan dengan aplikasi
- tipe-tipe assets, seperti : static data (JSON files), icons, images, dan font file (ttf)

Menentukan assets

- flutter menggunakan pubspec.yaml
- pubspec.yaml terletak pada root project, untuk mengidentifikasi assets yang dibutuhkan aplikasi
- gunakan karakter "/" untuk memasukkan semua assets

B. Image

Deskripsi

- image atau gambar akan membuat tampilan aplikasi menjadi lebih menarik
- flutter mendukung format gambar seperti JPEG < WebP, GIF, Animated Web/GIF, PNG, BMP, dan WBMP
- menampilkan gambar dari project assets dan internet
- membuat image di flutter
- menggunakan widget image
- membutuhkan property image dengan nilai class AssetImage()
- menggunakan metode image.asset, untuk mendapatkan image yang sudah ditambahkan dalam project
- menggunakan data image melalui internet dengan menggunakan string url nya

C. Font

Deskripsi

- penggunaan font dengan style tertentu akan menjadi kebutuhan keunikan pada sebuah aplikasi
- penentuan font yang mau dipakai biasanya oleh UI desainer
- penerapan font menggunakan custom font atau dari package

Custom font

cara menggunakan custom font:

- step 1: cari dan download font (Ex, <https://font.google.com/>)
- step 2: import file .ttf
- step 3: daftarkan font di pubspec.yaml
- step 4: mengatur font sebagai default
- step 5: gunakan font di spesifik widget

Font dari packages

cara menggunakan font dari packages:

- step 1: tambahkan package `google_font` di dependencies
- step 2: import package di file dart
- step 3: gunakan font dengan memanggil `GoogleFont.namaFont()`

BAB 18

DIALOG BOTTOM SHEET

A. Aplikasi task management

Deskripsi

- menambahkan dan menghapus kegiatan

B. Alert dialog

Deskripsi

- untuk tampilan android , dari material design
- menginformasikan pengguna tentang situasi tertentu
- bisa digunakan untuk mendapatkan input dari user
- membutuhkan helper method `showDialog`

C. Bottom sheet

Deskripsi

- seperti dialog tetapi muncul dari bawah layar aplikasi
- menggunakan fungsi bawaan flutter `showModalBottomSheet`
- membutuhkan dua property yaitu `context` dan `builder`

BAB 19

FLUTTER NAVIGATION

A. Apa itu navigation?

yaitu untuk berpindah dari halaman satu ke halaman yang lainnya

B. Navigation dasar

Deskripsi

- `navigator.push()`=> digunakan untuk menggunakan perpindahan ke halaman selanjutnya
- `navigator.pop()`=> digunakan untuk menggunakan perpindahan ke halaman sebelumnya
- untuk mengirim data ke halaman baru flutter navigator biasanya menggunakan parameter pada constructor halaman

C. Navigation dengan named routes

Deskripsi

- tiap halaman memiliki alamat yang disebut dengan route. untuk perpindahan ke halaman selanjutnya menggunakan `Naviagator.pushNamed()` -sedangkan untuk kembali ke halaman sebelumnya menggunakan `Navigator.pop()`
- mendaftarkan route
- tambahkan `initialRoute` dan `routes` pada `materialapp`
- tiap route adalah fungsi yang membentuk halaman

BAB 20

FLUTTER GLOBAL STATE MANAGEMENT

A. State

Deskripsi

- data yang dapat dibaca saat pembuatan widget
- dapat berubah saat widget sedang aktif
- hanya dimiliki oleh `StatefulWidget`

kenapa perlu global state?

yaitu karena supaya antar widget dapat memanfaatkan state yang sama dengan mudah

manfaat state

- dapat dibuat sebagai property dari class
- digunakan pada widget saat build

B. Global state

yaitu state yang bisa digunakan pada seluruh widget

C. Provider

yaitu sebuah state management yang penggunaannya perlu di install agar dapat digunakan

- instalasi provider
 - 1) menambahkan `package provider` pada bagian `dependencies` dalam file `pubspec.yaml`
 - 2) jalankan perintah `flutter pub get`
 - 3) membuat `package state provider`
 - 4) buat file bernama `contact.dart`
 - 5) definisikan state dalam bentuk class
 - 6) mendaftarkan state provider
 - 7) import dalam file `main.dart`
 - 8) daftarkan pada `runApp` dengan `MultiProvider`
 - 9) menggunakan state dari provider
 - 10) simpan provider dalam sebuah variabel
 - 11) ambil data dari provider melalui getter

BAB 21

FLUTTER GLOBAL STATE MANAGEMENT (BLoC)

A. Declarative UI

yaitu flutter yang memiliki sifat declarative yang artinya flutter membangun UI nya pada screen untuk mencerminkan keadaan state saat ini

1. state

- state adalah ketika sebuah widget sedang aktif dan widget tersebut menyimpan data di memori
- flutter akan membangun ulang UI nya ketika ada state atau data yang berubah
- ada dua jenis state dalam flutter, ephemeral state dan app state

2. ephemeral state

- digunakan ketika tidak ada bagian lain pada widget tree yang membutuhkan untuk mengakses data widget nya, contoh: PageView, BottomNavigationBar, Switch Button
- tidak perlu state management yang kompleks
- hanya membutuhkan StatefulWidget dengan menggunakan fungsi setState()

3. App state

- digunakan ketika di bagian yang lain suatu aplikasi membutuhkan akses ke data state widget, contoh: -login info -pengaturan preferensi pengguna -keranjang belanja, dimana informasi yang pengguna pilih di suatu screen yang mana informasi itu akan muncul di screen yang berbeda

4. pendekatan state management

- setState: lebih cocok penggunaannya pada ephemeral state
- provider: penggunaan untuk state management yang kompleks seperti app store, pendekatan ini direkomendasikan oleh tim flutter karena mudah dipelajari
- BLoC: menggunakan pada stream/observable, untuk memisahkan UI dengan basis logiknya

B. SetState

setState fungsinya untuk memberitahu framework bahwa ada object yang berubah pada State, kemudian akan melakukan build ulang pada Widget tersebut.

C. Provider

class yang perlu diperhatikan dalam penggunaan provider yaitu:

- dari package provider
- changenotifierprovider

- multiprovider
- consumer
- built in class dari flutter SDK
- changeNotifier

1. changeNotifier

- class yang menambah dan menghapus listeners
- digunakan dengan cara meng-extends
- memanggil notifyListener(), fungsi yang membantu widget yang menggunakan state bahwa terjadinya perubahan data sehingga UI nya harus dibangun ulang

2. changeNotifierProvider

- widget yang membungkus sebuah class
- mengimplementasikan changeNotifier dan menggunakan child untuk widget UI nya
- menggunakan create, provider yang akan menyimpan model yang dibuat

3. consumer

- widget yang mendengarkan perubahan kelas yang mengimplementasikan changeNotifier
- membangun ulang widget yang dibungkus consumer saja
- penting untuk meletakkan consumer pada lingkup sekecil mungkin
- membutuhkan property builder yang berisi context, model dan child

D. BLoC

1. deskripsi

- BLoC (Business Logic Component)
- memisahkan antar business logic dengan UI

2. cara kerja BLoC

- menerima event sebagai input
- dianalisa dan dikelola di dalam BLoC
- menghasilkan state sebagai output

3. stream

- rangkaian proses secara asynchronous
- actor utama di dalam BLoC

BAB 22

FLUTTER ANIMATION

A. Animation

- animasi untuk membuat aplikasi terlihat hidup
- widget yang bergerak menambah gaya tarik
- pergerakan dari kondisi A ke kondisi B

B. Implicit animation

- versi animasi dari widget yang sudah ada
- jika tanpa animasi, maka perubahan container tampak kaku jika dengan animasi perubahan container menjadi lebih hidup dan nyaman dilihat

C. Transition

- yaitu animasi yang terjadi saat perpindahan halaman
- dilakukan pada bagian `navigator.push()`;

D. Transition umum

1. fade transition

yaitu halaman yang muncul dengan efek redup menuju ke tampak jelas secara penuh

2. scale transition

yaitu halaman yang muncul dengan efek berubah ukuran dari tidak penuh menuju ke tampilan keseluruhan

BAB 23

STORAGE

A. Penyimpanan lokal

- diperlukan untuk efisiensi penggunaan data internet
- ada beberapa cara implementasi penyimpanan lokal, contohnya: shared preferences, local database

B. Shared preference

- menyimpan data yang sederhana
- penyimpanan dengan format key-value
- menyimpan tipe data dasar seperti teks, angka, dan boolean
- contoh penggunaan shared preference seperti penyimpanan data login dan menyimpan riwayat pencarian

C. Local database

- menyimpan dan meminta data dalam jumlah besar pada local device
- bersifat private
- menggunakan SQLite database melalui package sqflite
- SQLite adalah database open source yang mendukung insert, read, update dan remove data

D. Aplikasi task management (continue)

- yaitu digunakan untuk menambahkan data login

BAB 24

MEMBUAT CV ATAU RESUME YANG EFEKTIF

A. Beberapa hal yang harus diingat:

- kesalahan umum
- salah ketik (typo)
- tidak terstruktur
- sangat klise dan banyak buzzerword
- tidak ada achievement
- terlalu panjang
- CV umum atau tidak spesifik

B. Konten dalam CV

- nyambung/relavan
- informasi apapun yang kamu tulis di CV harus nyambung dengan pekerjaan yang kamu apply
- ada angka lebih baik
- tindakan lebih penting dari pada kata-kata
- hindari buzzerword
- jangan menggunakan kata-kata yang seolah menyombongkan diri
- singkat
- bukan harus pendek, tapi singkat
- perhatikan kata kunci
- cocokkan CV dengan keyword yang ada di job description
- pakai font yang tepat
- jangan pakai comic sans atau wingdings
- baca ulang
- rekruter melihat ketelitianmu dan salah ketik

BAB 25

REST API CLIENT

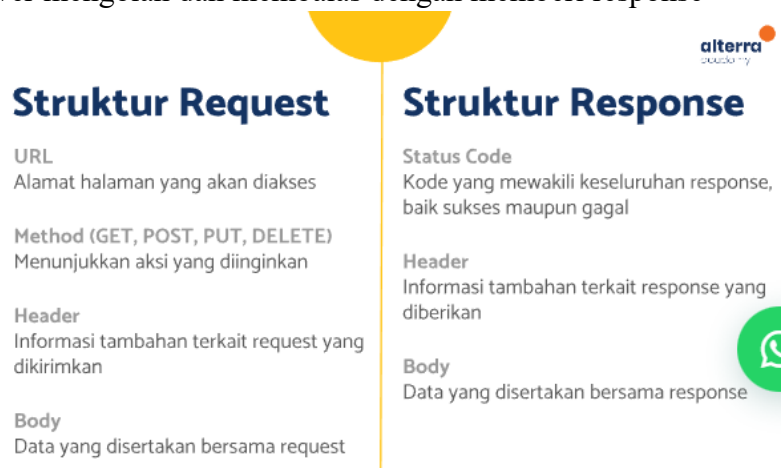
A. RESt API

Representational state transfer application programming interface
deskripsi

- arsitektural yang memisahkan tampilan dengan proses bisnis
- bagian tampilan dengan proses bisnis berkirim data melalui HTTP request

B. HTTP

- protokol yang digunakan untuk berkirim data pada internet, biasanya data tersebut berbentuk media web.
- pola komunikasi
- client mengirim request
- server mengolah dan membalas dengan memberi response



C. Dio

- sebagai HTTP client
- dimanfaatkan untuk melakukan REST API
- instalasi
- tambahkan dependency pada pubspec.yaml
- jalankan perintah flutter pub get pada terminal

D. serialisasi JSON

- javascript object notation
- cara penulisan data
- umum digunakan pada REST API
- serialisasi JSON
- mengubah struktur data ke bentuk JSON
- deserialisasi JSON
- mengubah bentuk JSON ke struktur data

BAB 26

MVVM ARCHITECTURE

A. MVVM architecture (model-view viewModel)

yaitu memisahkan logic dengan tampilan (view) ke dalam viewModel

B. Keuntungan

1. Reusability

jika ada beberapa tampilan yang memerlukan alur logic yang sama, mereka bisa menggunakan viewModel yang sama

2. Maintainability

mudah dirawat karena pekerjaan terkait tampilan tidak tertumpuk bersama logic

3. Testability

pengujian menjadi terpisah antara pengujian tampilan dengan pengujian logic sehingga dapat meningkatkan produktivitas pada pengujian

C. Melakukan MVVM

1. struktur directori

model memiliki 2 bagian, yaitu:

- bentuk data yang akan digunakan dan
- sumber dari data tersebut

2. model

- bentuk data yang akan digunakan, dibuat dalam bentuk class
- data-data yang dimuat, diletakkan pada property

3. viewModel

- menyimpan data-data dan logic diperlukan halaman ContactScreen
- jika widget lain memerlukan logic yang sama, dapat menggunakan viewModel ini juga

BAB 27

FINITE STATE MACHINE

A. Finite state machine

1. deskripsi
 - mesin yang memiliki sejumlah state
 - tiap state menunjukkan apa yang terjadi sebelumnya
2. contoh

terdapat 3 state yang menunjukkan kejadian dalam suatu proses

 - IDLE saat tidak terjadi proses
 - RUNNING saat proses sedang berjalan
 - ERROR saat proses gagal diselesaikan
3. contoh sukses
 - IDLE berubah menjadi RUNNING saat proses berjalan
 - RUNNING kembali menjadi IDLE saat proses selesai
4. contoh gagal
 - IDLE berubah menjadi RUNNING saat proses berjalan
 - RUNNING menjadi ERROR saat terjadi kegagalan
 - ERROR kembali menjadi IDLE

B. Implementasi pada widget

- perbedaan tampilan setiap state
- informasi finite state dapat berguna untuk menentukan seperti apa tampilan yang dimunculkan

BAB 28

UNIT TEST

A. Unit test

1. deskripsi

- salah satu jenis pengujian pada perangkat lunak
- pengujian dilakukan pada unit dalam perangkat lunak
- unit yang dimaksud umumnya adalah fungsi atau method

2. tujuan

- memastikan fungsi dapat mengolah berbagai jenis input
- memastikan hasil dari suatu fungsi atau method sudah sesuai
- menjadi dokumentasi

3. cara kerja

a. manual

unit dijalankan dan dicek hasilnya secara manual

b. automated

menulis script yang dapat dijalankan berkali-kali menggunakan test runner

B. Test group

- beberapa test dengan subjek yang sama diletakkan dalam sebuah kelompok
- dilakukan dengan menggunakan group

C. mocking

1. deskripsi

- proses yang terikat dengan sistem eksternal dapat mempengaruhi jalannya pengujian
- pengaruh tersebut dapat mengganggu proses pengujian
- dilakukan mocking untuk menghindari masalah-masalah tersebut

2. cara kerja

- suatu object dibuat bentuk tiruannya
- bentuk tiruan tersebut memiliki input dan output yang sama dengan yang asli
- bentuk tiruan menghilangkan ketergantungan pada sistem eksternal

BAB 29

UI TESTING

A. UI testing

yaitu pengujian yang dilakukan pada tampilan suatu aplikasi untuk memastikan aplikasi dapat menerima interaksi dan memberi respon kepada pengguna

1. UI testing flutter

disebut widget testing sehingga pengujian dilakukan pada widgets dengan menuliskan script yang dapat dijalankan secara otomatis

2. keuntungan

- memastikan seluruh widget memberi tampilan yang sesuai
- memastikan seluruh interaksi dapat diterima dengan baik
- menjadi ketentuan yang harus diikuti saat diperlukan perubahan pada widget

B. Melakukan UI testing

1. instalasai package testing

umumnya package testing sudah terinstal sejak project pertama kali dibuat, tetapi, cara ini dapat dilakukan jika ingin melakukan secara manual

2. penulisan script testing

- dilakukan pada folder test
- nama file harus diikuti `_test.dart`
- setiap file berisi fungsi `main()` yang didalamnya dapat dituliskan script testing
- setiap skenario pengujian disebut test case

3. script testing

- test case diawali dengan `"testWidgets"` dan diberi judul
- simulasi proses mengaktifkan halaman `AboutScreen`
- memeriksa apakah halaman tersebut terdapat teks `"About Screens"`
- menjalankan testing
- perintah flutter test akan menjalankan seluruh file test yang dibuat
- hasil dari perintah ini adalah lamanya waktu pengujian, total pengujian yang dilakukan, dan pesan bahwa pengujian berhasil.

BAB 30

OUTPUT BUILD FLUTTER

A. Apa itu output build

- file/berkas hasil keluaran dari project flutter
- digunakan untuk merilis aplikasi ke app store
- terdiri dari format APK atau AAB untuk android dan IPA untuk IOS
 - mode build
 - mode debug
 - mode profile
 - mode release

B. Build APK

1. deskripsi

- hanya dapat berjalan di android
- memiliki format .apk
- dapat di install langsung pada perangkat android
- proses sebelum APK
- mengatur nama aplikasi di AndroidManifest.xml
- mengatur ikon aplikasi
- mengatur perizinan aplikasi di AndroidManifest.xml

2. persiapan

- buat project bary beri nama counter_app
- AndroidManifest.xml
- file yang berisi informasi penting tentang aplikasi android, seperti nama, komponen, permission, dan fitur hardware aplikasi
- terletak di 3 folder untuk setiap mode build debug, release/main dan profile

C. Mengatur nama aplikasi

1. langkah-langkah

a. step 1

- ubah property android:label pada AndroidManifest.xml yang ada di folder **android/app/src/main/AndroidManifest.xml**

b. step 2

- stop dan jalankan aplikasi

D. Mengatur ikon aplikasi

1. langkah-langkah

- step 1: men-generate ikon menggunakan website
- step 2: download zip file hasil generate
- step 3: unzip hasil download
- step 4: copy seluruh folder mipmap ke project flutter
- step 5: stop dan jalankan ulang aplikasi

E. Build APK

1. deskripsi

- hasil output hanya dapat berjalan di IOS
- memiliki format.ipa
- perlu mendaftar aku ke **Apple Developer Program**
- perintah build hanya dapat berjalan atau dijalankan pada perangkat apple

2. info.plist

- tempat konfigurasi data informasi mengenai aplikasi IOS
- berada di folder io/Runner/info.plist
- untuk mengatur nama aplikasi, ubah nama di dalam tag string dibawah key bundle name yang ada di info.plist

F. Bonus

- package rename
- mengubah nama aplikasi menggunakan package **rename**
- persiapan
- buat project baru beri nama **bonus**

langkah-langkah

- step 1: install package **rename**
- step 2: jalankan perintah pada terminal
- step 3: stop dan jalankan ulang aplikasi