

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Fitur

Untuk memperoleh data yang dibutuhkan dalam perancangan sistem, maka penulis berdiskusi dengan Team Waktukita.com yang kemudian menghasilkan kebutuhan fitur seperti pada Tabel 3.1.

Tabel 3.1 Kebutuhan Fitur

Fitur	Deskripsi
Register	Pengunjung dapat mendaftar sebaga Talent
Login	Talent (Pengguna terdaftar dan terverifikasi) dapat melakukan login menggunakan aplikasi
Lihat Daftar Pekerjaan	Talent dapat melihat daftar pekerjaan yang ada di sistem
Cari Pekerjaan	Talent dapat mencari pekerjaan berdasarkan nama
Lihat Detail Pekerjaan	Talent dapat melihat detail informasi pekerjaan
Lamar Pekerjaan	Talent dapat apply ke pekerjaan yang tersedia
Lihat Daftar Kelas	Talent dapat melihat daftar kelas yang ada dalam sistem
Lihat Detail Kelas	Talent dapat melihat detail informasi kelas
Cari Kelas	Talent dapat mencari kelas berdasarkan nama
Lihat Detail Pekerjaan Terlarar	Talent dapat melihat detail informasi pekerjaan yang sudah dilamar
Lihat Daftar Pekerjaan Terlarar	Talent dapat melihat daftar pekerjaan yang sudah dilamar
Cari Pekerjaan Terlarar	Talent dapat mencari pekerjaan terlarar berdasarkan nama
Lihat profil akun	Talent dapat mengakses halaman <i>profilnya</i>
Sunting profil	Talent dapat menyunting data profil

3.2 Analisis Pengguna

Pengguna yang dapat menggunakan aplikasi adalah sebagai berikut :

a. Pengunjung

Pengunjung adalah pengguna yang belum terdaftar dalam aplikasi dan tidak dapat melakukan login. Pengunjung hanya dapat melakukan registrasi pada aplikasi.

b. Talent

Talent adalah pengguna yang sudah terdaftar dan terverifikasi dalam aplikasi dan dapat melakukan login akun untuk mengakses semua fitur selain fitur registrasi dan login

3.3 Analisis Kebutuhan Perangkat Lunak

Pada tahapan ini ditentukan perangkat lunak yang akan digunakan dalam membangun dan mengimplementasikan aplikasi, kebutuhan perangkat lunak sebagai berikut :

- a. Sistem Operasi Windows 10/Mac OSX Catalina
- b. Flutter Stable SDK (*Software Development Kit*)
- c. Android SDK
- d. iOS SDK
- e. Visual Studio Code
- f. Postman

3.4 Analisis Kebutuhan Perangkat Keras

Pada tahapan ini ditentukan spesifikasi minimum perangkat keras yang dibutuhkan untuk menjalankan aplikasi, sebagai berikut :

- a. Android Device dengan spesifikasi minimum :

Memori : 1GB RAM

Versi OS : Lollipop 5.1

- b. iOS Device dengan spesifikasi minimum :

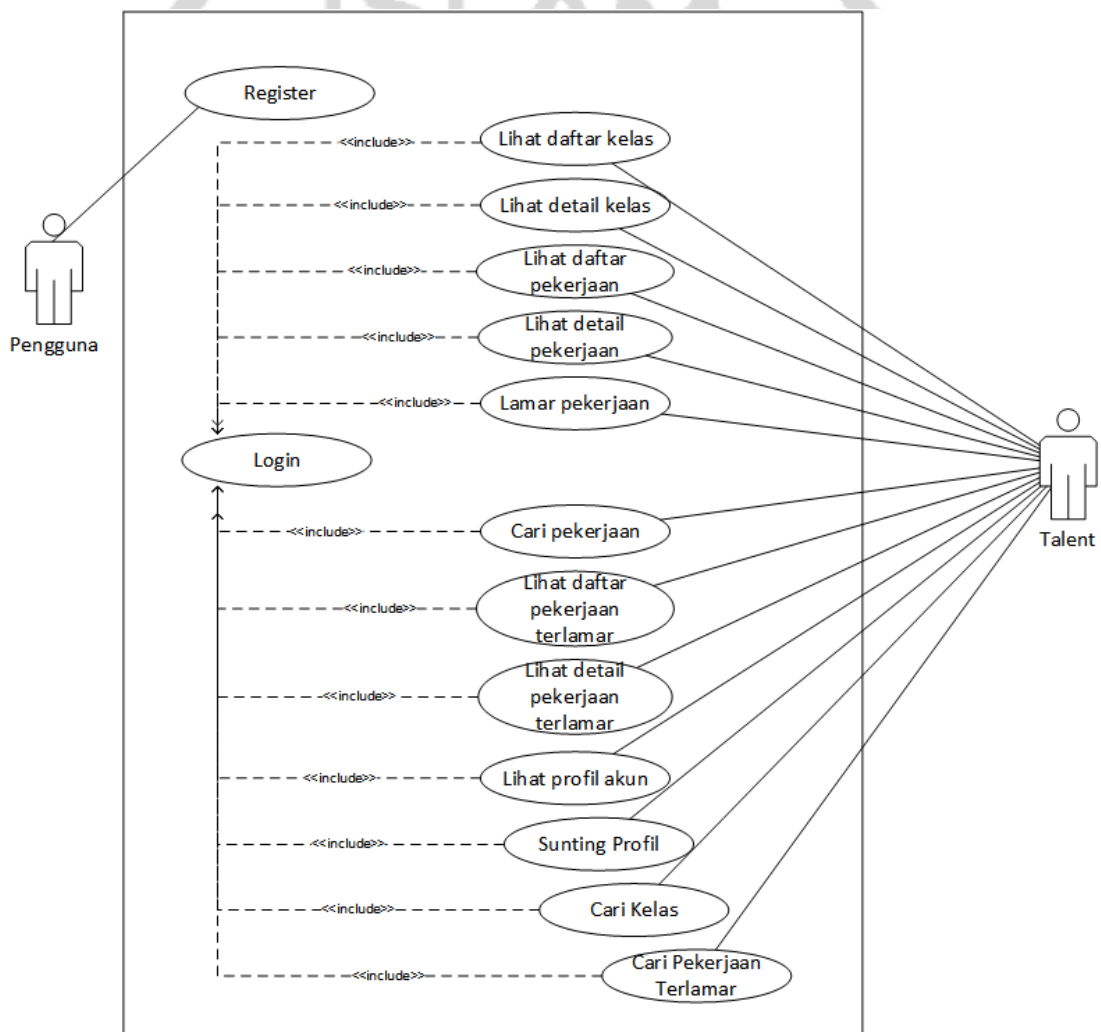
Memori : 512 MB

Versi OS : iOS 12.4.1

3.5 Usecase Diagram

Usecase diagram merupakan diagram untuk menggambarkan interaksi antara aktor dan system. Dari hasil analisa, terdapat dua actor yang berinteraksi dengan aplikasi, yaitu Talent dan Pengunjung.

Pengunjung hanya dapat melakukan registrasi pada aplikasi tanpa bisa mengakses fitur lain. Talent merupakan pengguna yang sudah melakukan registrasi sebelumnya dan terverifikasi dapat melakukan login akun pada aplikasi. Setelah login, talent dapat mencari kelas, melihat daftar kelas dan detail kelas, mencari pekerjaan, melihat daftar pekerjaan detail pekerjaan, mencari pekerjaan yang sudah terlarang, melihat daftar dan detail pekerjaan yang sudah terlarang, melihat profil dan menyuntingnya. Usecase diagram dari aplikasi yang akan dibuat adalah seperti pada Gambar 3.1.



Gambar 3.1 Usecase Diagram

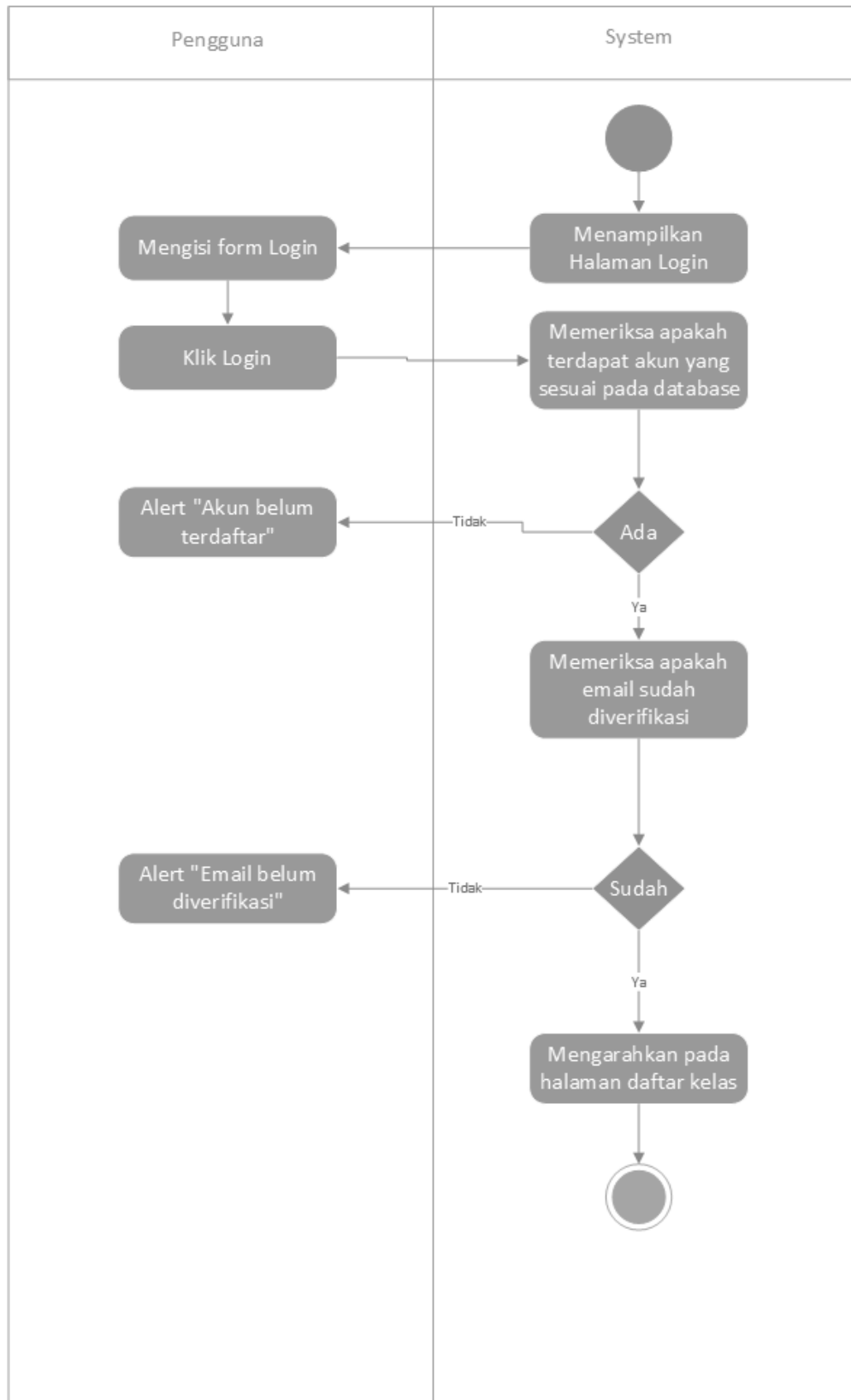
3.6 Activity Diagram

Diagram ini menggambarkan alur kerja urutan aktifitas yang terjadi pada sistem. Diagram ini menunjukkan langkah-langkah proses kerja yang terjadi pada sistem dari awal hingga akhir. Berikut *activity* diagram dari aplikasi *mobile e-recruitment*.

3.6.1 Activity Login

Gambar 3.2 merupakan desain *activity* diagram dari *usecase* login. Pengguna yang telah terdaftar atau disebut sebagai Talent mengisi form login, kemudian menekan tombol “Login”. Setelah itu sistem akan memeriksa apakah pengguna sudah terdaftar dalam database, jika belum maka akan keluar pemberitahuan bahwa akun belum terdaftar, jika sudah, maka sistem akan memeriksa apakah *email* akun tersebut sudah diverifikasi, jika belum maka akan keluar pemberitahuan bahwa *email* belum diverifikasi, jika sudah maka akan diarahkan ke Halaman Daftar Kelas.

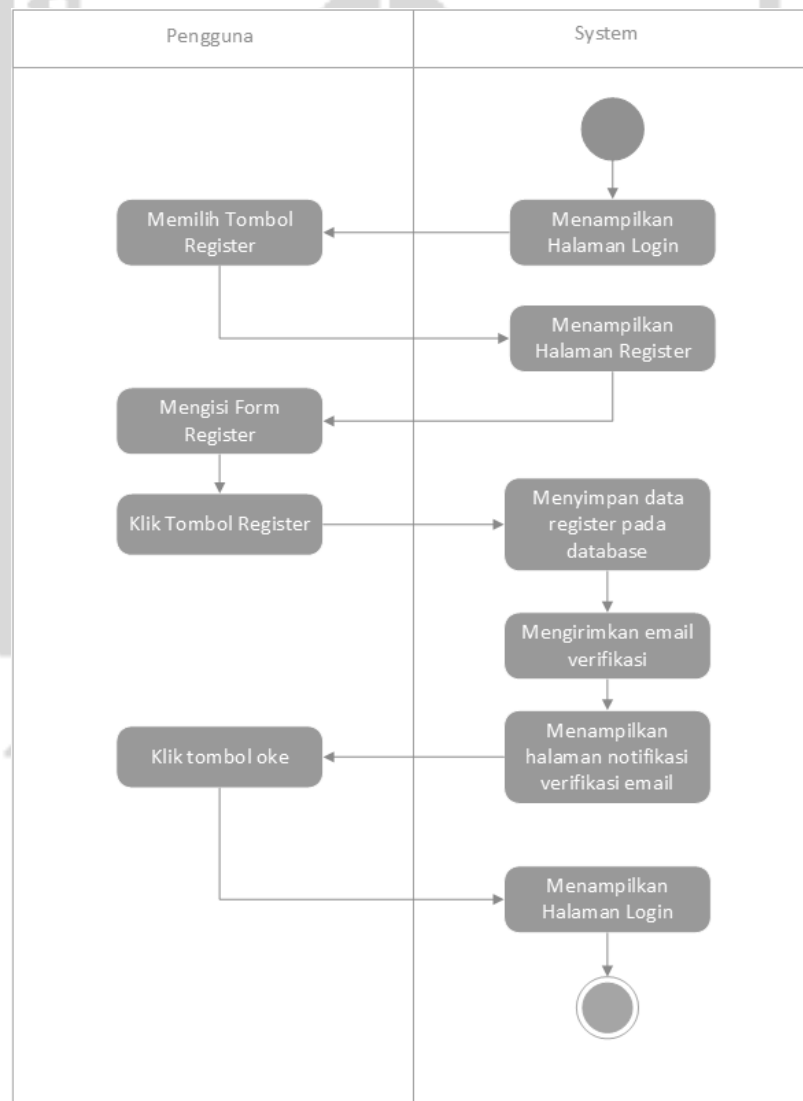




Gambar 3.2 Activity Login

3.6.2 Activity Register

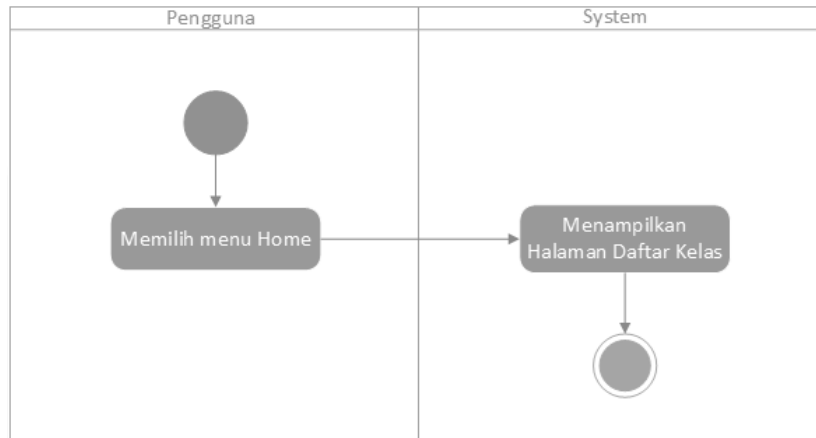
Gambar 3.3 merupakan desain *activity* diagram dari *usecase* register. Pada saat mengakses aplikasi, halaman awal yang ditampilkan adalah halaman login, pengguna dapat memilih tombol “Register” untuk melakukan pendaftaran akun. Setelah memilih “Register”, maka sistem akan mengarahkan ke halaman yang berisi form registrasi pengguna. Pengguna memasukan data registrasi lalu menekan tombol “Register”, kemudian sistem akan menyimpan data registrasi pada database, mengirimkan *email* verifikasi dan mengarahkan pengguna ke halaman notifikasi untuk memberi tahu untuk mengaktifkan akunnya melalui *email* yang di daftarkan. Ketika pengguna menekan tombol “Oke”, maka sistem akan mengarahkan ke halaman login.



Gambar 3.3 Activity Register

3.6.3 Activity Lihat Daftar Kelas

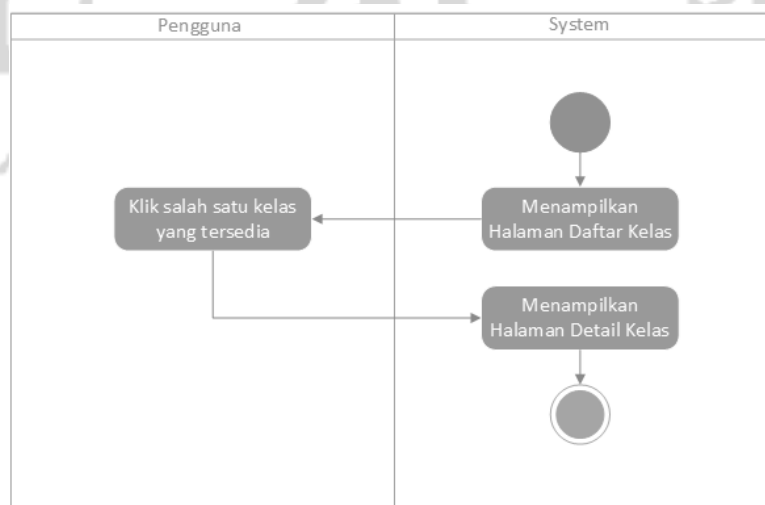
Gambar 3.4 merupakan desain *activity diagram usecase* Lihat Daftar Kelas. Halaman ini adalah halaman yang pertama kali ditampilkan setelah talent berhasil login. Talent juga dapat mengakses halaman ini dengan cara memilih menu Home pada aplikasi, yang kemudian sistem akan menampilkan Halaman Daftar Kelas.



Gambar 3.4 Activity Lihat Daftar Kelas

3.6.4 Activity Lihat Detail Kelas

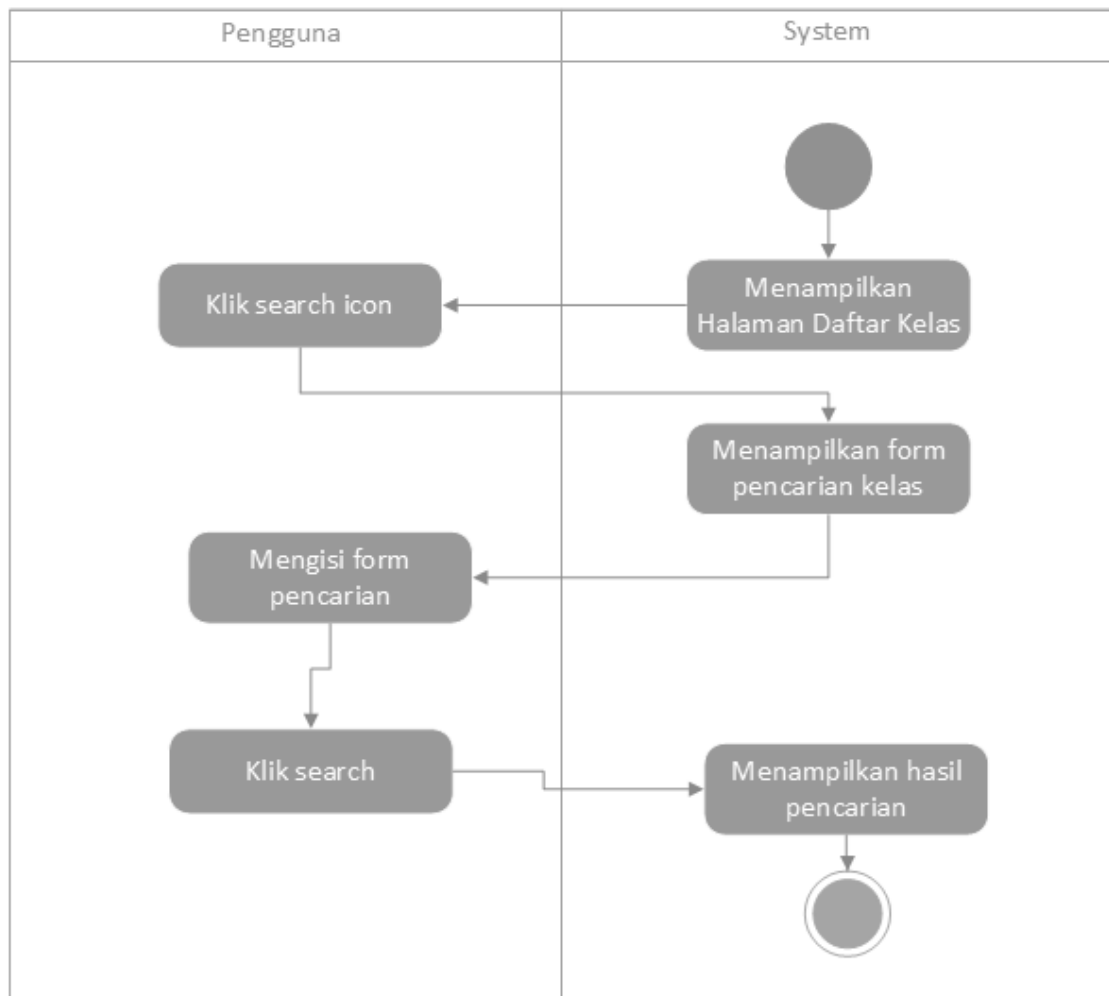
Gambar 3.5 merupakan desain *activity diagram usecase* Lihat Detail Kelas. Halaman ini dapat diakses ketika talent memilih salah satu kelas dalam daftar kelas. Kemudian system akan menampilkan Halaman Detail Kelas.



Gambar 3.5 Activity Lihat Detail Kelas

3.6.5 Activity Cari Kelas

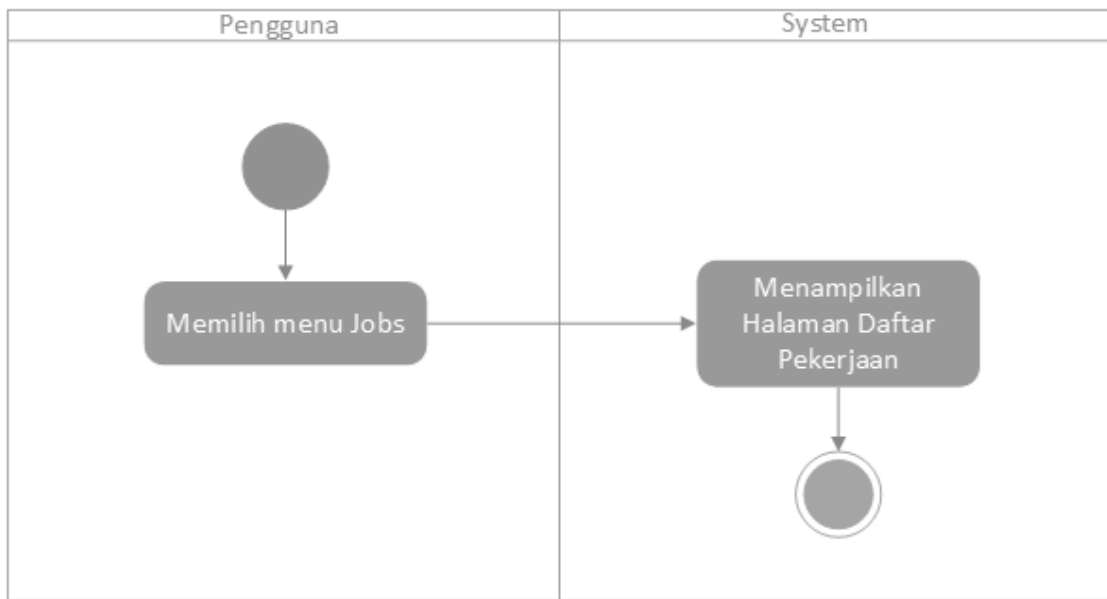
Gambar 3.6 merupakan desain *activity diagram usecase* Cari Kelas. Talent dapat mencari kelas berdasarkan nama kelas setelah menekan *icon* pencarian pada halaman daftar kelas. Setelah mengisi form pencarian dan menekan tombol “Search”, maka sistem akan menampilkan hasil pencarian.



Gambar 3.6 Activity Cari Kelas

3.6.6 Activity Lihat Daftar Pekerjaan

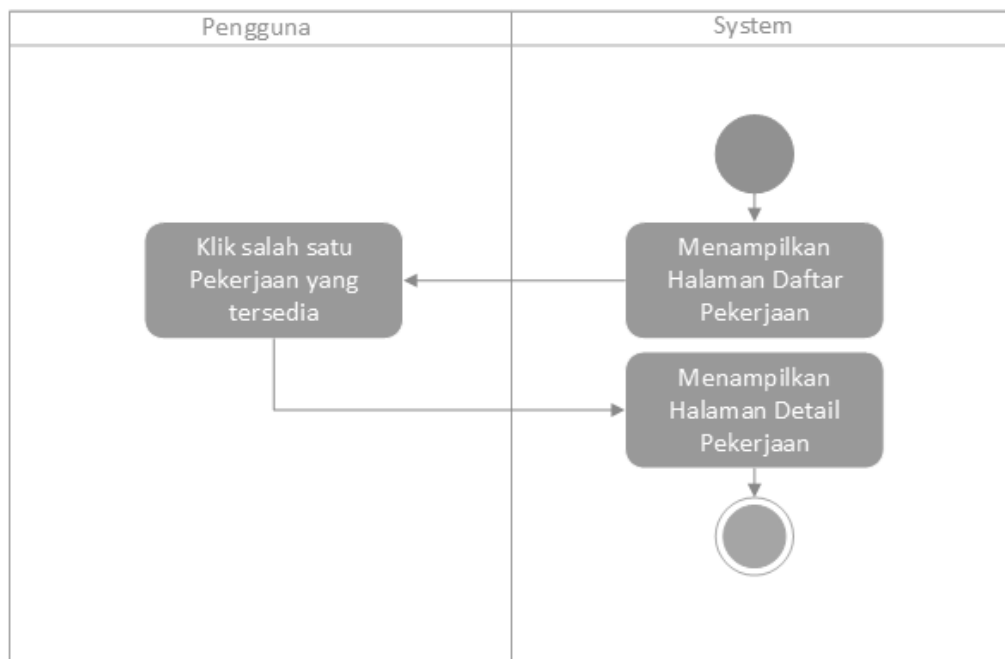
Gambar 3.7 merupakan desain *activity diagram usecase* Lihat Daftar Pekerjaan. Talent juga dapat mengakses halaman ini dengan cara memilih menu Jobs pada aplikasi, yang kemudian sistem akan menampilkan Halaman Daftar Pekerjaan.



Gambar 3.7 Activity Lihat Daftar Pekerjaan

3.6.7 Activity Lihat Detail Pekerjaan

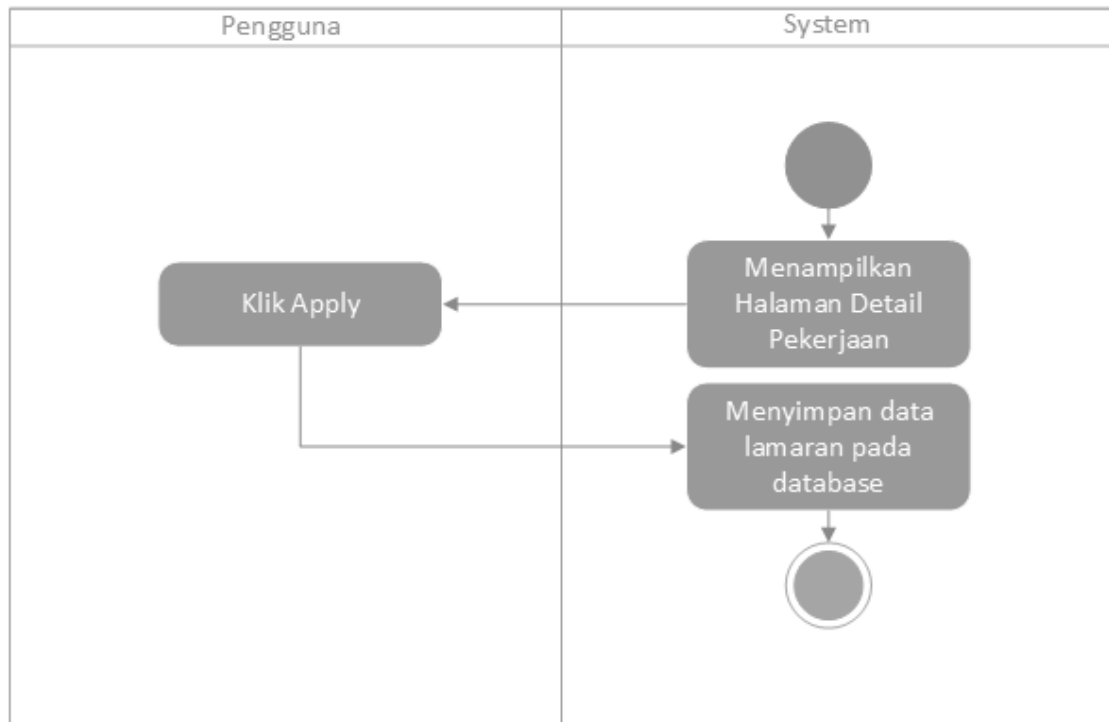
Gambar 3.8 merupakan desain *activity* diagram *usecase* Lihat Detail Pekerjaan. Halaman ini dapat diakses ketika talent memilih salah satu kelas dalam daftar pekerjaan. Kemudian system akan menampilkan Halaman Detail Pekerjaan.



Gambar 3.8 Activity Lihat Detail Pekerjaan

3.6.8 Activity Lamar Pekerjaan

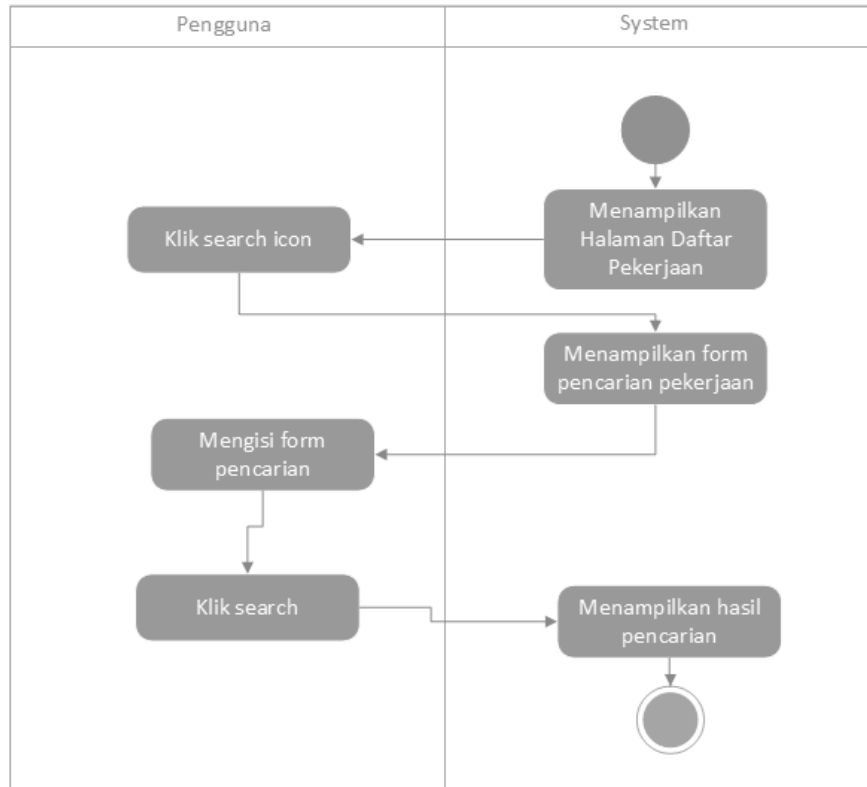
Gambar 3.9 merupakan desain *activity* diagram dari *usecase* Lamar Pekerjaan. Talent dapat melamar pekerjaan tertentu dengan cara menekan tombol “Apply” pada Halaman Detail Pekerjaan yang sudah dipilihnya. Kemudian sistem akan menyimpan data lamaran pekerjaan ke database.



Gambar 3.9 Activity Lamar Pekerjaan

3.6.9 Activity Cari Pekerjaan

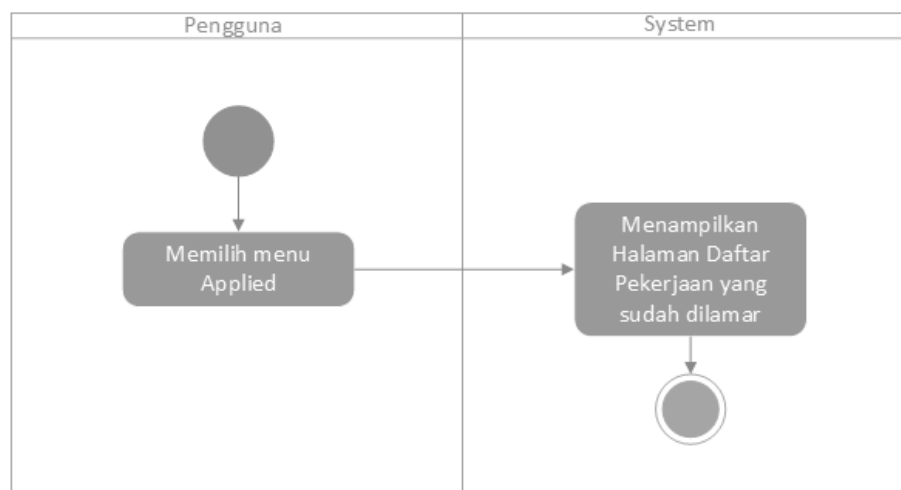
Gambar 3.10 merupakan desain *activity* diagram *usecase* Cari Pekerjaan. Talent dapat mencari pekerjaan berdasarkan nama pekerjaan setelah menekan *icon* pencarian pada halaman daftar pekerjaan. Setelah mengisi form pencarian dan menekan tombol “Search”, maka sistem akan menampilkan hasil pencarian.



Gambar 3.10 Activity Cari Pekerjaan

3.6.10 Activity Lihat Daftar Pekerjaan Terlarang

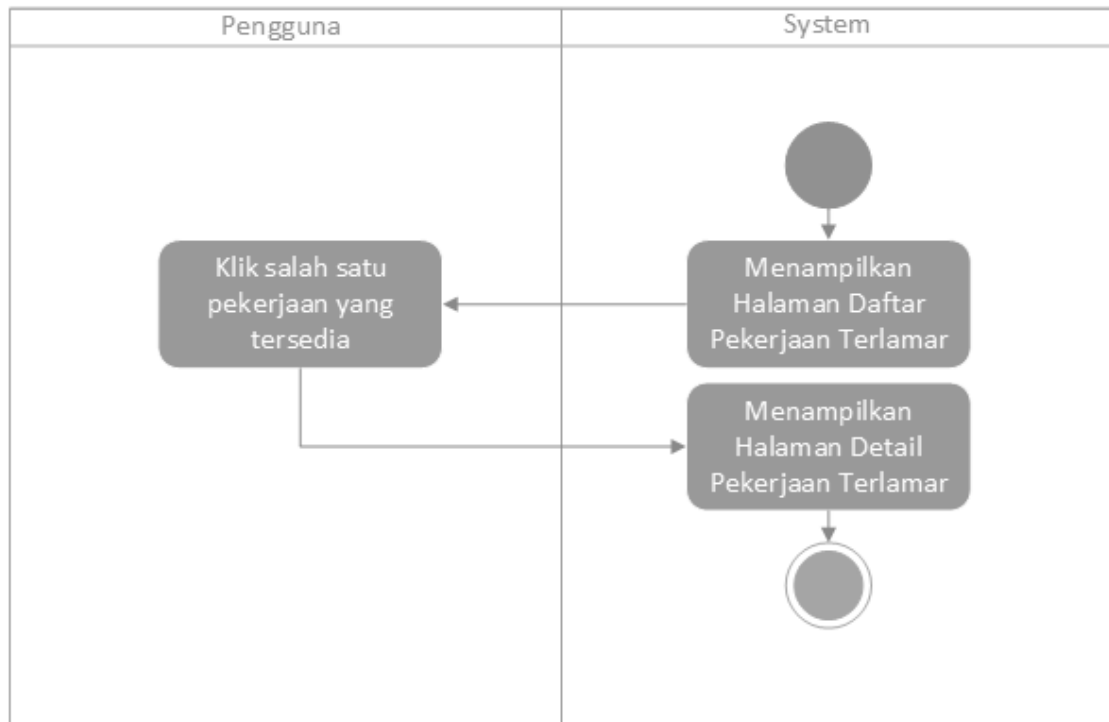
Gambar 3.11 merupakan desain *activity* dari *usecase* Lihat Daftar Pekerjaan Terlarang. Talent dapat melihat pekerjaan apa saja yang sudah dilamar sebelumnya beserta statusnya dengan cara mengakses menu “Applied” pada dalam aplikasi.



Gambar 3.11 Activity Lihat Daftar Pekerjaan Terlarang

3.6.11 Activity Lihat Detail Pekerjaan Terlaran

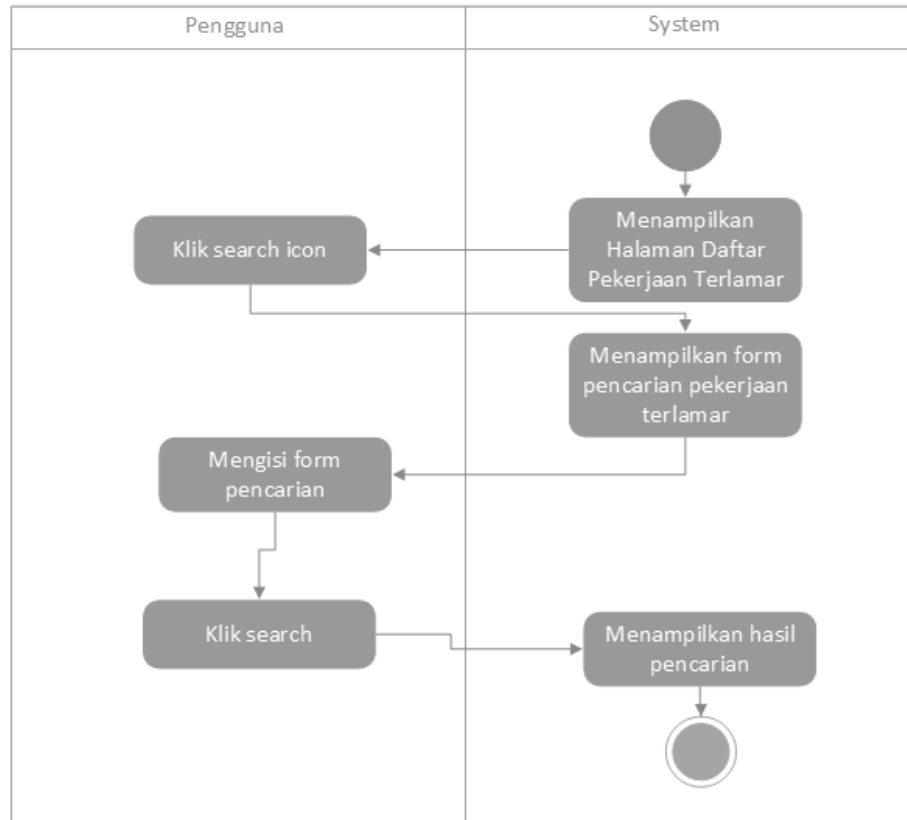
Gambar 3.12 merupakan desain *activity* diagram *usecase* Lihat Detail Pekerjaan Terlaran. Halaman ini dapat diakses ketika talent memilih salah satu kelas dalam daftar pekerjaan terlaran. Kemudian system akan menampilkan Halaman Detail Pekerjaan Terlaran.



Gambar 3.12 Activity Lihat Detail Pekerjaan Terlaran

3.6.12 Activity Cari Pekerjaan Terlaran

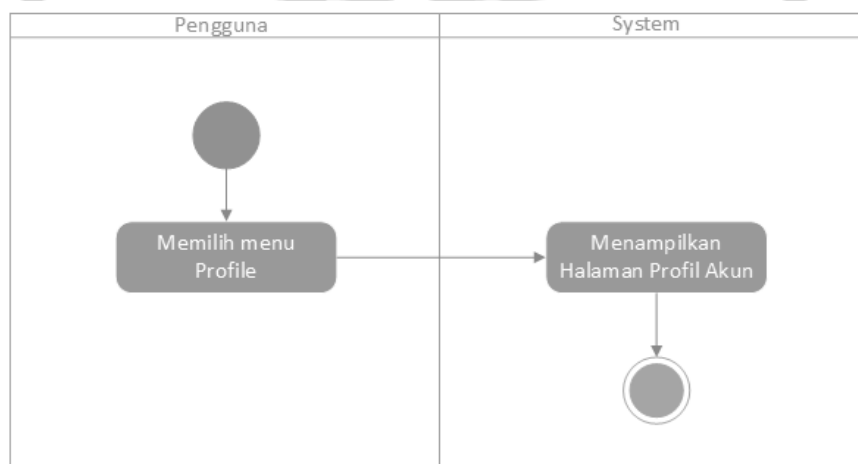
Gambar 3.13 merupakan desain *activity* diagram *usecase* Cari Pekerjaan Terlaran. Talent dapat mencari pekerjaan terlaran berdasarkan nama pekerjaan setelah menekan *icon* pencarian pada halaman daftar pekerjaan terlaran. Setelah mengisi form pencarian dan menekan tombol "Search", maka sistem akan menampilkan hasil pencarian.



Gambar 3.13 Activity Cari Pekerjaan Terlarar

3.6.13 Activity Lihat Profil

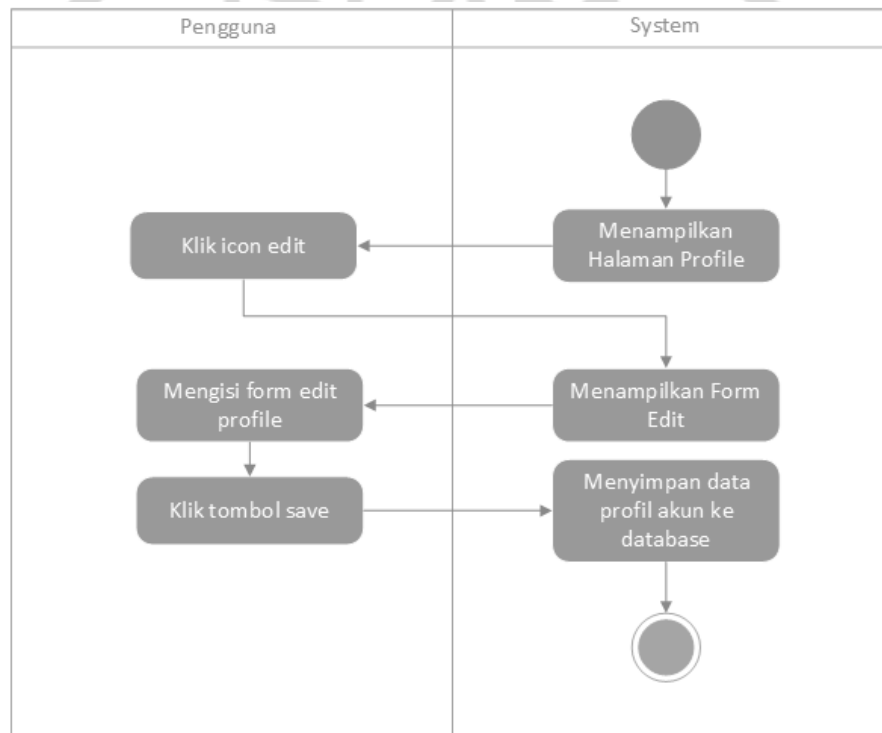
Gambar 3.14 merupakan *activity* diagram dari *usecase* Lihat Profil Akun. Talent dapat melihat informasi tentang profilnya sendiri dengan cara menekan menu “Profile” pada aplikasi.



Gambar 3.14 Activity Lihat Profil

3.6.14 Activity Sunting Profil

Gambar 3.15 merupakan desain *activity* diagram dari *usecase* Sunting *Profile*. Talent dapat merubah informasi profilnya setelah menekan tombol *icon* edit pada Halaman *Profile*, kemudian sistem akan menampilkan form untuk Edit *Profile*, pengguna dapat merubah dan mengisi informasi pada form yang sudah disediakan. Untuk menyimpan informasi, talent dapat menekan tombol “save” yang kemudian sistem akan menyimpan perubahan informasi tersebut dalam database.



Gambar 3.15 Activity Sunting Profil

3.7 Perancangan Antarmuka dan Skema Adopsi Flutter

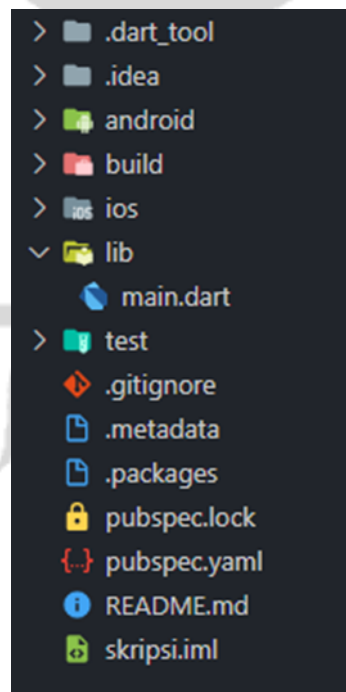
Dalam perancangan antarmuka dan skema adopsi flutter ini, penulis akan memaparkan terkait struktur direktori flutter, *dependencies* tambahan yang akan digunakan, dan *widget* yang akan digunakan dalam implementasi antarmuka.

Widget dalam Flutter merupakan elemen yang sangat penting dalam membangun antarmuka. Keseluruhan antarmuka aplikasi dan navigasi dibangun dengan menggunakan *widget*.

Pada perancangan antarmuka ini, penulis hanya akan menuliskan *widget-widget* yang membentuk objek tampilan selain *Scaffold*, *Container*, *Padding* dan *Margin* yang notabene menjadi kerangka dalam setiap antarmuka yang dibangun.

3.7.1 Struktur Direktori Project Flutter

Gambar 3.16 merupakan direktori default sebuah project baru dalam flutter. Dalam struktur direktori flutter terdapat *folder lib* yang secara default berisi *file main.dart* yg akan dcompile ketika program dijalankan.



Gambar 3.16 Struktur Direktori Flutter

Penulis akan menambahkan *folder* api untuk menampung *file* yang berisi kode untuk berkomunikasi dengan *API*, *folder* layout digunakan untuk menampung *file* yang berisi kode tampilan dan *folder* model digunakan untuk menampung *file* kode yang digunakan untuk memodelkan json agar dapat terbaca oleh flutter.

Penulis akan menambah *folder assets* yang di dalamnya berisi *folder fonts* untuk menampung *file font*, *folder icons* untuk menampung *file-file icon* dan *folder images* untuk menampung *file* gambar.

File *pubspec.yaml* digunakan untuk menambahkan daftar package *dependencies*, daftar *assets* dan *font* agar dapat dibaca oleh flutter.

3.7.2 Penggunaan Packages Dependencies

Packages Dependencies yang akan digunakan untuk aplikasi *front-end mobile apps e-recruitment* agar dapat berjalan sesuai kebutuhan adalah sebagai berikut:

a. Http

Merupakan *dependencies* untuk membantu flutter berkomunikasi dengan *web services* dan dapat mengkonsumsi *API* dengan metode GET, POST, PUT, PATCH, DELETE.

b. Shared Preferences

Merupakan *dependencies* yang akan digunakan untuk menyimpan data sederhana secara persisten.

c. Flutter Svg

Dependencies ini digunakan untuk membantu flutter membaca *file* dengan ekstensi *.svg*.

d. Url Launcher

Merupakan *dependencies* yang akan digunakan agar flutter dapat mengalihkan data url agar dapat dibuka di aplikasi pihak ketiga.

e. Toast

Merupakan *dependencies* yang berfungsi untuk menampilkan pesan pop-up sederhana.

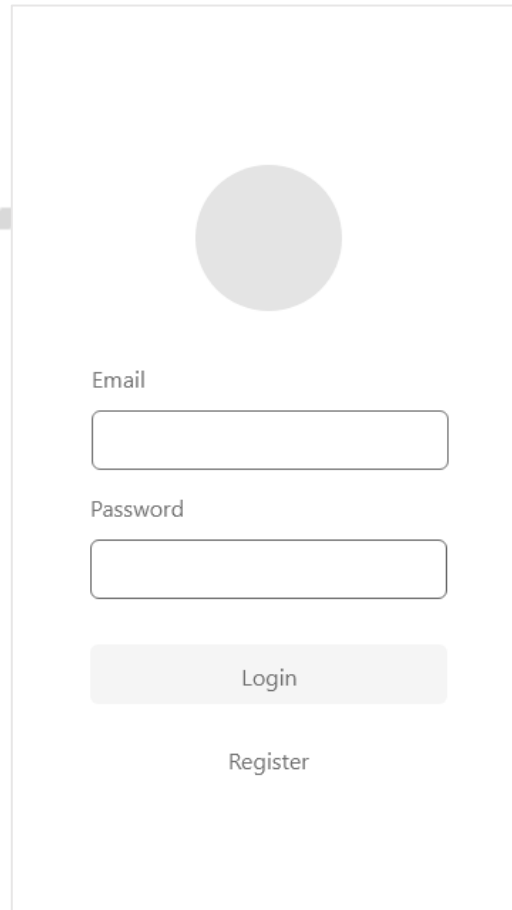
f. Image Picker

Dependencies ini akan digunakan untuk membantu mengambil data image dari storage device.

3.7.3 Antarmuka Login

Gambar 3.17 merupakan rancangan antarmuka halaman login. Halaman ini menjadi halaman awal pada saat membuka aplikasi. Terdapat form login yaitu *email* dan *password*,

hanya pengguna yang terdaftar dan memverifikasi *email* yang dapat melakukan login, jika tidak terdaftar atau belum memverifikasi *email* maka akan muncul alert sesuai kondisinya pada halaman ini.

The image shows a mobile application login screen. At the top center is a large grey circle representing a profile picture. Below it, the word "Email" is followed by a white rectangular input field with a thin border. Underneath that, the word "Password" is followed by another white rectangular input field with a thin border. Below the password field is a grey rounded rectangular button with the text "Login" in white. At the bottom of the screen is a white rounded rectangular button with the text "Register" in grey. The entire interface is centered on a white background.

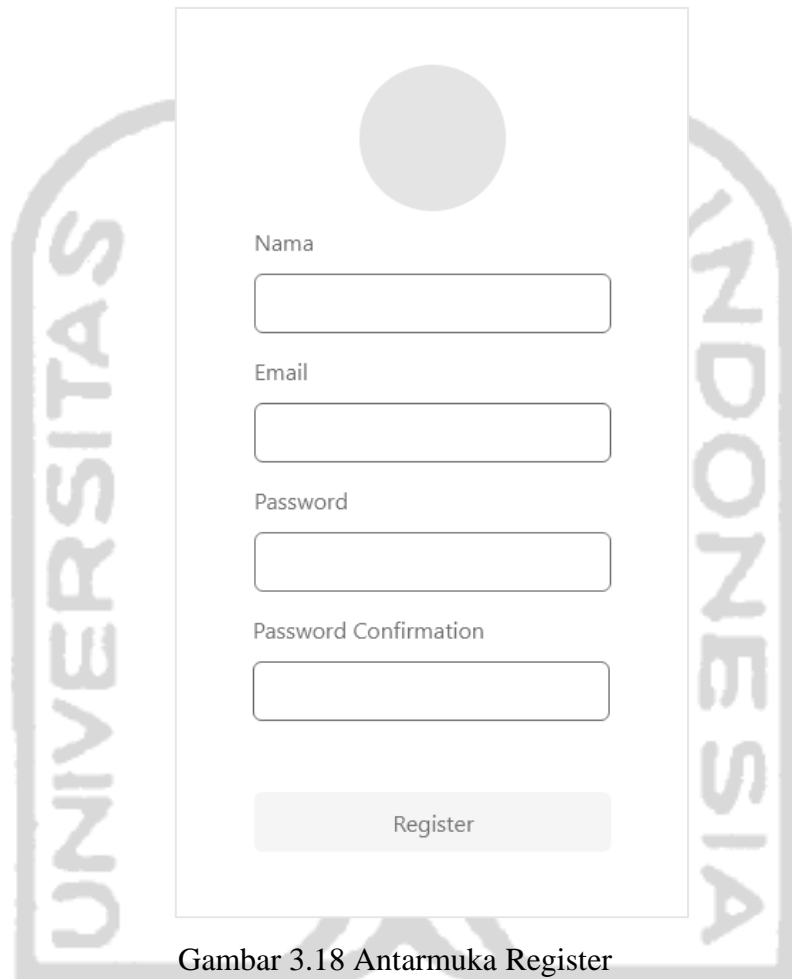
Gambar 3.17 Antarmuka Login

Pada Antarmuka Login ini akan diimplementasikan beberapa *widget* Flutter seperti :

- a. *Widget* `SvgPicture.asset` yang merupakan *dependencies* dari Flutter `svg` yang digunakan untuk menempatkan logo berupa *file* `svg`.
- b. *Widget* `TextFormField` untuk handle form *email* dan *password*
- c. *Widget* `RaisedButton` untuk handle tombol login
- d. *Widget* `FlatButton` untuk handle tombol register
- e. *Widget* `CircularProgressIndicator` untuk handle proses loading
- f. *Widget* `Toast` untuk menampilkan pesan sederhana

3.7.4 Antarmuka Register

Gambar 3.18 merupakan rancangan antarmuka dari halaman registrasi untuk pengguna yang belum terdaftar. Pada halaman ini terdapat form registrasi seperti nama, *email*, *password* dan *password confirmation*.



The image shows a registration form layout. At the top, there is a large grey circle representing a profile picture. Below it, there are four text input fields labeled 'Nama', 'Email', 'Password', and 'Password Confirmation'. At the bottom of the form is a grey button labeled 'Register'. The entire form is centered on a white background with a faint watermark of a university logo and the text 'UNIVERSITAS INDONESIA' in the background.

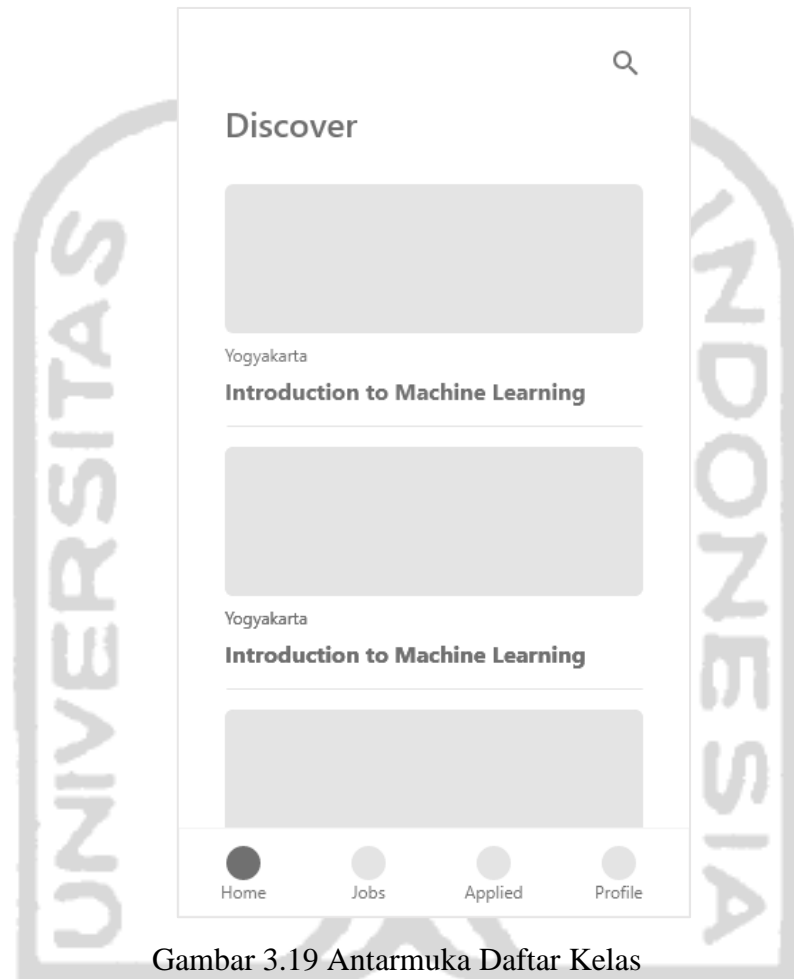
Gambar 3.18 Antarmuka Register

Pada Antarmuka Register ini akan diimplementasikan beberapa *widget* Flutter seperti :

- Widget* `SvgPicture.asset` yang merupakan *dependencies* dari Flutter `svg` yang digunakan untuk menempatkan logo berupa *file* `svg`.
- Widget* `TextFormField` untuk *handle* form nama, *email*, *password* dan *password confirmation*
- Widget* `RaisedButton` untuk *handle* tombol register
- Widget* `CircularProgressIndicator` untuk *handle* proses loading
- Widget* `Toast` untuk menampilkan pesan sederhana

3.7.5 Antarmuka Daftar Kelas

Gambar 3.19 merupakan rancangan antarmuka halaman Daftar Kelas. Halaman ini ditampilkan pertama kali setelah Talent melakukan login. Berisi daftar kelas yang tersedia dalam sistem.



Gambar 3.19 Antarmuka Daftar Kelas

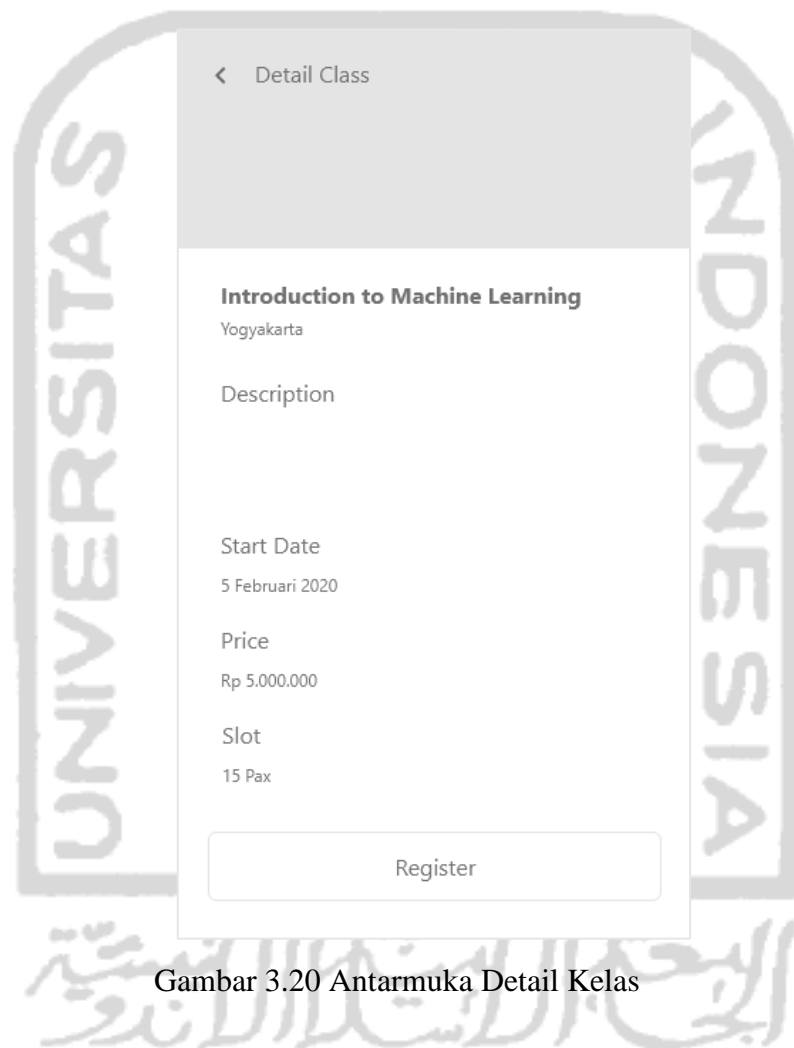
Pada Antarmuka Daftar Kelas ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- a. *Widget IconButton* yang digunakan untuk menempatkan *icon* search berupa *file* *svg* yang dibaca menggunakan *SvgPicture* dari *dependencies* Flutter *svg*.
- b. *Widget SliverAppBar* Digunakan untuk menampung *icon* search dan judul "Discover" yg akan menyesuaikan ketika konten di scroll
- c. *Widget SliverList* digunakan untuk menampung konten-konten berupa konten gambar, nama kota dan nama kelas yang ada di dalam *Widget Card* dan diiterasi sesuai jumlah datanya.

- d. *Widget* `CircularProgressIndicator` untuk *handle* proses loading

3.7.6 Antarmuka Detail Kelas

Gambar 3.20 merupakan rancangan antarmuka halaman Detail kelas. Talent dapat mengakses halaman ini dengan memilih salah satu kelas pada daftar kelas, setelah itu sistem akan menampilkan halaman berisi rincian informasi terkait kelas tertentu



Gambar 3.20 Antarmuka Detail Kelas

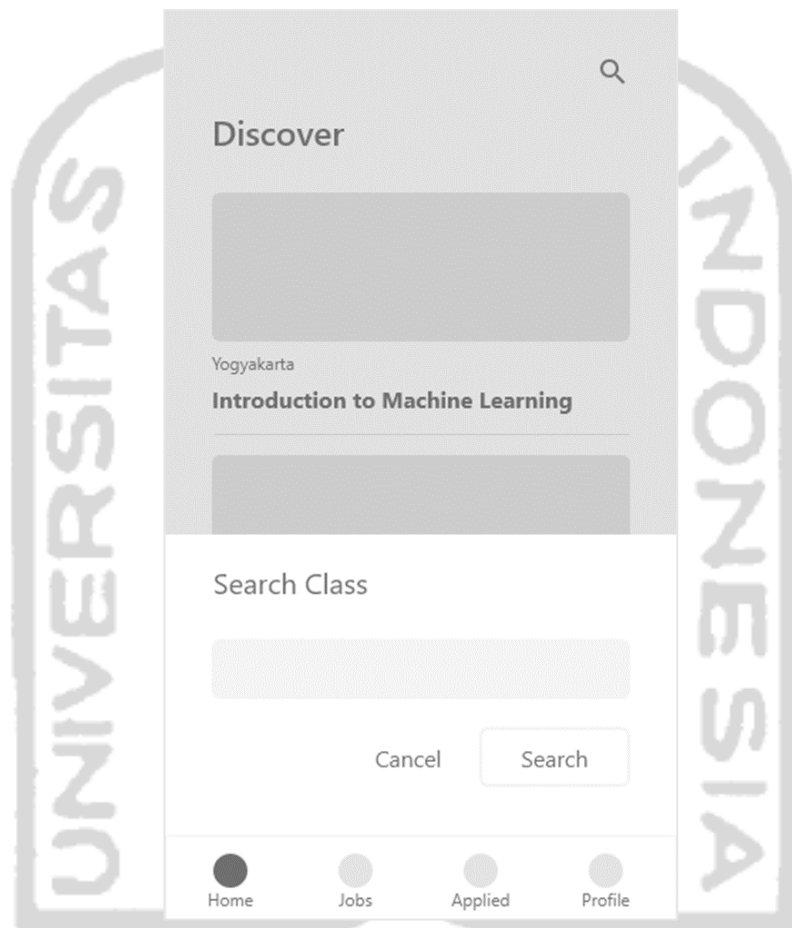
Pada Antarmuka Detail Kelas ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- Widget* `SliverAppBar` dengan *backgorund* *Widget* `Image.network` digunakan sebagai header page.
- Widget* `Text` untuk *handle* data nama, kota, deskripsi, tanggal mulai, harga dan slot.
- Widget* `OutlineButton` untuk *handle* tombol register

- d. *Widget* `CircularProgressIndicator` untuk *handle* proses loading

3.7.7 Antarmuka Cari Kelas

Gambar 3.21 merupakan rancangan antarmuka form cari kelas. Form pencarian dapat diakses setelah menekan *icon* pencarian pada halaman daftar kelas.



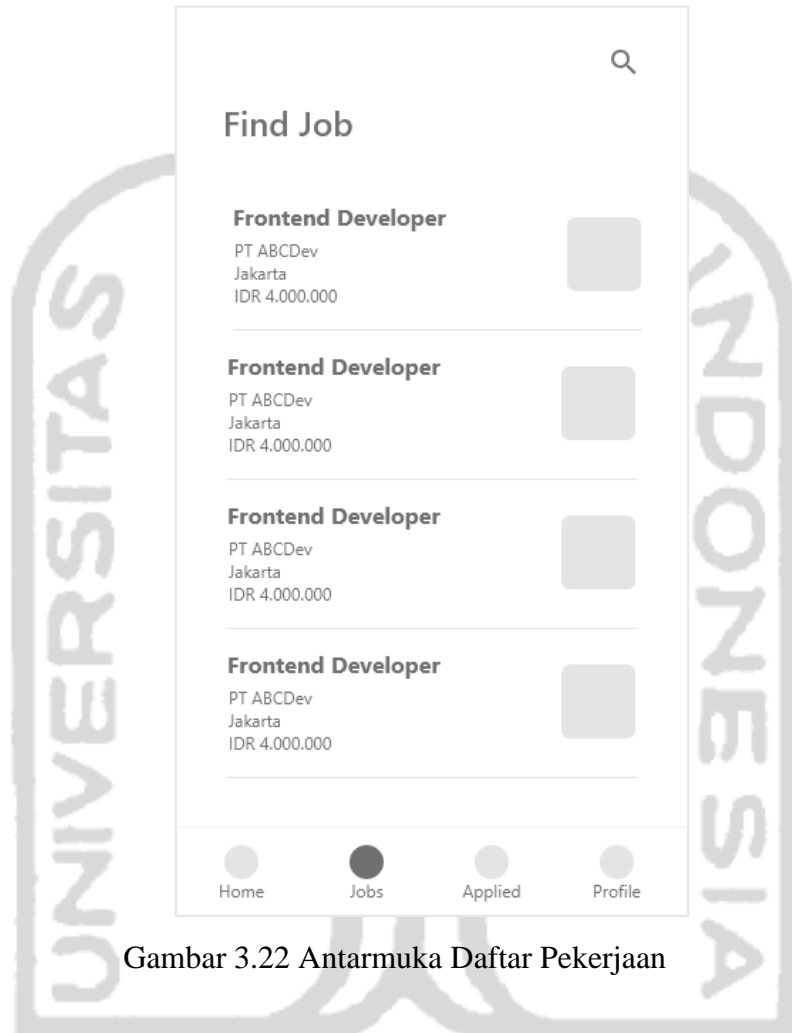
Gambar 3.21 Antarmuka Cari Kelas

Pada Antarmuka Cari Kelas ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- Widget* `ShowBottomSheet` digunakan untuk memunculkan form pencarian
- Widget* `Text` untuk *handle* kata "search class"
- Widget* `OutlineButton` untuk *handle* tombol "search"
- Widget* `FlatButton` untuk *handle* tombol "Cancel"

3.7.8 Antarmuka Daftar Pekerjaan

Gambar 3.22 merupakan rancangan antarmuka halaman Daftar Pekerjaan. Halaman ini berisi daftar pekerjaan yang tersedia dalam sistem.



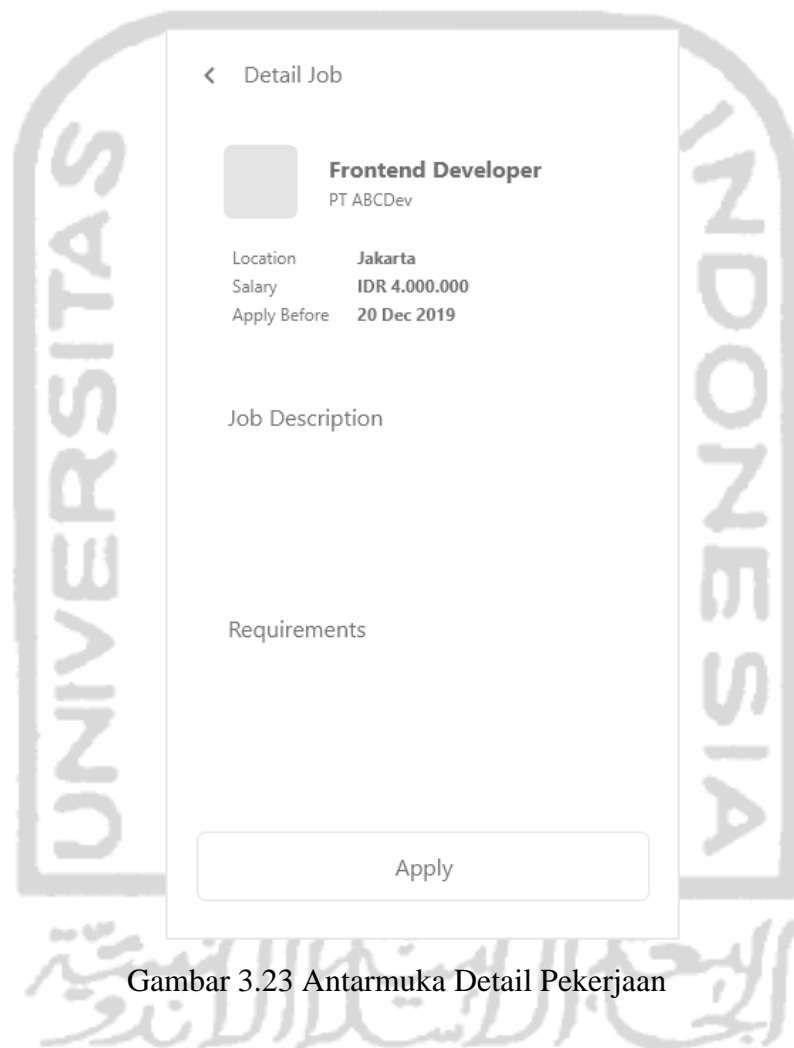
Gambar 3.22 Antarmuka Daftar Pekerjaan

Pada Antarmuka Daftar Pekerjaan ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- a. *Widget IconButton* yang digunakan untuk menempatkan *icon* search berupa *file* *svg* yang dibaca menggunakan *SvgPicture* dari *dependencies* Flutter *svg*.
- b. *Widget SliverAppBar* Digunakan untuk menampung *icon* search dan judul "Find Job" yg akan menyesuaikan ketika konten di scroll
- c. *Widget SliverList* digunakan untuk menampung konten-konten berupa konten gambar, nama pekerjaan, nama kota, nama perusahaan dan gaji yang ada di dalam *Widget ListTile* dan diiterasi sesuai jumlah datanya.
- d. *Widget CircularProgressIndicator* untuk handle proses loading.

3.7.9 Antarmuka Detail Pekerjaan

Gambar 3.23 merupakan rancangan antarmuka halaman Detail Pekerjaan. Talent dapat mengakses halaman ini dengan menekan salah pekerjaan dalam daftar pekerjaan, setelah itu sistem menampilkan halaman berisi rincian informasi terkait pekerjaan tertentu.



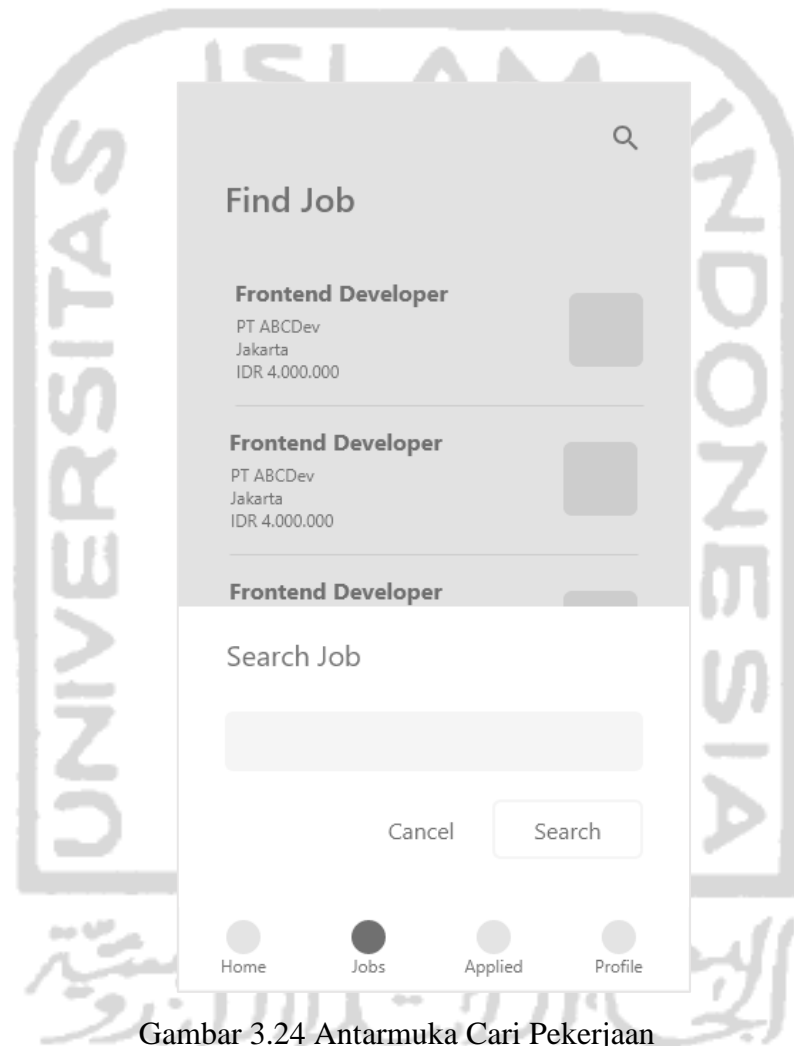
Gambar 3.23 Antarmuka Detail Pekerjaan

Pada Antarmuka Detail Pekerjaan ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- Widget* ListTile untuk menampung data gambar, nama pekerjaan dan nama kota.
- Widget* Text untuk menghandle job description dan requirements.
- Widget* OutlineButton untuk menghandle tombol apply
- Widget* Toast untuk menampilkan pesan sederhana ketika apply berhasil
- Widget* CircularProgressIndicator untuk menghandle proses loading

3.7.10 Antarmuka Cari Pekerjaan

Gambar 3.24 merupakan rancangan antarmuka Cari Pekerjaan. Ketika talent menekan *icon* pencarian yang terdapat dalam halaman Daftar Pekerjaan, maka form pencarian ini akan muncul, talent dapat mencari pekerjaan sesuai nama, lokasi dan atau gaji minimal pekerjaan yang dicari.



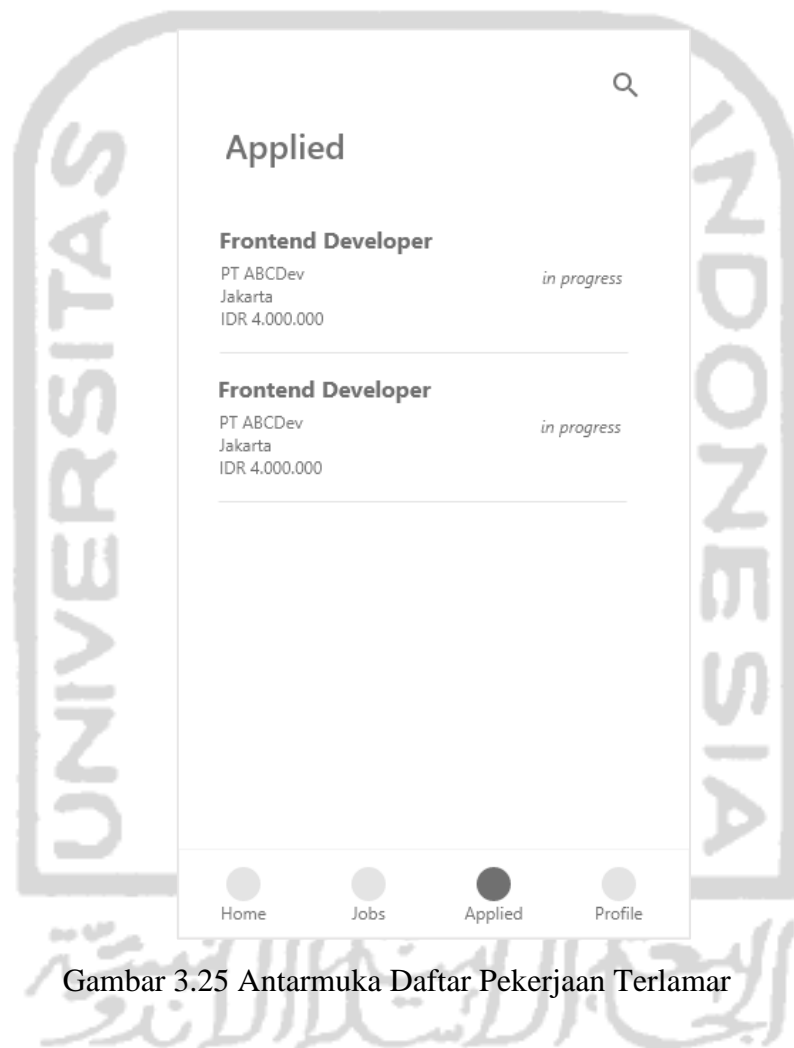
Gambar 3.24 Antarmuka Cari Pekerjaan

Pada Antarmuka Cari Pekerjaan ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- f. *Widget* ShowBottomSheet digunakan untuk memunculkan form pencarian
- g. *Widget* Text untuk menghandle kata "search job"
- h. *Widget* OutlineButton untuk menghandle tombol "search"
- i. *Widget* FlatButton untuk menghandle tombol "Cancel"

3.7.11 Antarmuka Daftar Pekerjaan Terlaran

Gambar 3.25 merupakan rancangan antarmuka Daftar Pekerjaan Terlaran. Halaman ini dapat diakses oleh talent melalui Halaman *Profile* dengan cara menekan tombol “Applied”. Halaman ini berisi daftar pekerjaan yang sudah dilamar oleh talent.



Gambar 3.25 Antarmuka Daftar Pekerjaan Terlaran

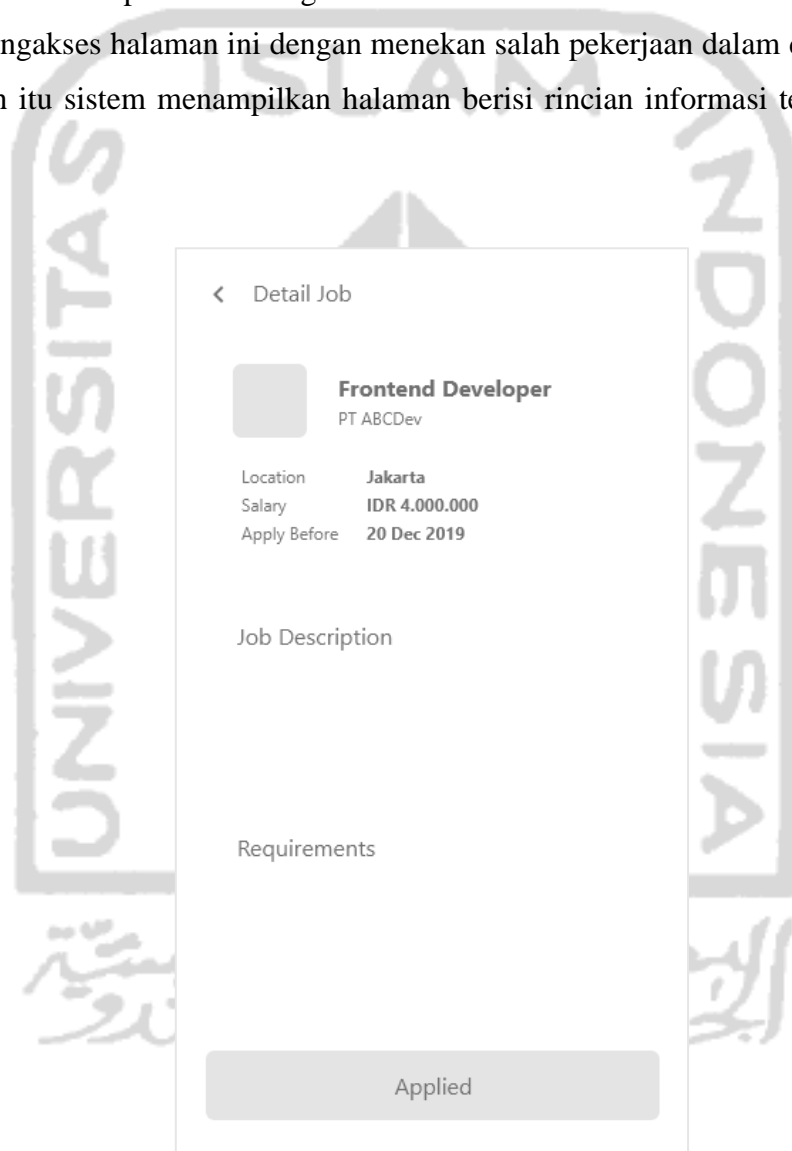
Pada Antarmuka Daftar Pekerjaan ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- a. *Widget IconButton* yang digunakan untuk menempatkan *icon* search berupa *file* *svg* yang dibaca menggunakan *SvgPicture* dari *dependencies* Flutter *svg*.
- b. *Widget SliverAppBar* Digunakan untuk menampung *icon* search dan judul ”Applied” yg akan menyesuaikan ketika konten di scroll

- c. *Widget SliverList* digunakan untuk menampung konten status, nama pekerjaan, nama kota, nama perusahaan dan gaji yang ada di dalam *Widget ListTile* dan diiterasi sesuai jumlah datanya.
- d. *Widget CircularProgressIndicator* untuk menghandle proses loading.

3.7.12 Antarmuka Lihat Detail Pekerjaan Terlaran

Gambar 3.26 merupakan rancangan antarmuka halaman Detail Pekerjaan Terlaran. Talent dapat mengakses halaman ini dengan menekan salah pekerjaan dalam daftar pekerjaan terlaran, setelah itu sistem menampilkan halaman berisi rincian informasi terkait pekerjaan tertentu.



Gambar 3.26 Antarmuka Detail Pekerjaan Terlaran

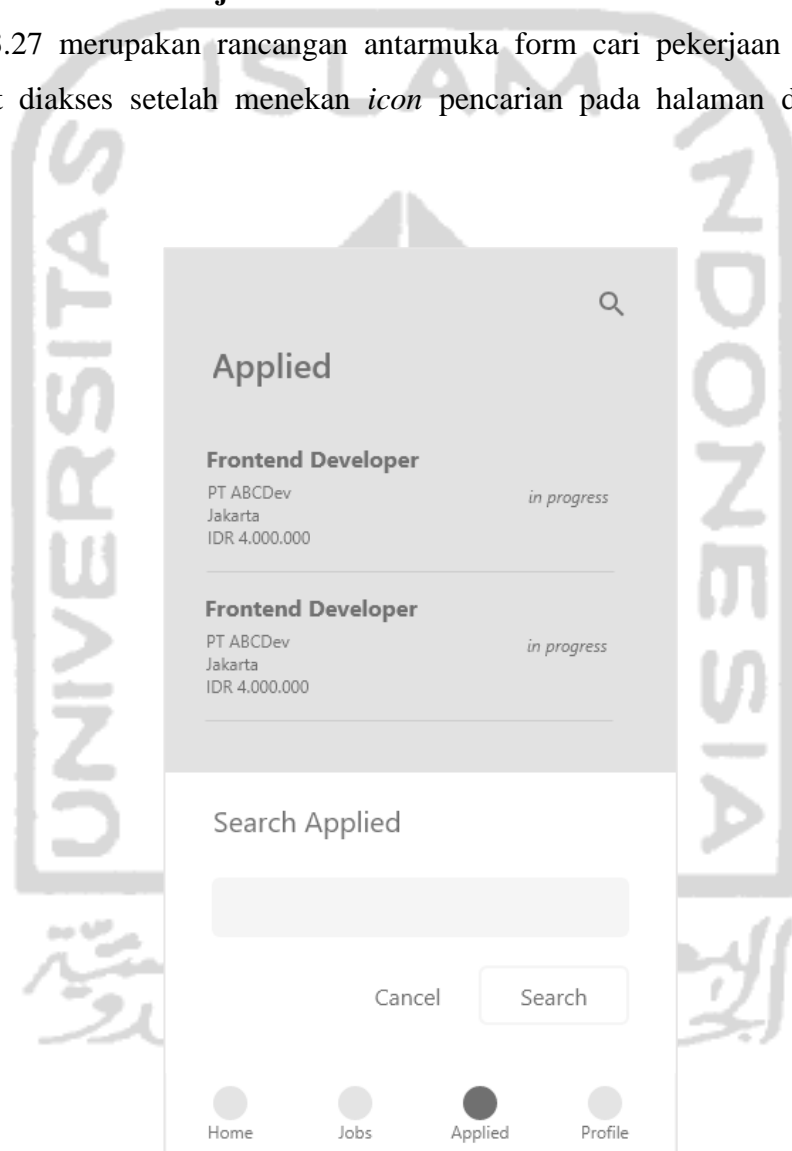
Pada Antarmuka Detail Pekerjaan Terlaran ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- e. *Widget ListTile* untuk menampung data gambar, nama pekerjaan dan nama kota.

- f. *Widget* Text untuk menghandle job description dan requirements.
- g. *Widget* OutlineButton untuk menghandle tombol apply
- h. *Widget* Toast untuk menampilkan pesan sederhana ketika apply berhasil
- i. *Widget* CircularProgressIndicator untuk menghandle proses loading

3.7.13 Antarmuka Cari Pekerjaan Terlarar

Gambar 3.27 merupakan rancangan antarmuka form cari pekerjaan terlarar. Form pencarian dapat diakses setelah menekan *icon* pencarian pada halaman daftar pekerjaan terlarar.



Gambar 3.27 Antarmuka Cari Pekerjaan Terlarar

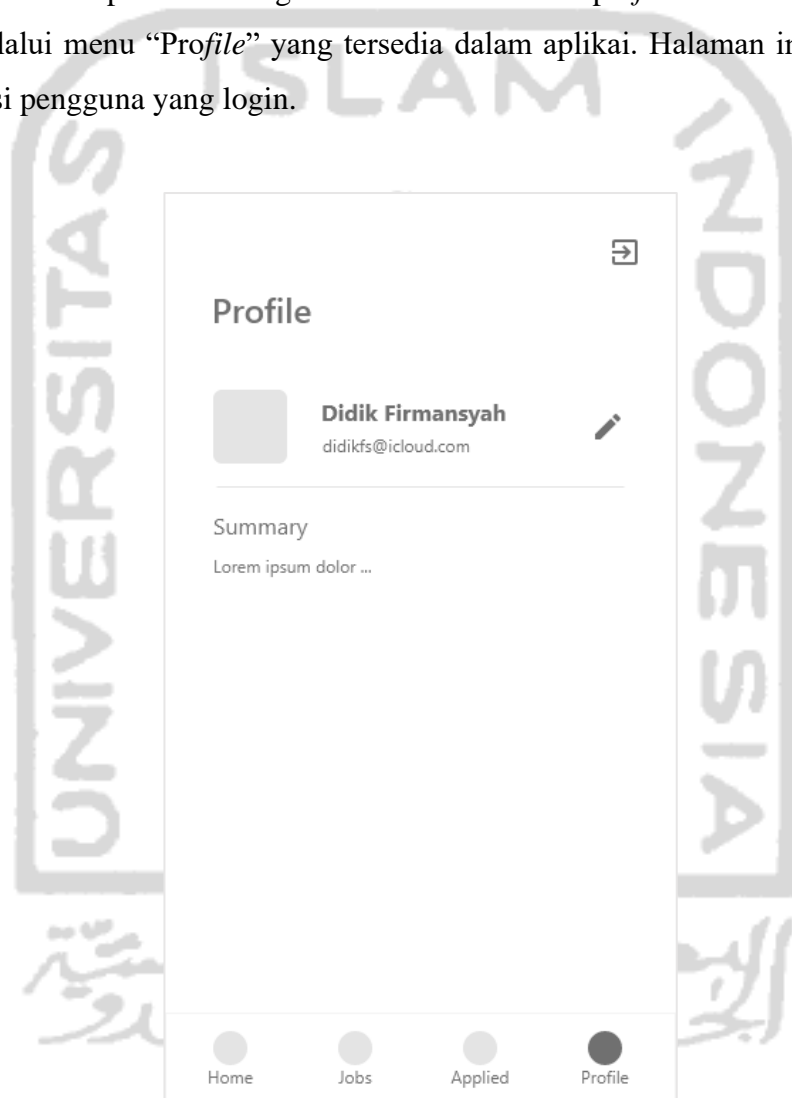
Pada Antarmuka Cari Pekerjaan Terlarar ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- j. *Widget* ShowBottomSheet digunakan untuk memunculkan form pencarian

- k. *Widget* Text untuk menghandle kata "search applied"
- l. *Widget* OutlineButton untuk menghandle tombol "search"
- m. *Widget* FlatButton untuk menghandle tombol "Cancel"

3.7.14 Antarmuka Profil

Gambar 3.28 merupakan rancangan antarmuka Halaman *profile*. Talent dapat mengakses halaman ini melalui menu "Profile" yang tersedia dalam aplikasi. Halaman ini berisi tentang rincian informasi pengguna yang login.



Gambar 3.28 Antarmuka Profil

Pada Antarmuka Profil ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut

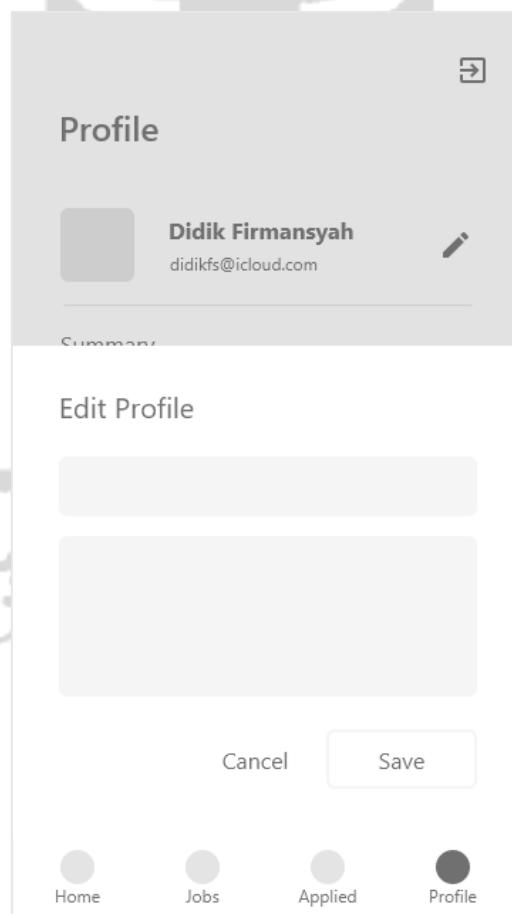
:

- a. *Widget* IconButton yang digunakan untuk menempatkan *icon* logout berupa *file* svg yang dibaca menggunakan *SvgPicture* dari *dependencies* Flutter *svg*.

- b. *Widget SliverAppBar* Digunakan untuk menampung *icon* search dan judul "Profile" yg akan menyesuaikan ketika konten di scroll
- c. *Widget ListTile* digunakan untuk menampung data gambar profil, nama pengguna, *email* pengguna, dan *icon* untuk edit
- d. *Widget Text* digunakan untuk menampung data summary yang berada di dalam *Widget Card*
- e. *Widget CircularProgressIndicator* untuk menghandle proses loading.

3.7.15 Antarmuka Sunting Profil

Gambar 3.31 merupakan raancangan antarmuka form sunting *profile*. Talent dapat mengakses halaman ini setelah menekan tombol *icon* edit pada Halaman *Profile*. Sistem akan menyimpan perubahan informasi setelah talent menekan tombol "save".



Gambar 3.29 Antarmuka Sunting Profil

Pada Antarmuka Cari Pekerjaan ini akan diimplementasikan beberapa *widget* Flutter sebagai berikut :

- a. *Widget ShowBottomSheet* digunakan untuk memunculkan form sunting profi
- b. *Widget TextFormField* digunakan untuk menghandle form sunting profil
- c. *Widget Text* untuk menghandle kata "Edit Profile"
- d. *Widget OutlineButton* untuk menghandle tombol "Save"
- e. *Widget FlatButton* untuk menghandle tombol "Cancel"

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Pada tahap ini penulis akan menjelaskan terkait pembuatan aplikasi berdasarkan analisis kebutuhan dan perancangan. Dalam implementasi ini akan dibahas mengenai pemodelan data, pembuatan presenter dan view dengan konteks pembuatan daftar kelas saja, kode program yang lain akan penulis cantumkan dalam lampiran.

4.1.1 Penambahan *Package, Assets Images, Assets Icon dan Font*

Dalam perancangan yang sudah ada, penulis perlu menambahkan beberapa package seperti *http*, *shared_preferences*, *flutter_svg*, *url_launcher*, *toast* dan *image picker*. Penulis menambahkan *folder api* untuk menghandle *file* presenter, layout untuk menghandle *file* view dan model untuk menghandle *file* pemodelan data. Penulis juga menambahkan *folder assets images* untuk menampung *file* gambar, *icons* untuk menampung *file icon* dan *fonts* untuk menampung *file font* yg akan digunakan dalam aplikasi. Struktur direktori setelah penambahan beberapa *folder* oleh penulis dapat dilihat pada Gambar 4.1.