

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Dalam pelaksanaan kerja magang di PT. Intermedia Multibahasa Indonesia, posisi *Flutter Developer Support* berada di bawah divisi Product dengan supervisi dari Arnes Respati selaku *Chief of Technology* PT. Intermedia Multibahasa Indonesia. Dalam melakukan proses kerja magang, Arnes Respati berperan dalam mengintegrasikan fitur-fitur yang sudah dibuat.

Selain melakukan koordinasi dengan Arnes Respati, Natasya Elora sebagai *Product Manager* bertindak menjembatani dengan *UI/UX Designer*. *Project Manager* yakni Marcell Septian Hardja membantu dalam hal integrasi aplikasi dengan database NodeJS.

3.2 Tugas yang Dilakukan

Selama pelaksanaan praktek kerja magang, tugas yang dilakukan adalah menerapkan desain yang dirancang oleh UI/UX menggunakan Figma dan juga menerapkan *Back-End* dari aplikasi agar terhubung dengan database NodeJS.

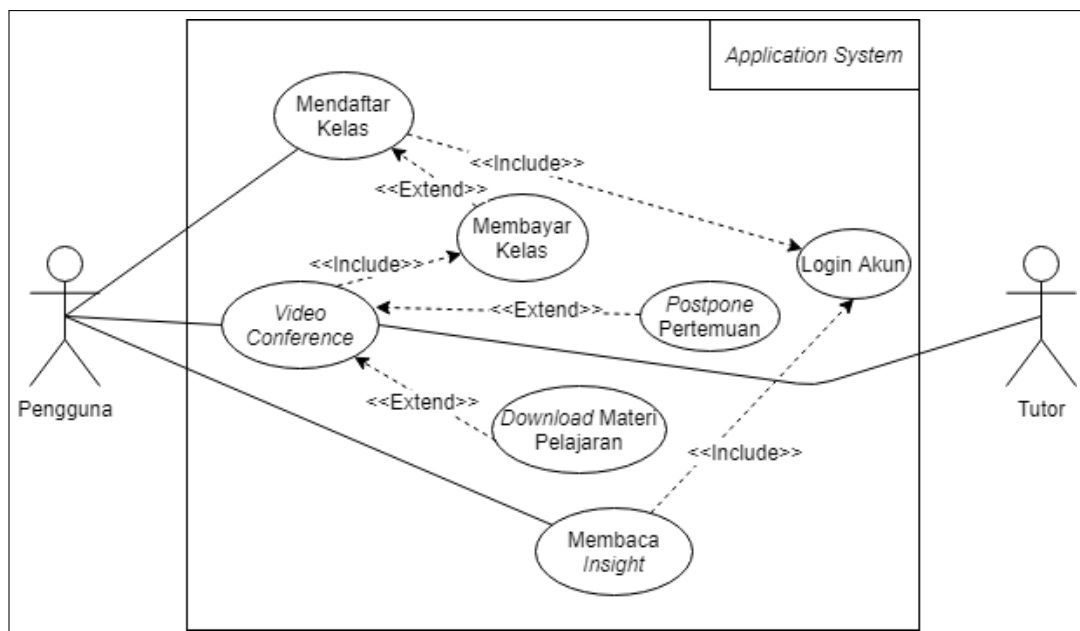
3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Memahami kode yang ada dan <i>refactoring</i> kode-kode yang berulang
2	Menerapkan desain UI/UX pada halaman <i>postponed</i>
3	Menerapkan desain UI/UX untuk <i>tooltip</i> dengan memanfaatkan <i>feature_discovery</i>
4	Melakukan <i>merge branch</i> dari beberapa fitur ke dalam 1 branch dan membuat <i>pop up</i> notifikasi dialog
5	Membuat halaman baru untuk notifikasi
6	Membuat halaman baru untuk <i>payment system</i>
7	Memperbaiki halaman <i>my courses</i>
8	Melakukan <i>Back-End Integration</i> terhadap aplikasi

Diagram *Use Case Diagram* dari aplikasi secara garis besar dapat dilihat pada Gambar 3.1



Gambar 3.1. Diagram *Use Case Diagram* dari aplikasi secara garis besar

3.3.1 Halaman Awal

Pada minggu pertama, dimulai dengan *onboarding* pengenalan rekan kerja, *tools* yang digunakan, serta metode komunikasi yang digunakan dalam lingkup kerja Lingotalk. *Onboarding* dibarengi dengan *handover* dari pendahulu sebelumnya yakni Elfana yang telah mengerjakan sekitar 80% dari *Front-End*.

Rekan kerja Natasya Elora sebagai *Product Manager* akan mempersiapkan *Product Requirement Document* (PRD) dan Kanban Board yang terdapat pada Jira Software yang digunakan untuk *tracking* progress dan waktu dibutuhkan selama pengerjaan tugas. Pembimbing lapangan Arnes Respati sebagai *Chief Technology Officer* akan melakukan *tracking* serta menjadi *reviewer* terhadap semua kode yang sudah di-*commit*. *Tools* yang digunakan yakni Jira Software sebagai Kanban Board yang akan mencatat tugas dan deadline dari tugas yang harus dikerjakan serta Git-Lab sebagai tempat penyimpanan kode yang sudah dikerjakan. Untuk metode komunikasi, dilakukan secara *video-call* melalui Microsoft Teams tiap minggu berdasarkan jadwal yang sudah ditetapkan.

Setelah *onboarding* selesai, dilakukan pengecekan kode terhadap pendahulu dan ditemukan kode yang berulang yang bertentangan dengan prinsip SOLID pada poin utama yakni *Single Responsibility* (Nesteruk, 2020). Oleh karena itu, dilakukan *refactoring* kode-kode yang berulang sehingga hanya bersumber pada satu class dan pada kode lainnya cukup memanggil class itu. Contoh kode sebelum dan sesudah *refactor* dapat dilihat di gambar 3.2.

```

Widget _getStartedButton(double givenHeight) {
  return RaisedButton(
    shape: StadiumBorder(),
    color: Color(0xFFF9663A),
    onPressed: () => Get.to(() => GetStartedRegister()),
    child: Text(
      'Get Started',
      style: GoogleFonts.poppins(
        fontSize: 14.0.sp,
        fontWeight: FontWeight.w500,
        color: Colors.white,
      ),
    ), // Text
  ); // RaisedButton
}

```

(a) Kode Widget `getStartedButton` sebelum *Refactor*

```

Widget _getStartedButton(double givenHeight) {
  return CustomButtonTextYellow(
    pressFunction: () => Get.to(() => GetStartedRegister()),
    text: 'Get Started',
    height: givenHeight,
    width: 80.0.w,
    roundedBorder: true,
  ); // CustomButtonTextYellow
}

```

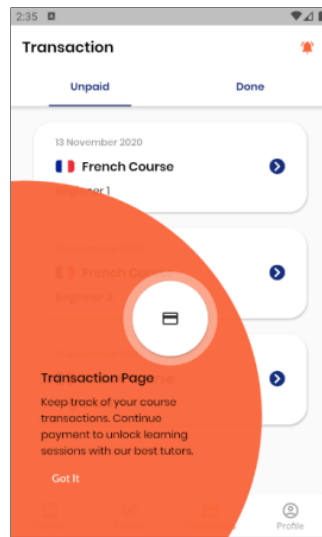
(b) Kode Widget `getStartedButton` setelah *Refactor*

Gambar 3.2. Perbandingan Kode Widget `getStartedButton` sebelum dan setelah *Refactor*

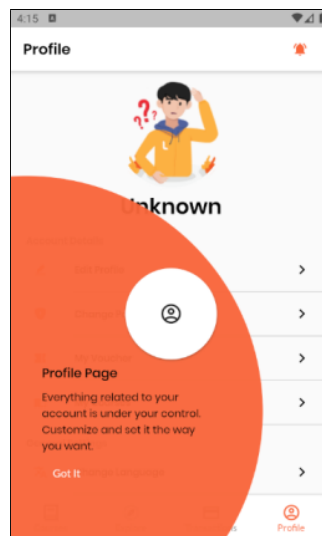
3.3.2 Halaman Utama

Pada aplikasi yang baru digunakan pertama kali, pengguna seringkali tidak mengetahui fungsi dari halaman yang ia buka. Untuk memberi kejelasan kepada pengguna akan fungsi dari halaman yang ia buka, maka dibuatlah tampilan *tooltip* dengan memanfaatkan *package* `feature_discovery`. `Feature_discovery` merupakan salah satu *Material Guidelines* yang ada di *Material Design* (Google, 2021b) yang merupakan desain standar yang diperkenalkan oleh Google pada tahun 2014.

Implementasi desain dapat dilihat di Gambar 3.3 dan 3.4.



Gambar 3.3. Implementasi *Feature Discovery* di halaman *Transaction*



Gambar 3.4. Implementasi *Feature Discovery* di halaman *Profile*

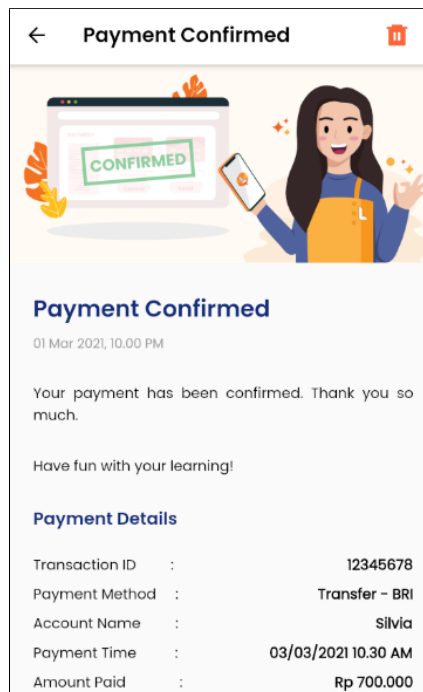
Pada gambar 3.4, terdapat ikon bel pada atas kanan aplikasi yang akan membuka halaman notifikasi. Notifikasi terdiri dari tiga jenis yakni:

- Notifikasi aktif yang akan membuka halaman *New Assignment*
- Notifikasi aktif yang akan membuka halaman *Payment Confirmed*
- Notifikasi tidak aktif yang akan membuka halaman *Assignment Feedback*

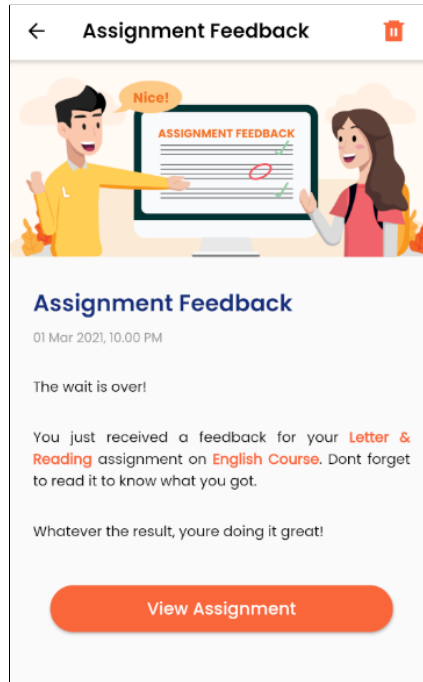
Hasil implementasi desain dapat dilihat pada Gambar 3.5 sampai 3.7.



Gambar 3.5. Implementasi halaman *New Assignment*



Gambar 3.6. Implementasi halaman *Payment Confirmed*



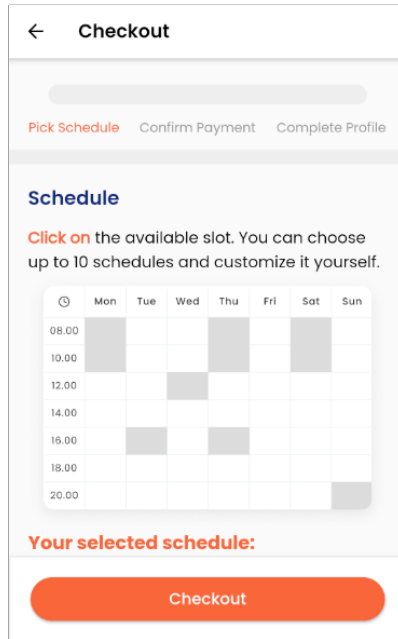
Gambar 3.7. Implementasi halaman *Assignment Feedback*

3.3.3 Halaman Course

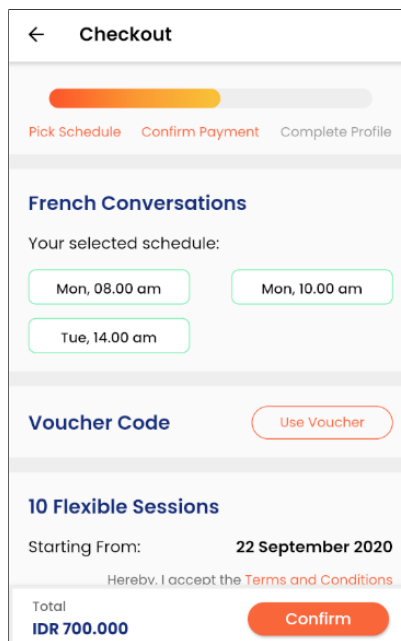
Di halaman *course*, pengguna dapat mendaftar *course*. Halaman tersebut bernama halaman *checkout*. Halaman ini terdiri dari tiga bagian yakni:

- Halaman *Pick Schedule*
- Halaman *Confirm Payment*
- Halaman *Complete Profile*

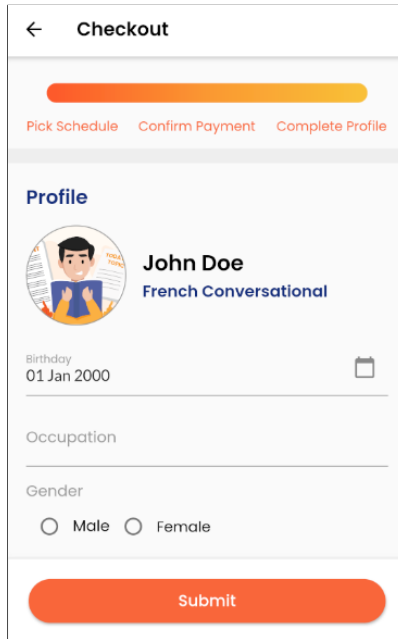
Hasil implementasi desain dapat dilihat pada Gambar 3.8 sampai 3.10.



Gambar 3.8. Halaman *checkout* untuk *Pick Schedule*

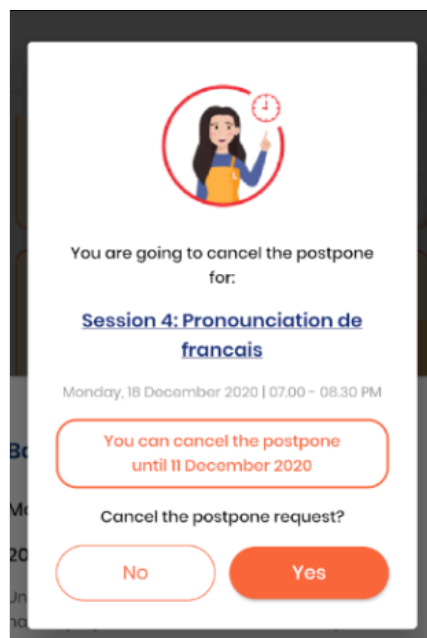


Gambar 3.9. Halaman *checkout* untuk *Confirm Payment*



Gambar 3.10. Halaman *checkout* untuk *Complete Profile*

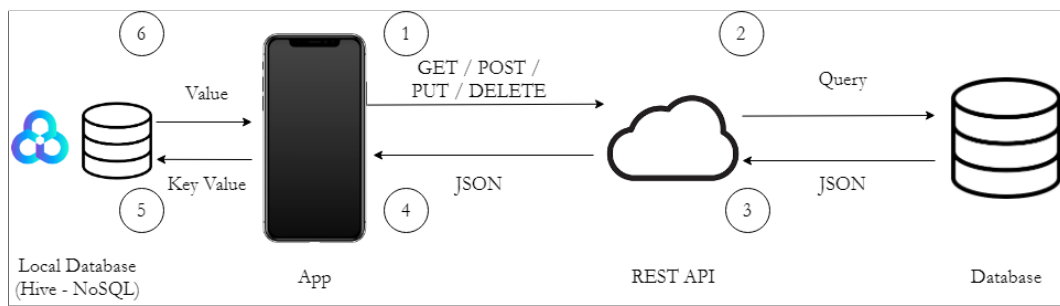
Di halaman ini, terdapat mode untuk membatalkan (*postpone*) jadwal kelas privat yang sudah pernah dibuat sebelumnya. Implementasi desain dapat dilihat di Gambar 3.11.



Gambar 3.11. Implementasi *postpone* pada desain

3.3.4 Halaman Explore

Pada halaman ini, terdapat *insight* yang akan diambil dari database NodeJS. Untuk mencapai tujuan tersebut, maka digunakan *package* http, get, dan hive. *Package* http berfungsi untuk mengoneksikan aplikasi dengan API. *Package* get berfungsi sebagai *state management* yang berfungsi untuk mengatur tampilan aplikasi sesuai keadaan/ variabel yang ada di kelas yang didefinisikan. *Package* hive sebagai database NoSQL yang berfungsi untuk menyimpan data yang didapat dari API. Proses pengambilan data dengan model REST API terjadi seperti Gambar 3.12.



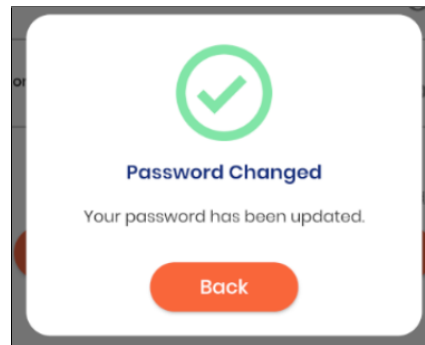
Gambar 3.12. Simulasi pengambilan data dengan REST API

Tahap-tahap dari pengambilan data:

1. Aplikasi meminta *request* data dengan metode *GET/ POST / PUT / DELETE* ke *endpoint* REST API
2. REST API akan melakukan *query* ke database
3. Database merespon dengan memberikan data dengan format JSON
4. REST API meneruskan JSON ke aplikasi
5. Data yang diterima disimpan pada penyimpanan lokal yakni dengan Hive yang merupakan database NoSQL dengan format *Key Value*
6. Database Hive akan memberikan data berupa *Value* sehingga meminimalisir koneksi ke REST API.

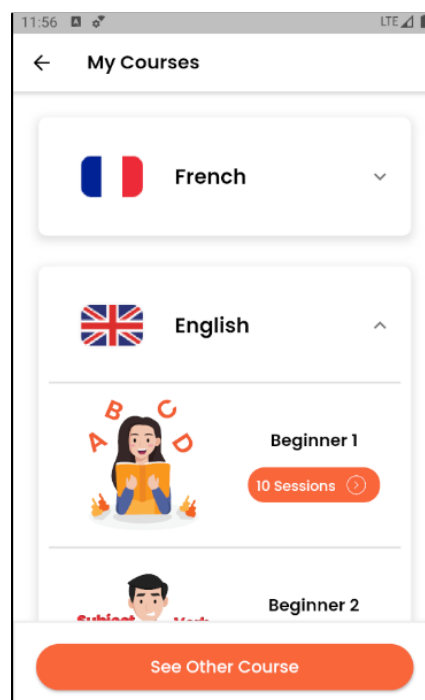
3.3.5 Halaman Profile

Di halaman profile, terdapat menu untuk mengganti password. Saat password selesai diganti, maka akan muncul *pop up* notifikasi dialog yang menyatakan password berhasil diubah. Implementasi desain dapat dilihat di Gambar 3.13.



Gambar 3.13. Implementasi *pop up* notifikasi dialog saat mengganti *password*

Di halaman profile, juga terdapat menu untuk melihat *my courses*. Halaman *my courses* diperbaiki dikarenakan pendahulu tidak membuat tampilan dengan sempurna. Implementasi desain dapat dilihat di Gambar 3.14.

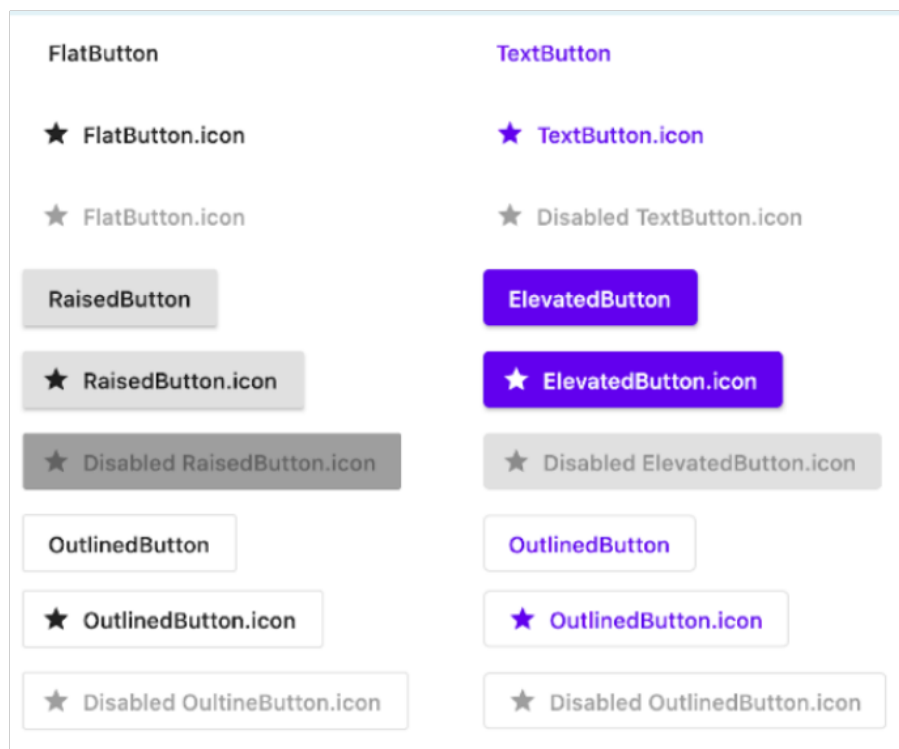


Gambar 3.14. Perbaikan halaman *my courses*

3.4 Kendala dan Solusi yang Ditemukan

Beberapa kendala yang ditemukan selama pelaksanaan praktek kerja magang beserta solusinya:

1. **Kendala:** Adanya API yang sudah usang (*deprecated*) yang disebabkan diluncurkannya Flutter versi 2.0.0 pada 3 Maret 2021 (Google, 2021a). List API yang sudah usang dan penggantinya terdata di Gambar 3.15 (Flutter, 2021).



Gambar 3.15. Tampilan dari API yang sudah usang (*deprecated*) dan penggantinya

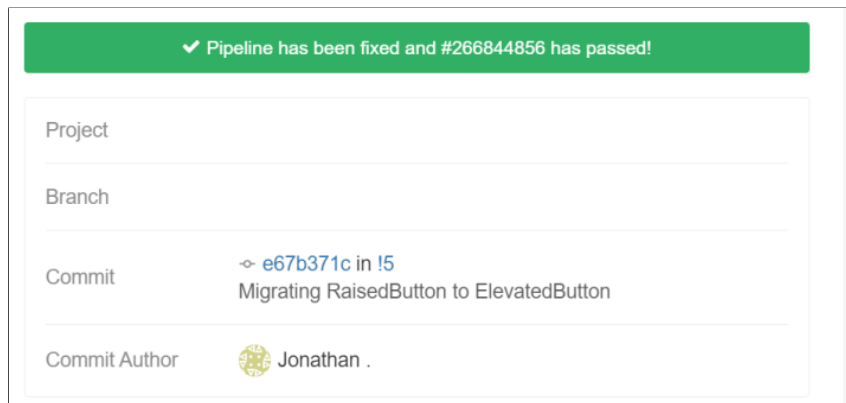
Solusi: Melakukan migrasi API dari yang sudah usang menjadi yang baru agar GitLab yang bertindak sebagai Docker yang melakukan pengecekan terhadap kode menjadi lulus (*passed*). Gambar 3.16 memperlihatkan GitLab gagal dan Gambar 3.17 memperlihatkan GitLab sukses.

```
lintcodebase Lint

info • 'RaisedButton' is deprecated and shouldn't be used. Use ElevatedButton instead. See the migration guide in
flutter.dev/go/material-button-migration-guide). This feature was deprecated after v1.25.0-8.1.pre. •
lib/widgets/custom_button.dart:36:12 • deprecated_member_use
info • 'RaisedButton' is deprecated and shouldn't be used. Use ElevatedButton instead. See the migration guide in
flutter.dev/go/material-button-migration-guide). This feature was deprecated after v1.25.0-8.1.pre. •
lib/widgets/custom_button.dart:63:12 • deprecated_member_use
info • 'RaisedButton' is deprecated and shouldn't be used. Use ElevatedButton instead. See the migration guide in
flutter.dev/go/material-button-migration-guide). This feature was deprecated after v1.25.0-8.1.pre. •
lib/widgets/lesson_dialog.dart:91:28 • deprecated_member_use
info • 'RaisedButton' is deprecated and shouldn't be used. Use ElevatedButton instead. See the migration guide in
flutter.dev/go/material-button-migration-guide). This feature was deprecated after v1.25.0-8.1.pre. •
lib/widgets/lesson_dialog.dart:108:28 • deprecated_member_use
info • 'RaisedButton' is deprecated and shouldn't be used. Use ElevatedButton instead. See the migration guide in
flutter.dev/go/material-button-migration-guide). This feature was deprecated after v1.25.0-8.1.pre. •
lib/widgets/lesson_dialog.dart:285:26 • deprecated_member_use

15 issues found. (ran in 34.0s)
Cleaning up file based variables
ERROR: Job failed: exit code 1
```

Gambar 3.16. Tampilan dari API yang sudah usang (*deprecated*) yang menyebabkan tidak lulusnya (*failed*) kode yang di-*commit* ke GitLab



Gambar 3.17. Notifikasi dari GitLab yang menandakan error sudah berhasil diperbaiki

2. **Kendala:** Adanya pembaruan *Null Safety* pada Flutter 2.0.0 menyebabkan beberapa *package* sudah menggunakannya sementara aplikasi masih belum dimigrasikan ke *Null Safety* (Thompsen, 2021).

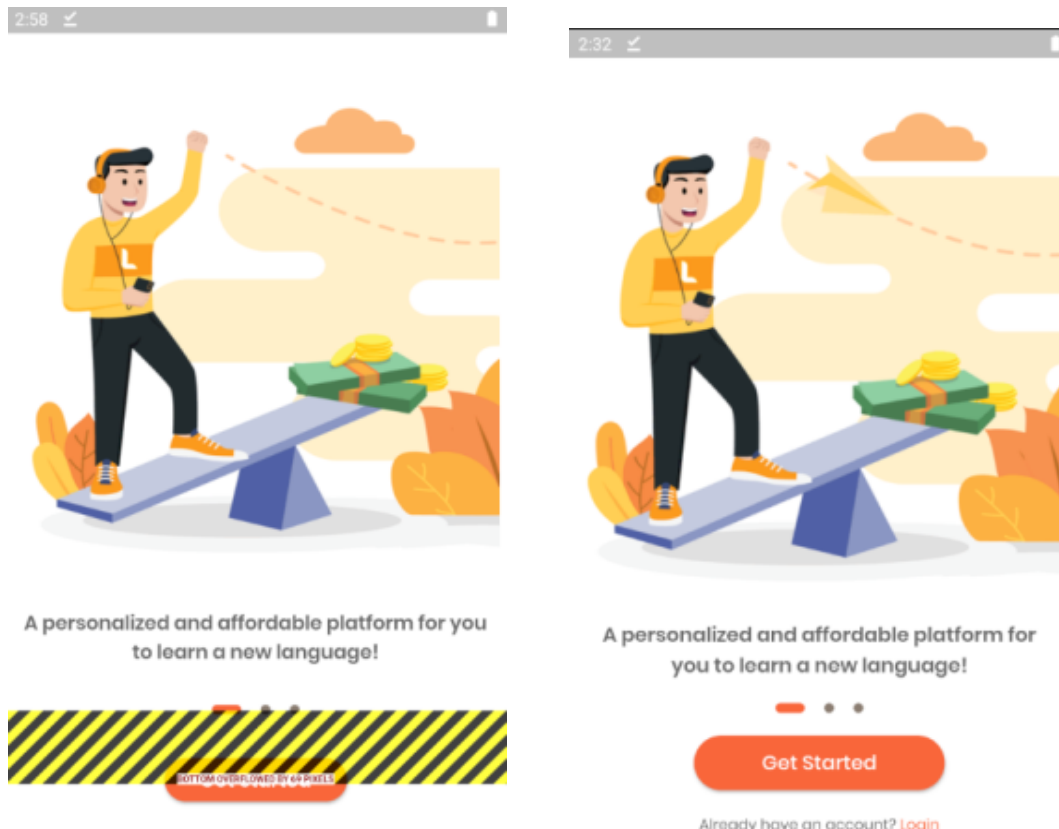
Solusi: Menggunakan versi *package* yang masih belum menggunakan *Null Safety* dengan mengecek pada *versioning package* di dokumentasinya.

3. **Kendala:** Tampilan tidak *responsive*. Tampilan yang tidak *responsive* akan membuat pengguna aplikasi menjadi tidak nyaman menggunakan aplikasi tersebut dan memberikan kesan buruk terhadap aplikasi seperti pada Gambar 3.18a.

Solusi: Memperbaiki *responsiveness* dengan menggunakan *package sizer* yang tersedia di <https://pub.dev/packages/sizer>. Tiga komponen utama yang digunakan pada *package* ini yakni:

- ‘.h’ yang mengambil tinggi gawai
- ‘.w’ yang mengambil lebar gawai
- ‘.sp’ yang mengambil ukuran tulisan gawai

Hasilnya dapat dilihat pada Gambar 3.18.



(a) Tampilan dari aplikasi yang tidak responsif

(b) Tampilan dari aplikasi yang responsif

Gambar 3.18. Perbandingan tampilan dari aplikasi yang tidak responsif dengan responsif

4. **Kendala:** Struktur file yang berantakan dan tidak dikelompokkan dalam fungsionalitasnya membuat kode susah dibaca dan dipelihara seperti pada Gambar 3.19a.

Solusi: Mengatur ulang file yang ada dan memasukkannya ke dalam subfolder-subfolder yang sesuai dengan fungsionalitasnya sehingga lebih rapi dan mudah dalam mencari kodenya seperti pada Gambar 3.19.



(a) Diagram struktur dari *source code* sebelum disusun ulang

(b) Diagram struktur dari *source code* setelah disusun ulang

Gambar 3.19. Perbandingan struktur dari *source code* sebelum dan setelah disusun ulang

5. **Kendala:** Belum adanya desain gambar yang mendeskripsikan koneksi internet tidak ada.

Solusi: Mengomunikasikan untuk meminta bagian UI/UX untuk mendesain gambar yang mendeskripsikan koneksi internet tidak ada.