

TRACER ALUMNI BERBASIS ANDROID

STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI

SKRIPSI

Karya Tulis sebagai syarat memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung

Disusun oleh:

DENI HIDAYAT

NIM. C1A150029



PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
BANDUNG
2019

LEMBAR PERSETUJUAN PENGUJI

**TRACER ALUMNI BERBASIS ANDROID
STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI**

Disusun oleh:

**DENI HIDAYAT
NIM. C1A150029**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Penguji 1

Penguji 2

Denny Rusdianto, S.T., M.Kom.
NIK. 04104808094

Yaya Suharya, S.Kom., M.T.
NIK. 01043170007

LEMBAR PERSETUJUAN PEMBIMBING

**TRACER ALUMNI BERBASIS ANDROID
STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI**

Disusun oleh:

DENI HIDAYAT

NIM. C1A150029

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Pembimbing 1

Pembimbing 2

Yudi Herdiana, S.T., M.T.

NIK. 04104808008

Nurul Imamah, S.T., M.T.

NIK. 04104808121

LEMBAR PENGESAHAN PROGRAM STUDI

**TRACER ALUMNI BERBASIS ANDROID
STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI**

Disusun oleh:

**DENI HIDAYAT
NIM. C1A150029**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Mengetahui,
Dekan,

Mengesahkan,
Ketua Program Studi

Yudi Herdiana, S.T., M.T.
NIK. 04104808008

Yaya Suharya, S.Kom., M.T.
NIK. 01043170007

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : Deni Hidayat

NIM : C1A150029

Judul Skripsi : **TRACER ALUMNI BERBASIS ANDROID STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI.**

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.

Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Agustus 2019

Yang membuat pernyataan,

Deni Hidayat

NIM. C1A150029

ABSTRACT

Tracer Study is a service provided by every educational institution to track alumni who have graduated from their institutions which can later be used for the accreditation and data collection of existing graduates aimed to determine the outcome of education in the form of transition from the world of education to the world of work. Alumni tracking has been done by the Faculty through the Google form web-based application which is located at <http://unibba.ac.id/fti/mahasiswa/alumni>. From the observations of the author of the application still has many shortcomings including, the use of functions on applications that have not been running well and as a solution the authors develop applications that are similar to being updated with android-based applications. The ease of alumni in accessing applications on the Google Play Store will make it easier for alumni to download and install tracer study applications on the alumni android mobile phones. This research uses Model Driven Development (MDD) which is written on the development of a google form tracer study information system to Android-based. By using the Firebase and Android platforms, it is hoped that the tracer study system will be able to assist in the tracer study process conducted at the Faculty of Information Technology at the University of Bale Bandung (FTI UNIBBA) and the results of the functionality and compatibility testing produced by the tracer study application. The process of filling data is easy and the familiar interface will find out the quality and quantity of alumni, and facilitate faculties in determining policies in order to improve the quality of graduates, the quality of education and accreditation. It is expected that the development of this application entitled "Android-Based Alumni Tracer Case Study of the Faculty of Information Technology" can provide an alternative solution for tracking alumni while making it easier to use.

Keywords: *alumni, android, development, tracer study.*

ABSTRAK

Tracer Study adalah sebuah layanan yang disediakan oleh setiap lembaga pendidikan untuk melacak alumni yang telah lulus dari lembaganya yang mana nantinya dapat digunakan untuk akreditasi dan pendaftaran para lulusan yang ada bertujuan untuk mengetahui *outcome* pendidikan dalam bentuk transisi dari dunia pendidikan ke dunia kerja. Pelacakan alumni sudah di lakukan oleh Fakultas melalui aplikasi berbasis *web google form* yang beralamat <http://unibba.ac.id/fti/> mahasiswa/alumni. Dari hasil pengamatan penulis aplikasi tersebut masih memiliki banyak kekurangan diantaranya, penggunaan fungsi pada aplikasi yang belum berjalan dengan baik dan sebagai solusinya penulis mengembangkan aplikasi yang serupa dengan diperbarui dengan aplikasi berbasis android. Kemudahan alumni dalam mengakses aplikasi di *google playstore* akan mempermudah alumni untuk mendownload dan menginstal aplikasi *tracer study* di *handphone* android masing-masing alumni. Penelitian ini menggunakan *Model Driven Development (MDD)* dimana yang dituliskan pada pengembangan dari sistem informasi *tracer study web google form* ke berbasis android. Dengan menggunakan *platform firebase* dan Android, diharapkan sistem *tracer study* mampu membantu dalam proses *tracer study* yang dilakukan di Fakultas Teknologi Informasi Universitas Bale Bandung (FTI UNIBBA) kemudian hasil pengujian fungsionalitas dan kompatibilitas yang dihasilkan oleh aplikasi *tracer study*. Proses pengisian data yang mudah serta *interface* yang familiar akan mengetahui kualitas serta kuantitas alumni, dan mempermudah fakultas dalam menentukan kebijakan dalam rangka meningkatkan kualitas lulusan, mutu pendidikan dan akreditas. Diharapkan pengembangan aplikasi ini yang berjudul “*Tracer Alumni Berbasis Android Studi Kasus Fakultas Teknologi Informasi*” bisa memberikan solusi alternatif untuk pelacakan alumni sekaligus memudahkan dalam penggunaannya.

Kata kunci: alumni, android, pengembangan, *tracer study*.

KATA PENGANTAR

Alhamdulillahirabbil‘alamin penulis panjatkan rasa syukur hanyalah kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah dan inayah-Nya. Shalawat serta salam semoga senantiasa tercurah limpahkan kepada baginda Rasulullah Nabi Muhamad SAW beserta keluarga, sahabat, dan para ahli ulama, sehingga penulis dapat menyelesaikan skripsi dengan judul **“TRACER ALUMNI BERBASIS ANDROID STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI”**.

Penyusunan laporan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.

Tidak lupa penulis mengucapkan terimakasih yang sebesar-besarnya kepada Ibunda tercinta Omih Susilawati dan Ayahanda Andi Sukandi yang telah mengasuh, membesarkan, mendidik dan memberikan dorongan lahir batin serta do'a agar penulis dapat menyelesaikan pendidikan. Mudah-mudahan Allah SWT selalu melimpahkan rahmat dan karunia-Nya kepada kedua orang tua. Serta Bapak Yudi Herdiana, S.T., M.T. selaku Dosen Pembimbing Utama dan Bapak Ridwan, S.T., M.Kom. selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu untuk membimbing, mengarahkan, memberikan koreksi dan motivasi dalam penyusunan Skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, baik dari isi maupun penyajiannya. Namun demikian penulis telah berusaha semaksimal mungkin untuk dapat menyajikan skripsi ini dengan sebaik-baiknya.

Adanya wujud karya sederhana ini merupakan kebahagian serta kebanggaan bagi penulis, pikiran dan tenaga yang telah penulis curahkan akhirnya dapat terealisasikan, namun penulis sadari tidak akan ada Laporan Skripsi ini tanpa adanya bantuan semua pihak sehingga tersusunnya laporan ini. Maka untuk itulah, sudah sepantasnya penulis mengucapkan terimakasih kepada:

1. Allah SWT yang telah memberikan rahmat dan karunia-Nya dalam proses penggerjaan laporan skripsi ini.

2. Yudi Herdiana, S.T., M.T., selaku dekan di Fakultas Teknologi Informasi dan juga selaku dosen pembimbing utama yang telah membimbing dalam penulisan skripsi ini.
3. Yaya Suharya, S.Kom., M.T., selaku Ketua Prodi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Nurul Imamah, S.T., M.T., selaku pembimbing pendamping yang telah banyak meluangkan waktu untuk membantu dan membimbing penulis.
5. Yang terhormat Bapak dan Ibu Dosen Program Studi Teknik Informatika Fakultas Teknologi Informasi yang tidak dapat penulis sebutkan satu-persatu.
6. Teman-teman semester VII Tahun Angkatan 2015, atas dukungan dan bantuannya dalam menyusun laporan ini.
7. Keluarga di rumah yang telah *mensupport* dan memberikan do'a selama proses penggerjaan laporan.
8. Sri Rahayuningrum yang memberikan semangat, selalu ada untuk membantu dalam menyelesaikan skripsi ini serta do'a dan dukungannya.
9. Dolly Hambali, S.T., selaku Kepala Bagian *EDP* (*electronic data processing*) di PT Multi Star Rukun Abadi (*Sharon Bakery*), yang tiada henti memberi dukungan dan motivasi kepada penulis.

Penulis menyadari bahwa skripsi ini mungkin terdapat kesalahan, baik dari segi penyusunan, tata bahasa maupun data-data yang dilaporkan. Oleh karena itu penulis memohon saran dan kritik yang membangun guna melengkapi dan menyempurnakan laporan ini.

Atas perhatian segala pihak yang telah membantu penulis dalam menyusun skripsi, penulis ucapkan terima kasih dan semoga skripsi ini dapat memberikan manfaat bagi penulis khususnya dan bagi semua umumnya.

Bandung, Agustus 2019

Deni Hidayat

DAFTAR ISI

| | |
|---|------|
| <i>ABSTRACT</i> | vi |
| ABSTRAK | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI | x |
| DAFTAR TABEL | xiii |
| DAFTAR GAMBAR..... | xv |
| DAFTAR LAMPIRAN..... | xx |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan..... | 3 |
| 1.5 Metodologi..... | 4 |
| 1.5.1 Metode Pengumpulan Data | 4 |
| 1.5.2 Metode Analisis dan Perancangan..... | 5 |
| 1.6 Sistematika Penulisan..... | 6 |
| BAB II TINJAUAN PUSTAKA | 7 |
| 2.1 Landasan Teori..... | 7 |
| 2.2 Dasar Teori | 14 |
| 2.2.1 <i>Tracer Study</i> | 14 |
| 2.2.2 Android | 15 |
| 2.2.3 <i>Google Play</i> | 16 |
| 2.2.4 <i>Software Development Kit (SDK)</i> | 16 |
| 2.2.5 <i>Java Development Kit (JDK)</i> | 17 |
| 2.2.6 <i>Gradle</i> | 17 |
| 2.2.7 <i>Manifes Aplikasi.....</i> | 18 |

| | | |
|--------|---|-----------|
| 2.2.8 | <i>Activity</i> | 19 |
| 2.2.9 | <i>Kotlin</i> | 20 |
| 2.2.10 | <i>Firebase</i> | 21 |
| 2.2.11 | <i>NoSQL</i> | 22 |
| 2.2.12 | <i>Xml</i> | 22 |
| 2.2.13 | <i>Json</i> | 23 |
| 2.2.14 | <i>Android Studio</i> | 27 |
| 2.2.15 | <i>Genymotion</i> | 28 |
| 2.2.16 | <i>Postman</i> | 28 |
| 2.2.17 | <i>Balsamiq Mockup</i> | 28 |
| 2.2.18 | <i>Start UML</i> | 29 |
| 2.2.19 | Tipe <i>Diagram UML</i> | 30 |
| 2.2.20 | <i>Model Driven Development (MDD)</i> | 32 |
| 2.2.21 | <i>Use Case Diagram</i> | 33 |
| 2.2.22 | <i>Activity Diagram</i> | 35 |
| 2.2.23 | <i>Sequence Diagram</i> | 36 |
| 2.2.24 | <i>Class Diagram</i> | 37 |
| | BAB III METODOLOGI PENELITIAN | 39 |
| 3.1 | Kerangka Pikir | 39 |
| 3.2 | Deskripsi..... | 40 |
| 3.2.1 | Investigasi | 40 |
| 3.2.2 | Analisis Masalah | 41 |
| 3.2.3 | Analisis Kebutuhan..... | 41 |
| 3.2.4 | Desain | 43 |
| 3.2.5 | Konstruksi | 45 |
| 3.2.6 | Pengujian..... | 47 |

| | | |
|--|----------------------------------|------------|
| 3.2.7 | Pelaporan..... | 47 |
| BAB IV ANALISIS DAN PERANCANGAN | | 49 |
| 4.1 | Analisis | 49 |
| 4.1.1 | Analisis Masalah | 49 |
| 4.1.2 | Analisis Kebutuhan..... | 50 |
| 4.2 | Perancangan..... | 56 |
| 4.2.1 | <i>Use Case Diagram</i> | 56 |
| 4.2.2 | <i>Activity Diagram</i> | 63 |
| 4.2.3 | <i>Sequence Diagram</i> | 77 |
| 4.2.4 | <i>Class Diagram</i> | 101 |
| 4.2.5 | Desain | 105 |
| BAB V IMPLEMENTASI DAN PENGUJIAN..... | | 130 |
| 5.1 | Implementasi..... | 130 |
| 5.1.1 | Lingkungan Implementasi..... | 130 |
| 5.1.2 | Implementasi Sistem..... | 131 |
| 5.2 | Pengujian | 162 |
| 5.2.1 | Rencana Pengujian | 162 |
| 5.2.2 | Kasus dan Hasil Pengujian..... | 167 |
| 5.2.3 | Kesimpulan Hasil Pengujian | 178 |
| BAB VI KESIMPULAN DAN SARAN | | 179 |
| 6.1 | Kesimpulan | 179 |
| 6.2 | Saran..... | 179 |
| DAFTAR PUSTAKA | | 181 |
| LAMPIRAN | | 183 |

DAFTAR TABEL

| | |
|---|-----|
| Tabel 2.1 Komparasi Review Penelitian..... | 9 |
| Tabel 2.2 Perbandingan Penelitian Terdahulu dengan Sekarang | 12 |
| Tabel 2.3 Versi Sistem Operasi Android | 15 |
| Tabel 2.4 Tipe <i>Diagram UML</i> | 31 |
| Tabel 2.5 Simbol <i>Use Case Diagram</i> | 33 |
| Tabel 2.6 Simbol <i>Activity Diagram</i> | 35 |
| Tabel 2.7 Simbol <i>Sequence Diagram</i> | 36 |
| Tabel 2.8 Simbol <i>Class Diagram</i> | 37 |
| Tabel 4.1 Waktu pengisian <i>tracer study</i> | 50 |
| Tabel 4.2 Analisis <i>Hardware</i> | 50 |
| Tabel 4.3 Analisis <i>software library android dependencies</i> | 51 |
| Tabel 4.4 Analisis Software | 53 |
| Tabel 4.5 Klasifikasi Pengguna pada aplikasi..... | 54 |
| Tabel 4.6 Analisis Biaya | 56 |
| Tabel 4.7 Skenario <i>Use Case Administrator – Admin</i> | 59 |
| Tabel 4.8 Skenario <i>Use Case Registrasi – Alumni</i> | 60 |
| Tabel 4.9 <i>Use Case</i> Kuesioner – Alumni..... | 61 |
| Tabel 4.10 Skenario <i>Use Case Login – Alumni</i> | 62 |
| Tabel 4.11 Tabel alumni | 102 |
| Tabel 4.12 Tabel chat..... | 102 |
| Tabel 4.13 Tabel fakultas | 103 |
| Tabel 4.14 Tabel flp..... | 103 |
| Tabel 4.15 Tabel tracerstudy | 104 |
| Tabel 4.16 Tabel users | 104 |
| Tabel 4.17 Tabel users_register..... | 105 |
| Tabel 5.1 Spesifikasi <i>Hardware</i> | 130 |
| Tabel 5.2 Spesifikasi <i>Software</i> | 131 |
| Tabel 5.3 Implementasi Tabel alumni | 133 |
| Tabel 5.4 Implementasi Tabel chat..... | 134 |
| Tabel 5.5 Implementasi Tabel fakultas | 135 |

| | |
|---|-----|
| Tabel 5.6 Implementasi Tabel flp | 135 |
| Tabel 5.7 Implementasi Tabel tracerstudy | 136 |
| Tabel 5.8 Implementasi Tabel users | 138 |
| Tabel 5.9 Implementasi Tabel users_register..... | 139 |
| Tabel 5.10 Rencana Pengujian Fungsionalitas Sistem | 162 |
| Tabel 5.11 Rencana Pengujian <i>Interface</i> Sistem..... | 164 |
| Tabel 5.12 Hasil Pengujian Fungsionalitas Sistem | 168 |
| Tabel 5.13 Hasil Pengujian <i>Inteface</i> Sistem..... | 171 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 <i>Json Object</i> | 24 |
| Gambar 2.2 <i>Json Array</i> | 24 |
| Gambar 2.3 <i>Json Value</i> | 25 |
| Gambar 2.4 <i>Json String</i> | 25 |
| Gambar 2.5 <i>Json Number</i> | 26 |
| Gambar 2.6 <i>Json Whitespace</i> | 26 |
| Gambar 2.7 <i>Balsamic Mockup</i> | 29 |
| Gambar 2.8 <i>Model Driven Development (MDD) Route</i> | 32 |
| Gambar 3.1 Gambar Kerangka Pikir | 39 |
| Gambar 4.1 <i>Use Case Diagram</i> | 57 |
| Gambar 4.2 Skenario <i>Use Case Login - Admin</i> | 58 |
| Gambar 4.3 <i>Activity Diagram</i> Alumni Registrasi | 63 |
| Gambar 4.4 <i>Activity Diagram</i> Alumni <i>Login</i> | 64 |
| Gambar 4.5 <i>Activity Diagram</i> Alumni <i>Reset Password</i> | 64 |
| Gambar 4.6 <i>Activity Diagram</i> Alumni Kuesioner | 65 |
| Gambar 4.7 <i>Activity Diagram</i> Alumni <i>Dashboard</i> | 65 |
| Gambar 4.8 <i>Activity Diagram</i> Alumni <i>Chat</i> | 66 |
| Gambar 4.9 <i>Activity Diagram</i> Alumni Angkatan..... | 66 |
| Gambar 4.10 <i>Activity Diagram</i> Alumni Tempat Bekerja Alumni | 67 |
| Gambar 4.11 <i>Activity Diagram</i> Alumni Forum..... | 67 |
| Gambar 4.12 <i>Activity Diagram</i> Alumni Loker | 68 |
| Gambar 4.13 <i>Activity Diagram</i> Alumni Lokasi | 68 |
| Gambar 4.14 <i>Activity Diagram</i> Alumni Profil | 69 |
| Gambar 4.15 <i>Activity Diagram</i> Admin <i>Login</i> | 69 |
| Gambar 4.16 <i>Activity Diagram</i> Admin <i>Reset Password</i> | 70 |
| Gambar 4.17 <i>Activity Diagram</i> Admin <i>Dashboard</i> | 70 |
| Gambar 4.18 <i>Activity Diagram</i> Admin <i>Chat</i> | 71 |
| Gambar 4.19 <i>Activity Diagram</i> Admin Angkatan..... | 71 |
| Gambar 4.20 <i>Activity Diagram</i> Admin Tempat Bekerja Alumni..... | 72 |
| Gambar 4.21 <i>Activity Diagram</i> Admin Forum..... | 72 |

| | |
|--|----|
| Gambar 4.22 <i>Activity Diagram</i> Admin Loker..... | 73 |
| Gambar 4.23 <i>Activity Diagram</i> Admin Pengumuman..... | 73 |
| Gambar 4.24 <i>Activity Diagram</i> Admin Lokasi | 74 |
| Gambar 4.25 <i>Activity Diagram</i> Admin Profil | 74 |
| Gambar 4.26 <i>Activity Diagram</i> Pengelolaan user | 75 |
| Gambar 4.27 <i>Activity Diagram</i> Data Alumni..... | 75 |
| Gambar 4.28 <i>Activity Diagram</i> Rekap <i>Tracer</i> Alumni..... | 76 |
| Gambar 4.29 <i>Activity Diagram</i> Fakultas..... | 76 |
| Gambar 4.30 <i>Activity Diagram</i> Database | 77 |
| Gambar 4.31 <i>Sequence Diagram</i> Alumni Registrasi..... | 77 |
| Gambar 4.32 <i>Sequence Diagram</i> Alumni Login | 78 |
| Gambar 4.33 <i>Sequence Diagram</i> Alumni Reset Password..... | 78 |
| Gambar 4.34 <i>Sequence Diagram</i> Alumni Kuesioner | 79 |
| Gambar 4.35 <i>Sequence Diagram</i> Alumni Dashboard..... | 80 |
| Gambar 4.36 <i>Sequence Diagram</i> Alumni Chat..... | 81 |
| Gambar 4.37 <i>Sequence Diagram</i> Alumni Informasi Angkatan | 82 |
| Gambar 4.38 <i>Sequence Diagram</i> Alumni Informasi Tempat Bekerja..... | 82 |
| Gambar 4.39 <i>Sequence Diagram</i> Alumni Informasi Forum..... | 83 |
| Gambar 4.40 <i>Sequence Diagram</i> Alumni Informasi Loker | 84 |
| Gambar 4.41 <i>Sequence Diagram</i> Alumni Lokasi..... | 85 |
| Gambar 4.42 <i>Sequence Diagram</i> Alumni Profil | 86 |
| Gambar 4.43 <i>Sequence Diagram</i> Admin Login | 87 |
| Gambar 4.44 <i>Sequence Diagram</i> Admin Reset Password..... | 87 |
| Gambar 4.45 <i>Sequence Diagram</i> Admin Dashboard..... | 88 |
| Gambar 4.46 <i>Sequence Diagram</i> Admin Chat..... | 89 |
| Gambar 4.47 <i>Sequence Diagram</i> Admin Informasi Angkatan | 90 |
| Gambar 4.48 <i>Sequence Diagram</i> Admin Tempat Bekerja Alumni..... | 90 |
| Gambar 4.49 <i>Sequence Diagram</i> Admin Informasi Forum | 91 |
| Gambar 4.50 <i>Sequence Diagram</i> Admin Informasi Loker | 92 |
| Gambar 4.51 <i>Sequence Diagram</i> Admin Informasi Pengumuman | 94 |
| Gambar 4.52 <i>Sequence Diagram</i> Admin Lokasi..... | 95 |
| Gambar 4.53 <i>Sequence Diagram</i> Admin Profil | 96 |

| | |
|---|-----|
| Gambar 4.54 <i>Sequence Diagram</i> Admin Pengelolaan User | 96 |
| Gambar 4.55 <i>Sequence Diagram</i> Admin Data Alumni | 97 |
| Gambar 4.56 <i>Sequence Diagram</i> Admin Rekap <i>Tracer</i> Alumni | 98 |
| Gambar 4.57 <i>Sequence Diagram</i> Admin Fakultas | 99 |
| Gambar 4.58 <i>Sequence Diagram</i> Admin Database | 100 |
| Gambar 4.59 <i>Class Diagram</i> | 101 |
| Gambar 4.60 <i>Diagram model</i> admin | 108 |
| Gambar 4.61 <i>Diagram model</i> alumni | 110 |
| Gambar 4.62 <i>User Interface</i> Halaman Gagal Koneksi Internet | 111 |
| Gambar 4.63 <i>User Interface</i> Halaman Registrasi..... | 112 |
| Gambar 4.64 <i>User Interface</i> Halaman Login..... | 112 |
| Gambar 4.65 <i>User Interface</i> Halaman Reset Password..... | 113 |
| Gambar 4.66 <i>User Interface</i> Halaman Registrasi Menunggu Validasi | 114 |
| Gambar 4.67 User Interface Halaman Registrasi Gagal Validasi | 114 |
| Gambar 4.68 Halaman Registrasi Berhasil Validasi | 115 |
| Gambar 4.69 Halaman Kuesioner Menu..... | 116 |
| Gambar 4.70 Halaman Kuesioner Data Pribadi | 116 |
| Gambar 4.71 Halaman Kuesioner Pengalaman Metode Pembelajaran | 117 |
| Gambar 4.72 Halaman Kuesioner Kepuasan Pengguna Lulusan..... | 117 |
| Gambar 4.73 Halaman Kuesioner Riwayat Pekerjaan..... | 117 |
| Gambar 4.74 Halaman Dashboard..... | 118 |
| Gambar 4.75 Halaman Chat | 119 |
| Gambar 4.76 Halaman Chat Kirim..... | 119 |
| Gambar 4.77 Halaman Informasi Angkatan | 120 |
| Gambar 4.78 Halaman Informasi Tempat Bekerja Alumni | 120 |
| Gambar 4.79 Halaman Informasi FLP..... | 121 |
| Gambar 4.80 Halaman Informasi Form FLP | 121 |
| Gambar 4.81 Halaman Lokasi..... | 122 |
| Gambar 4.82 Halaman Profil..... | 122 |
| Gambar 4.83 Halaman Profil Lihat Gambar | 123 |
| Gambar 4.84 Halaman Administrator..... | 123 |
| Gambar 4.85 Halaman Pengelolaan <i>User Register</i> | 124 |

| | |
|--|-----|
| Gambar 4.86 Halaman Pengelolaan <i>User Users</i> | 124 |
| Gambar 4.87 Halaman Data Alumni Angkatan | 125 |
| Gambar 4.88 Halaman Data Alumni Tempat Bekerja..... | 125 |
| Gambar 4.89 Halaman Data Alumni List Alumni..... | 126 |
| Gambar 4.90 Halaman Pelaporan Rekap Tracer Alumni | 126 |
| Gambar 4.91 Halaman Fakultas | 127 |
| Gambar 4.92 Halaman Form Fakultas | 127 |
| Gambar 4.93 Halaman Form Fakultas Program Studi..... | 128 |
| Gambar 4.94 Halaman <i>Database Backup</i> | 128 |
| Gambar 4.95 Halaman <i>Database Restore</i> | 129 |
| Gambar 5.1 <i>Firebase Konfiguasi Service</i> | 132 |
| Gambar 5.2 <i>Firebase Database</i> | 132 |
| Gambar 5.3 Halaman Gagal Koneksi Internet | 139 |
| Gambar 5.4 Halaman Registrasi Aplikasi..... | 140 |
| Gambar 5.5 Halaman <i>Login</i> Aplikasi | 140 |
| Gambar 5.6 Halaman <i>Forgot Password</i> Aplikasi | 141 |
| Gambar 5.7 Halaman Menunggu Validasi..... | 141 |
| Gambar 5.8 Halaman Gagal Validasi | 142 |
| Gambar 5.9 Halaman Berhasil Validasi Aplikasi..... | 142 |
| Gambar 5.10 Halaman Kuesioner Menu Aplikasi..... | 143 |
| Gambar 5.11 Halaman Kuesioner Data Pribadi Aplikasi | 143 |
| Gambar 5.12 Halaman Kuesioner Pengalaman Pembelajaran Aplikasi.... | 144 |
| Gambar 5.13 Halaman Kepuasan Pengguna Lulusan Aplikasi..... | 144 |
| Gambar 5.14 Halaman Riwayat Pekerjaan Aplikasi | 145 |
| Gambar 5.15 Halaman Dashboard Admin Aplikasi | 145 |
| Gambar 5.16 Halaman Dashboard User Aplikasi | 146 |
| Gambar 5.17 Halaman Chat List Aplikasi | 146 |
| Gambar 5.18 Halaman Chat Kirim Aplikasi | 147 |
| Gambar 5.19 Halaman Informasi Angkatan Aplikasi | 147 |
| Gambar 5.20 Halaman Informasi Tempat Bekerja Alumni Aplikasi | 148 |
| Gambar 5.21 Halaman Informasi Forum Aplikasi | 148 |
| Gambar 5.22 Halaman Informasi Loker Aplikasi | 148 |

| | |
|---|-----|
| Gambar 5.23 Halaman Informasi Pengumuman Aplikasi | 149 |
| Gambar 5.24 Halaman Lokasi Aplikasi..... | 149 |
| Gambar 5.25 Halaman Lokasi Pencarian Aplikasi..... | 150 |
| Gambar 5.26 Halaman Profil Aplikasi..... | 150 |
| Gambar 5.27 Halaman Profil Item Menu Aplikasi..... | 151 |
| Gambar 5.28 Halaman Ubah <i>Password</i> Aplikasi..... | 151 |
| Gambar 5.29 Halaman <i>Administrator</i> Aplikasi..... | 152 |
| Gambar 5.30 Halaman Pengelolaan <i>User List Registrasi</i> Aplikasi | 152 |
| Gambar 5.31 Halaman Pengelolaan <i>User List Users</i> Aplikasi | 153 |
| Gambar 5.32 Halaman Data Alumni Angkatan Aplikasi | 153 |
| Gambar 5.33 Halaman Data Alumni Tempat Bekerja Aplikasi..... | 154 |
| Gambar 5.34 Halaman Rekap <i>Tracer</i> Alumni Aplikasi | 154 |
| Gambar 5.35 Halaman Rekap <i>Tracer</i> Alumni <i>Export</i> Aplikasi | 155 |
| Gambar 5.36 Halaman Fakultas Aplikasi | 155 |
| Gambar 5.37 Halaman Fakultas Aksi Tambah Aplikasi | 156 |
| Gambar 5.38 Halaman Fakultas Aksi Tambah Program Studi Aplikasi ... | 156 |
| Gambar 5.39 <i>Database Backup</i> Aplikasi..... | 157 |
| Gambar 5.40 <i>Database Restore</i> Aplikasi..... | 157 |
| Gambar 5.41 Halaman <i>Google Play Console Dashboard</i> | 158 |
| Gambar 5.42 Halaman <i>Google Play Console Create Application</i> | 158 |
| Gambar 5.43 Halaman <i>Google Play Console Panel</i> | 159 |
| Gambar 5.44 Halaman <i>Google Play Console App Releases</i> | 159 |
| Gambar 5.45 Halaman <i>Google Play Console Store Listing</i> | 160 |
| Gambar 5.46 Halaman <i>Google Play Console Content Rating</i> | 160 |
| Gambar 5.47 Halaman <i>Google Play Console App Content</i> | 161 |
| Gambar 5.48 Halaman <i>Google Play Console Pricing & Distribution</i> | 161 |
| Gambar 5.49 Halaman <i>Google Play Publish</i> | 162 |

DAFTAR LAMPIRAN

| | |
|---|-----|
| Lampiran 1 <i>Form</i> Wawancara..... | 183 |
| Lampiran 2 <i>Testing</i> Program..... | 185 |
| Lampiran 3 <i>Listing</i> Program | 190 |
| Lampiran 4 Kartu Bimbingan..... | 255 |
| Lampiran 5 Biodata Penulis | 256 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tracer Study adalah sebuah layanan yang disediakan oleh setiap lembaga pendidikan untuk melacak alumni yang telah lulus dari lembaganya yang mana nantinya dapat digunakan untuk akreditasi dan pendataan para lulusan yang ada bertujuan untuk mengetahui *outcome* pendidikan dalam bentuk transisi dari dunia pendidikan ke dunia kerja. Mendapatkan informasi tersebut dilaksanakan sebuah kegiatan survey alumni yang dinamakan kegiatan *Tracer Study Alumni* (TSA). Menurut (Schomburg, 2003), (Millington & College, 2006) dan (Muzakki, 2016) kegiatan yang dilakukan oleh tim TSA adalah melakukan survey untuk mendapatkan informasi dari alumni dengan menggunakan sistem *Questionnaire online*, informasi yang diperoleh dari *Questionnaire online* adalah seputar karakteristik dan sosiologi sebelum kuliah, kegiatan pendidikan dan pembelajaran dikampus, transisi kedunia kerja, pekerjaan dan kompetensi hubungan pekerjaan dengan program studi. Setelah informasi atau data diperoleh, selanjutnya data akan diolah oleh tim olah data. Disamping itu *tracer study* juga berguna untuk mengetahui hubungan antara perguruan tinggi dengan dunia kerja profesional dan pemangku kepentingan, serta merupakan salah satu kelengkapan syarat untuk akreditasi perguruan tinggi (Yadi & Susihono, 2013), yaitu standar ketiga dari ketujuh standar dalam Standar Akreditasi Program Sarjana yakni Mahasiswa dan Lulusan, sehingga untuk poin lulusan dapat ditelusuri dengan kegiatan *tracer study* (Hakim, 2017).

Tracer alumni yang dilakukan Fakultas Teknologi Informasi Universitas Bale Bandung (FTI UNIBBA) saat ini masih menggunakan layanan google form berbasis web dengan link akses yang sudah disediakan <http://unibba.ac.id/fti/mahasiswa/alumni> sehingga alumni dapat mengakses ke alamat web tersebut untuk mengisi kuisioner sebagai masukan penunjang pelaksanaan program penyelarasan pendidikan dengan dunia kerja atau yang disebut dengan aspek relevansi. Pada aspek relevansi ini, perguruan tinggi dituntut

mampu menghasilkan lulusan yang memiliki daya saing yang siap berkiprah dalam pembangunan. Salah satu cara untuk menggali informasi berkaitan dengan transisi dari kuliah ke pekerjaan adalah dengan melaksanakan suatu studi yang disebut sebagai *tracer study*.

Beberapa hal menjadi kendala dalam pelaksanaan penelitian *tracer study* FTI UNIBBA dalam pengumpulan data tersebut memakai layanan aplikasi pada web google form yang masih memiliki keterbatasan dalam fitur maupun fungsinya, banyaknya berkas yang harus dibuat kemudian diolah lagi oleh dosen, berkas yang berupa kertas sangat rawan bila rusak dan kotor, serta apabila lewat e-mail tidak semua lulusan merespon dengan cepat, begitu pula bila melewati telepon yang terkadang terkendala dengan biaya dan repot bila harus menggunakan wawancara dengan telepon, hingga terkadang bila memakai tenaga pos, sering lulusan tidak mengirim kembali data *tracer study* kuisioner yang dibutuhkan dan untuk kebagian akademik belum terintegrasi, sehingga dalam penyajian laporan data tersebut memerlukan waktu validasi untuk diolah kembali.

Berdasarkan fakta di atas, maka dibuatlah sarana untuk mendapatkan data lulusan dengan menggunakan aplikasi *mobile* dengan database menggunakan firebase. Selain kemudahan untuk mengakses *smartphone* oleh semua kalangan, penggunaan sarana aplikasi *mobile* untuk mengumpulkan data *tracer study* akan lebih mudah, lebih efisien, efektif, murah, dan pengelolaan data akan lebih mudah dibandingkan dengan sistem yang sedang berjalan saat ini dan pengorganisasianya masih terpisah. Penggunaan sistem informasi *tracer study* dengan aplikasi *mobile* akan membuat peneliti lebih mudah dalam mengolah data dari alumni. Tak hanya akan membantu para peneliti atau dosen yang ingin melakukan *tracer study*, tapi juga untuk para lulusan tentu akan lebih mudah dalam pengisian data *tracer study*, komunikasi antar alumni dan fakultas.

Pendapat diatas dapat disimpulkan bahwa *tracer study* adalah studi penelusuran yang dilakukan terhadap alumni yang telah lulus dan bekerja. Kegiatan penelusuran yang dimaksud adalah mengenai masa tunggu kerja, jenis pekerjaan dan pendapatan awal alumni. Berdasarkan uraian di atas penulis tertarik untuk mengambil penelitian yang berjudul “*Tracer Alumni Berbasis Android Studi Kasus Di Fakultas Teknologi Informasi*”.

1.2 Rumusan Masalah

Pada paparan latar belakang di atas, terdapat beberapa rumusan masalah yang terlihat adalah sebagai berikut:

1. Bagaimana membuat sistem untuk mendapatkan data lulusan dengan menggunakan aplikasi *mobile* berbasis android?
2. Bagaimana sistem tracer alumni dapat terintegrasi ke bagian akademik?
3. Bagaimana membuat media informasi yang mudah di akses untuk melihat tracer alumni?

1.3 Batasan Masalah

Penelitian ini akan dilakukan dengan terfokus pada penyajian informasi mengenai alumni dan dapat sesuai dengan tujuan utama. Batasan masalah adalah sebagai berikut:

1. Sistem informasi tracer study ini dikhkususkan hanya untuk alumni FTI UNIBBA.
2. Aplikasi ini menggunakan format kuesioner yang disusun atas persetujuan dari FTI UNIBBA.
3. Menggunakan database Firebase versi free sehingga dibatasi dengan 100 user dan memiliki storage 1 gb.
4. Laporan yang akan dihasilkan dalam aplikasi ini adalah laporan identitas alumni, laporan kegiatan saat ini, laporan informasi lowongan pekerjaan, laporan umpan balik proses pembelajaran, laporan penilaian kinerja alumni, dan laporan perbandingan *tracer study* alumni.

1.4 Tujuan

1. Mengetahui penyerapan, proses, dan posisi lulusan FTI UNIBBA dalam dunia kerja.
2. Memudahkan untuk akses kegiatan *tracer* alumni yang berupa aplikasi online berbasis android yang di tunjukan untuk para alumni FTI UNIBBA.

3. Mendukung proses manajemen untuk memberikan informasi yang lengkap dan akurat mengenai data alumni khususnya alumni FTI UNIBBA.

1.5 Metodologi

Metodologi yang digunakan dalam penelitian ini untuk mendapat data penelitian, digunakan metode pengumpulan data dan metode analisis adalah sebagai berikut:

1.5.1 Metode Pengumpulan Data

Penelitian ini dilangsungkan pada objek penelitian. Dalam hal ini data yang penulis kumpulkan terdiri dari data primer dan sekunder, yaitu :

1. Data primer

Mengumpulkan data secara langsung dari objek yang diteliti. Adapun cara-cara yang dipakai untuk mengumpulkan data tersebut adalah sebagai berikut:

- a. Metode Wawancara (Interview)

Penulis mengadakan tanya-jawab dan dialog langsung dengan pegawai pada bagian kemahasiswaan FTI UNIBBA. Wawancara dilakukan untuk mencari atau menemukan informasi terhadap proses yang dilakukan dalam kegiatan *tracer* alumni.

- b. Metode Dokumentasi

Penulis melakukan pengumpulan data atau dokumen petunjuk pelaksanaan kegiatan *tracer* alumni. Bentuk dokumen yang dikumpulkan dapat berupa peraturan pemerintah maupun peraturan FTI UNIBBA.

- c. Metode Observasi

Pada metode ini penulis melakukan pengamatan langsung terhadap objek yang akan diteliti yang bertujuan untuk memperkuat data, mengetahui serta mendapatkan informasi secara langsung. Bagaimana sistem yang sedang berjalan pada kegiatan *tracer* alumni di FTI UNIBBA.

2. Data Sekunder

Data sekunder yang diperoleh dan digunakan berupa pengetahuan teoritis yang penulis peroleh selama ini, baik dari bahan – bahan buku yang penulis dapat dari beberapa sumber maupun media internet yang berhubungan dengan penelitian ini. Pencarian lewat media internet dilakukan dengan cara melakukan pencarian ke situs – situs atau jurnal – jurnal untuk kelengkapan data penelitian yang berhubungan dengan judul penelitian yang penulis buat.

1.5.2 Metode Analisis dan Perancangan

1. Metode Analisis

Pada tahap Metode Analisis menggunakan metode MDD *Model Driven Development* adalah sebagai berikut :

- a. Identifikasi Masalah Tahap awal analisis yang memiliki tujuan untuk menjawab semua pertanyaan yang ada dan melakukan identifikasi resiko yang mungkin terjadi dalam proses pengembangan sistem.
- b. Analisis Masalah Tahap dimana peneliti melakukan pemahaman terhadap permasalahan yang muncul dalam perusahaan agar dapat menarik akar permasalahan dari sistem yang sedang digunakan.
- c. Analisis Kebutuhan Sistem Untuk mengetahui kebutuhan sistem pada perusahaan, maka perlu dilakukan identifikasi input, proses, dan output dalam perusahaan agar pengembangan sistem yang baru dapat menutupi kekurangan sistem yang sudah ada.

2. Perancangan

Pada tahap perancangan sistem dibuat dengan menggunakan *Unified Modeling Language (UML)*, merupakan sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, membangun dan mendokumentasikan dari sebuah sistem pembangunan perangkat lunak berbasis objek, diantaranya menggunakan *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram* dan *flowmap*.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini terdiri dari enam bab adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang informasi umum yaitu latar belakang penelitian, perumusan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, waktu dan tempat penelitian, metodologi penelitian, dan sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini memuat tentang penelitian terkait sebelumnya dan juga dasar teori yang berfungsi sebagai sumber atau alat dalam memahami permasalahan yang berkaitan dengan konsep dasar sistem informasi, dan tentang sistem informasi *tracer study*.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang pengembangan metodologi yang terdiri dari kerangka pemikiran dan deskripsi.

BAB IV ANALISIS PERANCANGAN DAN HASIL

Bab ini menjelaskan analisis sistem yang sedang berjalan, menjelaskan perancangan sistem yang akan dibuat seperti perancangan struktur proses, perancangan basis data, dan perancangan aplikasi yang akan dibuat, serta berisi hasil dari pengujian aplikasi yang telah dibuat.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dan pengujian serta hasil pengamatan secara keseluruhan terhadap aplikasi yang akan dibangun.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang memuat tentang jawaban dari pada rumusan masalah pada penelitian. Saran yang ditujukan kepada para pengguna hasil penelitian yang bersangkutan dan kepada peneliti selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori merupakan teori yang relevan yang digunakan untuk menjelaskan tentang variabel yang akan diteliti dan sebagai dasar untuk memberi jawaban sementara terhadap rumusan masalah yang diajukan (hipotesis), dan penyusunan instrument penelitian. Teori yang digunakan bukan sekedar pendapat dari pengarang atau pendapat lain, tetapi teori yang benar-benar telah teruji kebenarannya. Berikut adalah beberapa jurnal penelitian yang telah dilakukan sebelumnya terkait dengan penelitian yang saat ini dilakukan:

1. Jurnal oleh (Lukman H, M. Ade O, 2017) dengan judul “Perancangan Sistem *Tracer Alumni* Pada Stmik Musi Rawas Berbasis Web Mobile”. Vol. 2, No.2.

Penelitian ini membuat sebuah sarana untuk mendapatkan data lulusan dengan menggunakan teknologi website dalam hal ini selain kemudahan untuk mengakses website oleh semua kalangan, penggunaan sarana *website* untuk mengumpulkan data *tracer study* akan lebih mudah, lebih efisien, efektif, murah, dan pengelolaan data *tracer study* akan lebih mudah dibandingkan dengan sistem yang masih manual dan pengorganisasianya masih terpisah. Metode pengumpulan data yang digunakan adalah observasi, wawancara, dan dokumentasi sedangkan metode pengembangan sistem yang di pergunakan adalah *Prototype* dan *UML* sebagai bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk mencapai visi dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan dengan yang lain. Penggunaan sistem informasi *tracer study* akan membuat pihak kampus lebih mudah dalam mengakses dan mengolah data dari lulusan untuk dijadikan data matang yang siap di gunakan untuk berbagai keperluan. Berdasarkan hasil penelitian dan analisa serta perancangan aplikasi yang telah dihasilkan, maka dapat ditarik kesimpulan bahwa Penelitian ini menghasilkan suatu sistem *tracer study* berbasis *web mobile* yang dapat diakses melalui perangkat

komputer ataupun *smartphone* dan dengan adanya sistem *tracer study* alumni berbasis *website* maka, kegiatan *tracer study* dapat dilakukan lebih mudah karena proses pengambilan data ke alumni dapat dilakukan secara global dan tidak terbatas waktu.

2. Jurnal oleh (Giat K, M Zaenudin, 2013) dengan judul “Informasi *Tracer Study* Pada Stmik Amikom Purwokerto”. Vol. 6, No.1.

Salah satu faktor keberhasilan proses pendidikan di perguruan tinggi adalah menghasilkan lulusan yang berkualitas, daya saing lulusan dalam dunia kerja menjadi tolak ukur keberhasilan suatu perguruan tinggi, daya saing lulusan dapat ditunjukkan melalui profil pekerjaan, gaji yang diperoleh dan manfaat mata kuliah dalam pekerjaan. Belum tersedianya sistem informasi mengenai data alumni menyulitkan pihak prodi maupun para pemangku kepentingan (*stakeholder*) untuk mengetahui informasi berkaitan dengan transisi alumni dari kuliah ke pekerjaan, dengan adanya *tracer study* diharapkan dapat menyediakan informasi mengenai hubungan antara pendidikan tinggi dan dunia kerja, menilai relevansi pendidikan tinggi, informasi bagi para pemangku kepentingan (*stakeholder*), atau untuk kepentingan evaluasi hasil pendidikan dan selanjutnya dapat digunakan untuk penyempurnaan dan penjaminan kualitas. Tujuan pada penelitian ini adalah membuat suatu sistem informasi *website tracer study* yang digunakan untuk informasi alumni. Metode pengumpulan data yang digunakan adalah observasi, wawancara, dan dokumentasi sedangkan metode pengembangan sistem yang digunakan dalam pembuatan sistem aplikasi ini adalah metode *SDLC* atau *waterfall*. Hasil dari penelitian ini adalah memberikan informasi tentang alumni dalam berkompetensi di dunia pekerjaan, dapat dijadikan masukan dalam perbaikan program Pendidikan khususnya di STMIK Amikom Purwokerto supaya menghasilkan lulusan yang berkualitas.

3. Jurnal oleh (Dilson, Noviardi, 2018) dengan judul “Perancangan Model dan Prototype Aplikasi *Tracer Study* Berbasis Mobile”.

Sulitnya Bidang Akademik dan Administrasi Kemahasiswaan dalam mengumpulkan data alumni Sekolah Tinggi Teknologi Payakumbuh yang sudah atau belum bekerja dan besarnya biaya yang dibutuhkan untuk

mendapatkan data alumni merupakan salah satu masalah yang dihadapi dalam pengisian *tracer study* dan untuk data penunjang akreditasi secara valid. Tujuan penelitian adalah merancang aplikasi *tracer study* yang dapat diakses secara *realtime* oleh alumni sebagai media untuk pengisian data *tracer study* berbasis *mobile* menggunakan aplikasi Android studio, dan *database SQLite* sebagai media penyimpanan data. Metode pengumpulan data yang digunakan adalah observasi, wawancara, dan dokumentasi sedangkan metode pengembangan sistem yang digunakan dalam penelitian adalah metode Spiral. Fokus penelitian adalah melakukan perancangan proses yang terdiri dari pembuatan arsitektur aplikasi, model *activity*, mobile *use case* diagram, dan mobile *class* diagram. Perancangan *interface* aplikasi untuk alumni terdiri dari halaman registrasi, login alumni, menu utama, edit data alumni dan pengisian kuisioner, sedangkan perancangan *interface* aplikasi untuk BAAK terdiri dari halaman login, menu dashboard, rekapitulasi tracer study dan *response rate*. Pengujian aplikasi secara alfa menggunakan *blackbox testing*. Hasil pengujian didapat beberapa fungsi dan modul maupun *interface* dapat berjalan dengan baik dan dapat mempermudah bagian BAAK dalam mengumpulkan data lulusan/alumni untuk melengkapi data yang dibutuhkan akreditasi institusi maupun program studi. Penelitian selanjutnya disarankan adanya penambahan fitur chat antar alumni, dan pengujian dapat dikembangkan secara beta.

Dari tiga penelitian yang telah direview, maka dibuatlah tabel perbandingannya pada tabel 2.1 dibawah.

Tabel 2.1 Komparasi Review Penelitian

| No | Perbandingan | Penelitian 1 | Penelitian 2 | Penelitian 3 |
|----|----------------------------|--------------------------|----------------------------|-------------------------------------|
| 1 | Pengguna | Stmik Musi Rawas | Stmik Amikom Purwokerto | Sekolah Tinggi Teknologi Payakumbuh |
| 2 | Metode Pengembangan sistem | <i>Prototype</i> dan UML | SDLC atau <i>Waterfall</i> | Metode Spiral |
| 3 | Metode Pengujian | Pengujian | Pengujian <i>black</i> | Pengujian <i>error</i> |

| | Perangkat Lunak | alfa | <i>box</i> | <i>guesting</i> |
|---|--------------------------|-------------|-------------|-----------------|
| 4 | Aktivasi user oleh admin | Ada | Tidak ada | Tidak ada |
| 5 | Kuesioner | Ada | Ada | Ada |
| 6 | Registrasi | Oleh alumni | Oleh alumni | Oleh alumni |
| 7 | Isi biodata | Oleh alumni | Oleh alumni | Oleh alumni |

Penelitian tiga jurnal diatas mengenai *tracer study* alumni terhadap sistem yang akan dikembangkan terdapat beberapa persamaan dan perbedaan dengan penelitian terdahulu. Adapun persamaan dan perbedaannya ialah:

1. Perbedaan dan Persamaan antara penelitian ini dengan penelitian yang dilakukan oleh Lukman Hakim dan M. Ade Oktariandi.

Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan oleh Lukman Hakim dan M. Ade Oktariandi yaitu sama-sama menggunakan kuesioner standar form *tracer study* 2017 dari riset dikti, memiliki 2 hak akses antara admin dengan User Alumni, Informasi Lowongan Pekerjaan, Pengumuman dan Laporan Rekapitulasi hasil *tracer study*.

Perbedaan antara keduanya terletak pada database, basis aplikasi, metode pengembangan sistem dan fitur fungsional yang dilakukan oleh Lukman Hakim dan M. Ade Oktariandi yaitu menggunakan database MySQL dikembangkan ke database NoSQL *firebase realtime*, menggunakan aplikasi berbasis web dikembangkan ke aplikasi berbasis android yang dapat diinstal di smartphone android sehingga alumni dapat dengan mudah dalam berbagi informasi, menggunakan metode pengembangan sistem *Prototype* dan UML dikembangkan ke metode pengembangan sistem MDD (*Model Driven Development*) , penambahan fitur fungsional (*chat*, lokasi/*map* posisi keberadaan alumni dan notifikasi atau pemberitahuan yang dapat diterima secara *realtime*). Lukman Hakim dan M. Ade Oktariandi melakukan penelitian di tahun 2017 dengan objek penelitian Perancangan Sistem *Tracer Alumni* Pada Stmik Musi Rawas Berbasis *Web Mobile* sedangkan penelitian ini dilakukan pada tahun 2019 dengan objek penelitian *Tracer Alumni* Berbasis Android Studi Kasus Di Fakultas Teknologi Informasi.

2. Perbedaan dan Persamaan antara penelitian ini dengan penelitian yang dilakukan oleh Giat Karyono dan Muhamad Zaenudin.

Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan oleh Giat Karyono dan Muhamad Zaenudin yaitu sama-sama menggunakan kuesioner standar form *tracer study* 2017 dari riset dikti, memiliki 2 hak akses antara admin dengan user alumni, *Dashboard* dan laporan bentuk grafik, Forum / Agenda, Informasi Lowongan Pekerjaan, Pengumuman dan laporan rekapitulasi hasil *tracer study*.

Perbedaan antara keduanya terletak pada database, basis aplikasi, metode pengembangan sistem dan fitur fungsional yang dilakukan oleh Giat Karyono dan Muhamad Zaenudin yaitu Menggunakan database MySQL dikembangkan ke database NoSQL *firebase realtime*, menggunakan aplikasi berbasis web dikembangkan ke aplikasi berbasis android yang dapat diinstal di *smartphone* android sehingga alumni dapat dengan mudah dalam berbagi informasi, menggunakan metode pengembangan sistem *SDLC* atau *Waterfall* dikembangkan ke metode pengembangan sistem *MDD (Model Driven Development)* dan penambahan fitur fungsional (chat, lokasi / map posisi keberadaan alumni dan notifikasi atau pemberitahuan yang dapat diterima secara *realtime*). Giat Karyono dan Muhamad Zaenudin melakukan penelitian di tahun 2013 dengan objek penelitian Informasi *Tracer Study* Pada Stmik Amikom Purwokerto sedangkan penelitian ini dilakukan pada tahun 2019 dengan objek penelitian *Tracer Alumni Berbasis Android Studi Kasus Di Fakultas Teknologi Informasi*.

3. Perbedaan dan persamaan antara penelitian ini dengan penelitian yang dilakukan oleh Dilson dan Noviardi.

Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan oleh Dilson dan Noviardi yaitu sama-sama menggunakan kuesioner standar form *tracer study* 2017 dari riset dikti, Menggunakan aplikasi berbasis android yang dapat diinstal di *smartphone* android sehingga alumni dapat dengan mudah dalam berbagi informasi, memiliki 2 hak akses antara admin dengan user alumni, *Dashboard* dan Laporan bentuk grafik, Forum / Agenda, Informasi

Lowongan Pekerjaan, Pengumuman dan Laporan Rekapitulasi hasil *tracer study*.

Perbedaan antara keduanya terletak pada database, metode pengembangan sistem dan fitur fungsional yang dilakukan oleh Dilson dan Noviardi yaitu menggunakan database SQLite dikembangkan ke database NoSQL *firebase realtime*, menggunakan Metode Spiral dikembangkan ke Metode *MDD (Model Driven Development)* dan Penambahan fitur fungsional (chat, lokasi / map posisi keberadaan alumni dan notifikasi atau pemberitahuan yang dapat diterima secara *realtime*). Dilson dan Noviardi melakukan penelitian di tahun 2018 dengan objek penelitian Perancangan Model dan *Prototype* Aplikasi *Tracer Study* Berbasis mobile sedangkan penelitian ini dilakukan pada tahun 2019 dengan objek penelitian *Tracer Alumni* Berbasis Android Studi Kasus Di Fakultas Teknologi Informasi.

Tabel 2.2 Perbandingan Penelitian Terdahulu dengan Sekarang

| No | Peneliti | Judul | Persamaan | Perbedaan |
|----|------------------------------------|--|---|--|
| 1 | Lukman Hakim dan M. Ade Oktariandi | Perancangan Sistem <i>Tracer Alumni</i> Pada Stmik Musi Rawas Berbasis <i>Web Mobile</i> | 1. Memakai kuesioner form <i>tracer study</i> 2017 standar riset dikti 2. Hak akses <i>Administrator</i> dan user alumni 3. Info Loker 4. Pengumuman 5. Rekapitulasi Hasil <i>tracer study</i> | 1. Aplikasi Berbasis Android 2. Database <i>Firebase NoSQL</i> 3. <i>Dashboard</i> dan laporan bentuk grafik 4. Fitur chat 5. Lokasi / Map Alumni 6. Notifikasi <i>realtime</i> |

| | | | | |
|---|-----------------------------------|--|---|--|
| 2 | Giat Karyono dan Muhamad Zaenudin | Informasi <i>Tracer Study</i> Pada Stmik Amikom Purwokerto | <ol style="list-style-type: none"> 1. Memakai kuesioner form <i>tracer study</i> 2017 standar riset dikt 2. Hak akses Administrator dan user alumni 3. <i>Dashboard</i> dan laporan bentuk grafik 4. Menu Forum / Agenda 5. Info Loker 6. Pengumuman 7. Rekapitulasi Hasil <i>tracer study</i> | <ol style="list-style-type: none"> 1. Aplikasi Berbasis Android 2. <i>Database Firebase Nosql</i> 3. Fitur chat 4. Lokasi / Map Alumni 5. Notifikasi <i>realtime</i> |
| 3 | Dilson dan Noviardi | Perancangan Model dan <i>Prototype</i> Aplikasi <i>Tracer Study</i> Berbasis <i>Mobile</i> | <ol style="list-style-type: none"> 1. Memakai kuesioner form <i>tracer study</i> 2017 standar riset dikt 2. Aplikasi Berbasis Android 3. Hak akses Administrator dan user alumni 4. Menu Forum 5. Menu Loker 6. Pengumuman 7. Rekapitulasi Hasil <i>tracer study</i> | <ol style="list-style-type: none"> 1. <i>Database Firebase Nosql</i> 2. <i>Dashboard</i> dan laporan bentuk grafik 3. Fitur chat 4. Lokasi / Map Alumni 5. Notifikasi <i>realtime</i> |

2.2 Dasar Teori

2.2.1 *Tracer Study*

Tracer study adalah studi pelacakan jejak lulusan/alumni yang dilakukan 2 (dua) tahun setelah lulus dan bertujuan untuk mengetahui outcome pendidikan dalam bentuk transisi dari dunia pendidikan tinggi ke dunia kerja, situasi kerja terakhir, keselarasan dan aplikasi kompetensi di dunia kerja. *Tracer study* juga dapat memberikan informasi mengenai *output* pendidikan yaitu penilaian diri terhadap penguasaan dan pemerolehan kompetensi, proses pendidikan berupa evaluasi proses pembelajaran dan kontribusi pendidikan tinggi terhadap pemerolehan kompetensi serta input pendidikan berupa penggalian lebih lanjut terhadap informasi sosiobiografis lulusan (Dr. & Sapto, n.d.).

Dasar hukum pelaksanaan *tracer study* tersebut telah ditulis dalam Surat Edaran Kementerian Riset, Teknologi, dan Pendidikan Tinggi Dirjen Pembelajaran dan Kemahasiswaan No. 313/B/SE/2016. Pada prinsipnya Dirjen Pembelajaran dan Kemahasiswaan meminta seluruh Pimpinan Perguruan Tinggi di Indonesia serta Koordinator Kopertis Wilayah I s.d XIV untuk melaksanakan pelacakan alumni / *tracer study*. Hal tersebut dilakukan untuk memperoleh data nasional yang akurat mengenai situasi kondisi hubungan pendidikan tinggi dengan dunia kerja, dan sistem pengembangan pusat karir perguruan tinggi. Dalam pelaksanaan *tracer study* tersebut perguruan tinggi menggunakan standar model, desain, metodologi, dan kuisioner. Sehingga dapat membuat kompilasi dan komparasi data antar fakultas / program studi tingkat nasional (dikti, n.d.).

Penerapan *tracer study* paling tepat dilakukan oleh Badan atau Pusat Pengembangan Karir Alumni dalam tingkat perguruan tinggi. Alasannya adalah mempunyai target yang sama yaitu alumni / lulusan baru (*fresh graduate*). Kedua untuk menghindari tumpah tindih keperluan pelayanan yang berhubungan dengan alumni dari berbagai fakultas atau program studi. Ketiga memiliki struktur fungsi / organisasi dasar hukum yang kuat, resmi, dan, kredible terkait persoalan alumni. Keempat dapat menjadi tempat yang digunakan untuk alokasi sumber daya bagi pelaksanaan *tracer study*. Kelima, dapat membuat suatu model / sistem yang sangat efisien dalam penggunaan sumber dayanya untuk membantu menangani

kebutuhan terkait alumni. Terakhir hasil yang diperoleh terkait *tracer study* terjamin perbandingan / komparabilitas data antar prodi / fakultas.

2.2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Aplikasi android merupakan sebuah perangkat lunak yang dirancang untuk dapat berjalan pada ponsel, komputer tablet dan perangkat *mobile* lainnya, dan merupakan aplikasi yang paling banyak digunakan oleh beberapa produsen *smartphone* ternama saat ini, karena secara *open source* dapat digunakan oleh pengembang aplikasi *mobile* untuk merancang berbagai macam aplikasi berbasis Android (Wikipedia, 2019).

Android merupakan sistem operasi yang terbuka (*open source*) yang mana berarti jika pihak *Google* memperbolehkan dan membebaskan bagi pihak manapun untuk dapat mengembangkan sistem operasi tersebut.

Pada perkembangannya, sistem operasi Android telah mengalami beberapa perubahan dan perbaikan dan yang paling menarik adalah versi keluaran Android yang diberi nama seperti nama-nama makanan. Berikut ini nama kode Android sejak versi 1.5 hingga 9.0.

Tabel 2.3 Versi Sistem Operasi Android

| No | Versi | Nama kode | Api Level | Tanggal rilis |
|----|-------------|---------------------------|-----------|-------------------|
| 1 | 1.5 | <i>Cupcake</i> | 1-3 | 30 April 2009 |
| 2 | 1.6 | <i>Donut</i> | 4 | 15 September 2009 |
| 3 | 2.0-2.1 | <i>Eclair</i> | 5-6 | 26 Agustus 2009 |
| 4 | 2.2 | <i>Froyo</i> | 7 | 20 Mei 2010 |
| 5 | 2.3.-2.3.2 | <i>Gingerbread</i> | 8-9 | 6 Desember 2010 |
| 6 | 2.3.3-2.3.7 | <i>Gingerbread</i> | 10-11 | 9 Februari 2011 |
| 7 | 3.1 | <i>Honeycomb</i> | 12 | 10 Mei 2011 |
| 8 | 3.2 | <i>Honeycomb</i> | 13 | 15 Juli 2011 |
| 9 | 4.0.3-4.0.4 | <i>Ice Cream Sandwich</i> | 14-15 | 16 Desember 2011 |
| 10 | 4.1.x | <i>Jelly Bean</i> | 16 | 9 Juli 2012 |
| 11 | 4.2.x | <i>Jelly Bean</i> | 17 | 13 November 2012 |
| 12 | 4.3.x | <i>Jelly Bean</i> | 18 | 24 Juli 2013 |
| 13 | 4.4.x | <i>Kitkat</i> | 19-20 | 31 Agustus 2013 |
| 14 | 5.x | <i>Lollipop</i> | 21-22 | 15 Agustus 2014 |
| 15 | 6.0 | <i>Marshmallow</i> | 23 | 19 Agustus 2015 |
| 16 | 7.0 | <i>Nougat</i> | 24-25 | 22 Agustus 2016 |

| | | | | |
|----|-----|-------------|-------|-----------------|
| 17 | 8.0 | <i>Oreo</i> | 26-27 | 21 Agustus 2017 |
| 18 | 9.0 | <i>Pie</i> | 28 | 6 Agustus 2018 |

2.2.3 *Google Play*

Google Play adalah layanan konten digital milik *Google* yang melingkupi toko daring untuk produk-produk seperti musik/lagu, buku, aplikasi, permainan, ataupun pemutar media berbasis *cloud* sebagai aplikasi induk kepada semua aplikasi yang lain. Layanan ini dapat diakses baik melalui *web*, aplikasi android (*Play Store*), dan *Google TV*. Konten yang telah dibeli tersedia pada seluruh *platform-platform* di atas *Google Play* mulai dikenalkan pada bulan Maret 2012 sebagai pengganti dari *Android Market* dan Layanan Musik *Google*. Sistem Android memiliki gudang aplikasi dan *game* yaitu *Google Playstore*, yang mana disini anda bisa mendownload serta menggunakan aplikasi atau *game* yang terdapat di *Google Play Store* sepuasnya dengan menggunakan perangkat seluler dengan sistem Android. (Wikipedia, 2019).

2.2.4 *Software Development Kit (SDK)*

Sebuah *Software Development Kit (SDK atau devkit)* tipikal merupakan satu set perkakas pengembangan *software* yang digunakan untuk mengembangkan atau membuat aplikasi untuk paket *software* tertentu, *software framework*, *hardware platform*, sistem komputer, konsol *video game*, sistem operasi atau *platform* sejenis lainnya. Ia mencakup mulai dari pemrograman sederhana seperti sebuah *application programming interface (API)*, sampai dengan pemrograman yang lebih rumit dengan *hardware* yang canggih atau pada sistem *embedded* termasuk perangkat *mobile*.

Perkakas yang lazim disertakan bersama SDK, termasuk *debugging aids* yang memudahkan penelusuran kekeliruan dan utiliti lainnya yang biasa dijumpai pada lingkungan pengembang terpadu (*integrated development environment*) IDE. Sebagian besar SDK berisi kode sampel, yang menyediakan pengembang dengan contoh program dan perpustakaan. Sampel ini membantu pengembang belajar bagaimana membangun program dasar dengan SDK, yang memungkinkan mereka

untuk akhirnya membuat aplikasi yang lebih kompleks. SDK juga menawarkan dokumentasi teknis, yang dapat mencakup tutorial dan *FAQ*.

2.2.5 Java Development Kit (JDK)

Java Development Kit (JDK) adalah satu set perangkat lunak komputer dan spesifikasi yang dikembangkan oleh *Sun Microsystems*, yang kemudian diakuisisi oleh *Oracle Corporation*, yang menyediakan sistem untuk mengembangkan aplikasi perangkat lunak dan menerapkan hal itu dalam sebuah *cross-platform* lingkungan komputasi (Wikipedia, 2019). digunakan di berbagai macam platform komputasi dari perangkat *embedded* dan ponsel ke server perusahaan dan super komputer. Sementara mereka yang kurang umum dari *standalone* aplikasi Java, Java *applet* yang berjalan di aman, *sandboxed* lingkungan untuk memberikan banyak fitur dari aplikasi asli dan dapat tertanam ke dalam HTML halaman.

Penulisan dalam bahasa pemrograman Java adalah cara utama untuk menghasilkan kode yang akan digunakan sebagai kode byte pada sebuah Mesin *Virtual Java* (JVM) byte code compiler juga tersedia untuk bahasa-bahasa lain, termasuk Ada, *JavaScript*, *Python*, dan *Ruby*. Selain itu, beberapa bahasa telah dirancang untuk berjalan secara *native* pada JVM, termasuk *Scala*, *Clojure*, dan *Apache Groovy*. sintaks meminjam banyak dari C dan C++, tetapi berorientasi objek memiliki dimodelkan setelah *Smalltalk* dan *Objective-C* *eschews* tertentu tingkat rendah konstruksi seperti *pointer* dan memiliki memori yang sangat sederhana model di mana setiap objek yang dialokasikan di heap dan semua variabel dari jenis objek referensi. Manajemen memori ditangani melalui otomatis terintegrasi pengumpulan sampah yang dilakukan oleh JVM.

2.2.6 Gradle

Gradle adalah alat bangun dengan fokus pada otomasi bangunan dan dukungan untuk pengembangan multi-bahasa. Jika membangun, menguji, menerbitkan, dan menggunakan perangkat lunak pada platform apa pun, *Gradle* menawarkan model fleksibel yang dapat mendukung seluruh siklus hidup pengembangan mulai dari menyusun dan mengemas kode hingga menerbitkan situs web.

Gradle dirancang untuk pembangunan multi-proyek, yang sedang berkembang menjadi proyek berskala besar. Mendukung bangunan bertahap dengan secara cerdas menentukan bagian mana dari pohon bangunan yang terkini yang bergantung hanya pada bagian-bagian itu tidak perlu dijalankan kembali.

Gradle telah dirancang untuk mendukung otomatisasi pembangunan lintas berbagai bahasa dan platform termasuk *Java*, *Scala*, *Android*, *C / C ++*, dan *Groovy*, dan terintegrasi erat dengan alat pengembangan dan server integrasi berkelanjutan termasuk *Eclipse*, *IntelliJ*, dan *Jenkins*.

2.2.7 Manifes Aplikasi

Setiap aplikasi harus memiliki file *AndroidManifest.xml* (bernama persis seperti ini) di direktori akar. File manifes menyediakan informasi penting tentang aplikasi ke sistem Android, yang harus dimiliki sistem agar bisa menjalankan setiap kode aplikasi. Yang dilakukan file manifes di antaranya:

1. Menamai paket Java untuk aplikasi. Nama paket berfungsi sebagai identifier unik untuk aplikasi.
2. Menjelaskan berbagai komponen aplikasi, yang menyertakan aktivitas, layanan, penerima siaran, dan penyedia materi yang membentuk aplikasi. Juga menamai kelas yang mengimplementasikan masing-masing komponen dan menerbitkan kemampuannya, seperti pesan Intent yang dapat mereka tangani. Deklarasi ini menginformasikan sistem Android mengenai komponen dan kondisi yang memungkinkan peluncurannya.
3. Menentukan proses yang menjadi host komponen aplikasi.
4. Mendeklarasikan izin aplikasi yang harus dimiliki aplikasi untuk mengakses bagian yang dilindungi pada API dan berinteraksi dengan aplikasi lain. Juga mendeklarasikan izin lain yang harus dimiliki untuk berinteraksi dengan komponen aplikasi.
5. Mencantumkan daftar kelas Instrumentation yang memberikan profil dan informasi lain saat aplikasi berjalan. Deklarasi ini hanya ada di manifes saat aplikasi dibuat dan dihapus sebelum aplikasi dipublikasikan.
6. Mendeklarasikan level minimum Android API yang diperlukan aplikasi.
7. Mencantumkan daftar pustaka yang harus ditautkan aplikasi.

2.2.8 *Activity*

Activity adalah sebuah komponen aplikasi yang menyediakan layar yang digunakan pengguna untuk berinteraksi guna melakukan sesuatu, misalnya memilih nomor ponsel, mengambil foto, mengirim email, atau menampilkan peta. Tiap aktivitas diberi sebuah jendela untuk menggambarkan penggunanya. Jendela ini biasanya mengisi layar, namun mungkin lebih kecil daripada layar dan mengambang di atas jendela lain.

Sebuah aplikasi biasanya terdiri atas beberapa aktivitas yang terikat secara longgar satu sama lain. Biasanya, satu aktivitas dalam aplikasi ditetapkan sebagai aktivitas "utama", yang ditampilkan kepada pengguna saat membuka aplikasi untuk pertama kali. Tiap aktivitas kemudian bisa memulai aktivitas lain untuk melakukan berbagai tindakan. Tiap kali aktivitas baru dimulai, aktivitas sebelumnya akan dihentikan, namun sistem mempertahankan aktivitas dalam sebuah tumpukan ("back-stack"). Bila sebuah aktivitas baru dimulai, aktivitas itu akan didorong ke atas back-stack dan mengambil fokus pengguna. Back-stack mematuhi mekanisme dasar tumpukan "masuk terakhir, keluar pertama", jadi, bila pengguna selesai dengan aktivitas saat ini dan menekan tombol Kembali, aktivitas akan dikeluarkan dari tumpukan (dan dimusnahkan) dan aktivitas sebelumnya akan dilanjutkan.

Bila aktivitas dihentikan karena ada aktivitas baru yang dimulai, aktivitas lama akan diberi tahu tentang perubahan status ini melalui metode callback daur hidupnya. Ada beberapa metode *callback* yang mungkin diterima aktivitas, karena sebuah perubahan dalam statusnya apakah sistem sedang membuatnya, menghentikannya, melanjutkannya, atau menghapuskannya dan masing-masing callback memberi Anda kesempatan melakukan pekerjaan tertentu yang sesuai untuk perubahan status itu. Misalnya, bila dihentikan, aktivitas Anda harus melepas objek besar, seperti koneksi jaringan atau database. Bila aktivitas dilanjutkan, Anda bisa memperoleh kembali sumber daya yang diperlukan dan melanjutkan tindakan yang terputus. Transisi status ini semuanya bagian dari daur hidup aktivitas.

Bagian selebihnya dari dokumen ini membahas dasar-dasar cara membangun dan menggunakan aktivitas, yang meliputi satu pembahasan lengkap

tentang cara kerja daur hidup aktivitas, sehingga Anda bisa dengan benar mengelola transisi di antara berbagai status aktivitas.

2.2.9 *Kotlin*

Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada Mesin *Virtual Java* ataupun menggunakan kompiler *LLVM* yang dapat pula dikompilasikan kedalam bentuk kode sumber JavaScript. Pengembang utamanya berasal dari tim *programer* dari *JetBrains* yang bermakas di Rusia. Meskipun sintaksisnya tidak kompatibel dengan bahasa *Java*, *Kotlin* didesain untuk dapat bekerja sama dengan kode bahasa *Java* dan bergantung kepada kode bahasa *Java* dari Kelas Pustaka *Java* yang ada, seperti berbagai *framework* *Java* yang ada. Tim Pengembang memutuskan menamakannya *Kotlin* dengan mengambil nama dari sebuah pulau di Rusia, sebagaimana *Java* yang mengambil nama dari pulau Jawa di Indonesia. Setelah Google mengumumkan bahwa *Kotlin* menjadi bahasa kelas satu bagi *Android*, maka bersama *Java* dan *C++*, *Kotlin* menjadi bahasa resmi untuk pengembangan aplikasi-aplikasi. Fitur Unggulan *Kotlin*:

1. Aman dari *Null*, tidak perlu lagi menghawatirkan kemungkinan timbulnya *Null Pointer Exception*. *Compiler* secara sistematis akan menandai *pointer* yang berpotensi *null*.
2. Pemrograman Fungsional, mendukung lambda dan kemampuan *mapping*, *folding*, pada *Collection Java*.
3. Anotasi Data yang otomatis dibuat untuk mengurangi *boilerplate* seperti *equals*, *hashCode*, *toString*.
4. *Syntax Ramping*, yang menawarkan penulisan *syntax* secara cepat. Membuat fungsi *one-liner*, membuat *Java Beans* hanya dengan satu baris, method *setter* dan *getter* otomatis.
5. Ekstensi Fungsi yang memungkinkan untuk menambahkan method pada class tanpa harus mengubah *source code*.
6. Interpolasi String yang memungkinkan untuk memasukkan variabel ke dalam String tanpa penyambungan seperti pada *EcmaScript*.
7. Penamaan argumen, *parameter* opsional, *default value* untuk *parameter*.

2.2.10 Firebase

Firebase adalah suatu layanan dari google untuk mempermudah para pengembang-pengembang aplikasi untuk mengembangkan aplikasinya. *Firebase*(‘*Backend as a Service*’) ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Developer* tanpa harus memberikan *effort* yang besar untuk urusan *backend*. *Firebase* memiliki beberapa yang disediakan., yaitu :

1. Google Analytic

Analytics menyajikan data seputar perilaku pengguna pada aplikasi Android dan iOS agar Anda dapat mengambil keputusan yang lebih baik tentang produk dan pengoptimalan pemasaran. Lihat data error, efektivitas *notification*, performa *deep link*, data pembelian dalam aplikasi, dan lain-lain.

2. Real-time database

Untuk Menyimpan dan sinkronkan data antara pengguna dan perangkat secara *realtime* menggunakan *database noSQL* yang dihosting secara *cloud*. *real-time* adalah kondisi pengoperasian dari suatu sistem perangkat keras dan perangkat lunak yang dibatasi oleh rentang waktu dan memiliki tenggat waktu (*deadline*) yang jelas, relatif terhadap waktu suatu peristiwa atau operasi terjadi, jadi data akan dikirim seketika waktu itu juga.

3. Authentication

Untuk mengelelola pengguna dengan cara yang mudah dan aman. *Firebase Auth* menawarkan beberapa metode autentikasi, termasuk email/sandi, penyedia pihak ketiga seperti *Google* atau *Facebook*, atau langsung menggunakan sistem akun Anda yang sudah ada.

4. Cloud Storage

Untuk menyimpan dan bagikan gambar, *audio*, *video*, atau konten lain yang dibuat pengguna secara mudah dengan penyimpanan *object* yang andal, sederhana, dan hemat biaya yang dikembangkan untuk skala *Google*.

5. Hosting

Permudah hosting web statis Anda dengan fitur yang dibuat khusus untuk aplikasi web modern. Saat Anda mengupload aset web, kami secara

otomatis akan memasukkannya ke *CDN* global kami, dan memberinya sertifikat *SSL* gratis, sehingga pengguna Anda akan mendapatkan pengalaman yang aman, andal, berlatensi rendah, di mana pun mereka berada.

Firebase memiliki banyak library yang memungkinkan untuk mengintegrasikan layanan ini dengan *Android*, *iOS*, *Javascript*, *Java*, *Objective-C* dan *Node.JS*. *Database Firebase* juga bersifat bisa diakses lewat *REST API* dan data binding untuk beberapa *framework Javascript* seperti halnya *AngularJS*, *ReactJS*, *Ember.JS*, dan *Backbone.JS*. *REST API* tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima *push notification* dari *server*. Pengembang juga bisa menggunakan database ini untuk mengamankan data mereka menggunakan *server Firebase* dengan *rules* yang ada.

2.2.11 NoSQL

NoSQL adalah jenis basis data yang tidak menggunakan perintah *SQL* dalam memanipulasi (menyimpan maupun mengambil data) basis data tersebut. Kebanyakan basis data *NoSQL* digunakan dalam dunia aplikasi web waktunya (*real-time web app*). Istilah yang dikenal dalam teknologi komputasi untuk merujuk kepada kelas yang luas dari sistem manajemen basis data yang diidentifikasi dengan tidak mematuhi aturan pada model sistem manajemen basis data relasional yang banyak digunakan (Wikipedia, 2019).

2.2.12 Xml

XML (Extensible Markup Language) adalah bahasa markup untuk keperluan umum yang disarankan oleh *W3C* untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. *XML* merupakan kelanjutan dari *HTML (HyperText Markup Language)* yang merupakan bahasa standar untuk melacak Internet (Wikipedia, 2019). *XML* didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama *XML* adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. *XML* menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, *XML* dapat digunakan untuk menggambarkan

sembarang view *database*, tetapi dengan satu cara yang standar. *XML* memiliki tiga tipe file:

1. *XML*, merupakan standar format dari struktur berkas (file) yang ada.
2. *XSL*, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. *XSD*, merupakan standar yang mendefinisikan struktur database dalam *XML*.

Keunggulan *XML* bisa diringkas sebagai berikut:

1. Pintar (Intelligence). *XML* dapat menangani berbagai tingkat (level) kompleksitas.
2. Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti Microsoft membuat bahasa *MSXML* atau Macromedia mengembangkan *MXML*.
3. Mudah pemeliharaannya.
4. Sederhana. *XML* lebih sederhana.
5. Mudah dipindah-pindahkan (*Portability*). *XML* mempunyai kemudahan perpindahan (portabilitas) yang lebih bagus.

2.2.13 Json

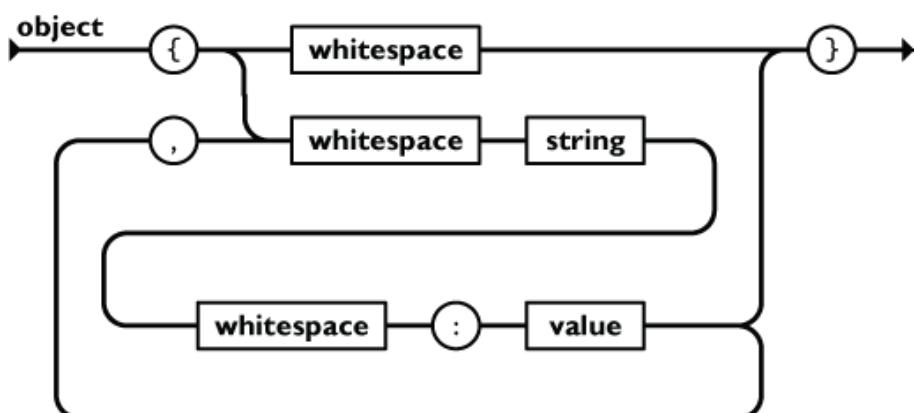
JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.

2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

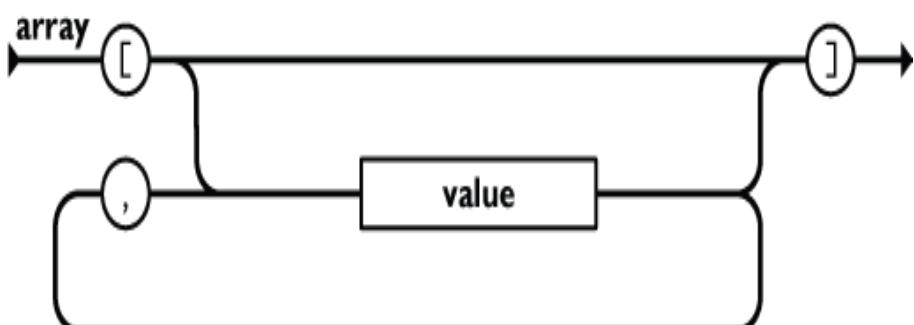
Struktur-struktur data ini disebut sebagai struktur data *universal*. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini. *JSON* menggunakan bentuk sebagai berikut:

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan {kurung kurawal buka dan diakhiri dengan }kurung kurawal tutup. Setiap nama diikuti dengan :titik dua dan setiap pasangan nama/nilai dipisahkan oleh koma.



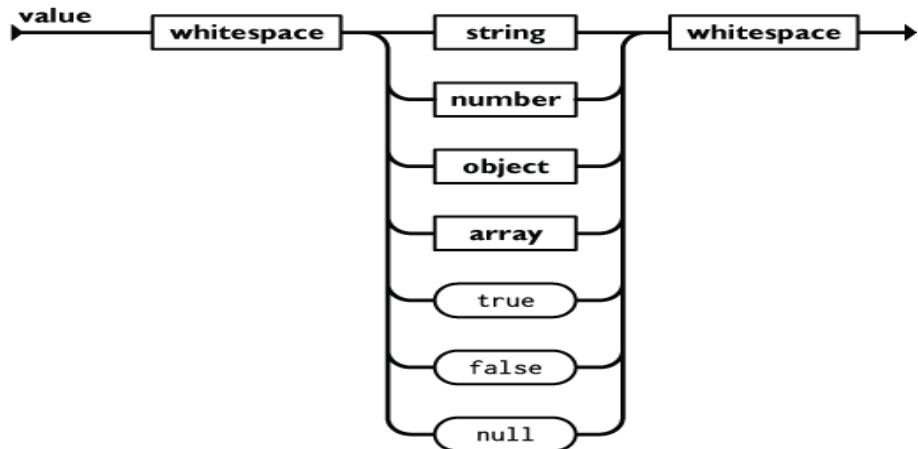
Gambar 2.1 Json Object

Larik (*Array*) adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [kurung kotak buka dan diakhiri dengan]kurung kotak tutup. Setiap nilai dipisahkan oleh koma.



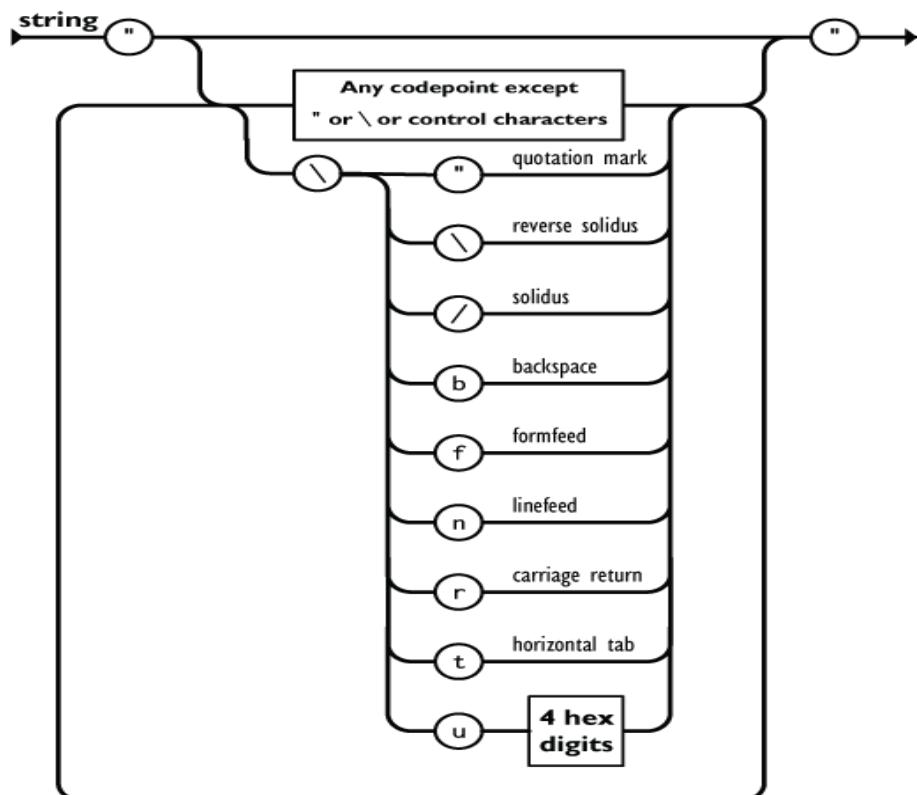
Gambar 2.2 Json Array

Nilai (*value*) dapat berupa sebuah *string* dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



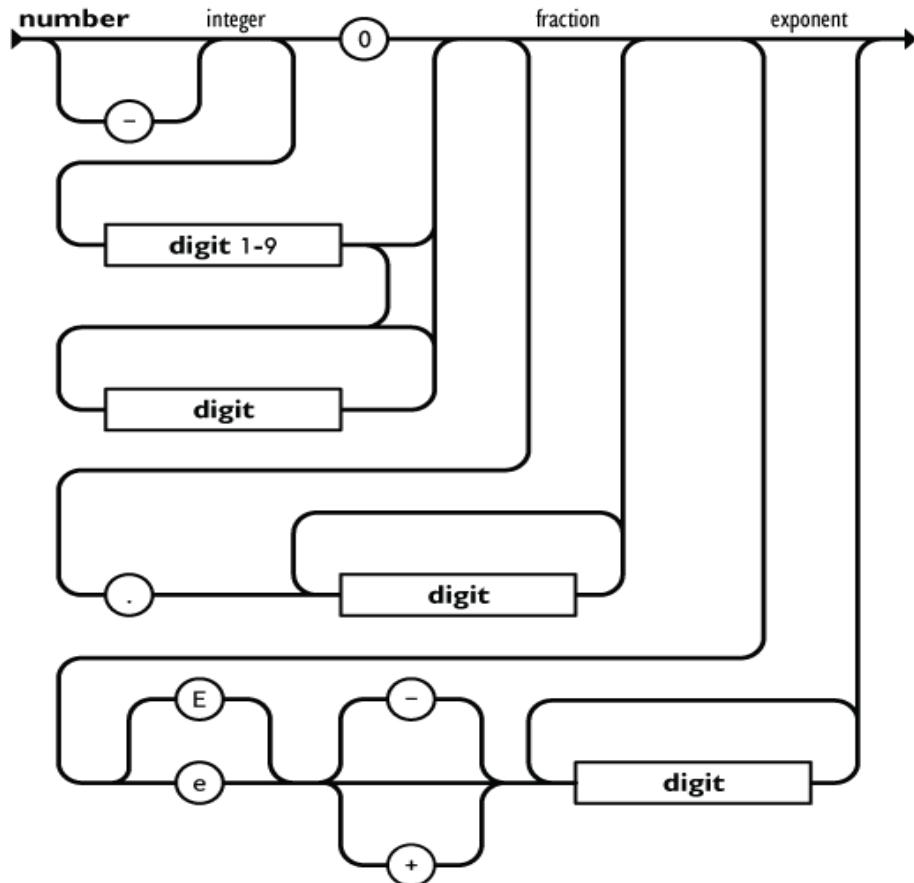
Gambar 2.3 Json Value

String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam *string* dapat digunakan *backslash escapes* "\\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. String sangat mirip dengan string C atau Java.



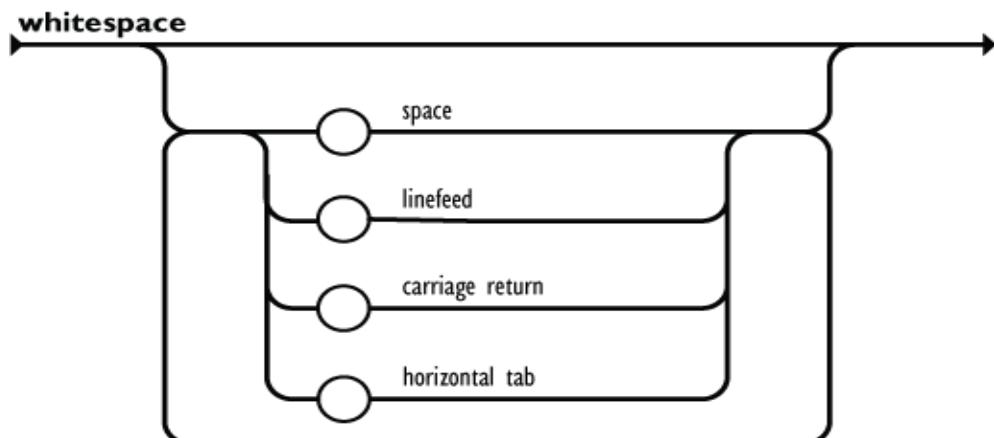
Gambar 2.4 Json String

Angka adalah sangat mirip dengan angka di C atau *Java*, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2.5 *Json Number*

Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detil *encoding* yang secara lengkap dipaparkan oleh bahasa pemrograman yang bersangkutan.



Gambar 2.6 *Json Whitespace*

2.2.14 *Android Studio*

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment (IDE)* untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA* (Android Studio, 2019). Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

1. Sistem versi berbasis *Gradle* yang fleksibel
2. *Emulator* yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*

Setiap proyek di *Android Studio* berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

1. Modul aplikasi Android
2. Modul Pustaka
3. Modul *Google App Engine*

Secara default, *Android Studio* akan menampilkan file proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam Gambar 1. Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek Anda. Semua file versi terlihat di bagian atas di bawah *Gradle Scripts* dan masing-masing modul aplikasi berisi folder berikut:

1. *manifests*: Berisi file *AndroidManifest.xml*.

2. *java*: Berisi file kode sumber Java, termasuk kode pengujian *JUnit*.
3. *res*: Berisi semua sumber daya bukan kode, seperti tata letak *XML*, *string* *UI*, dan gambar *bitmap*.

2.2.15 Genymotion

Genymotion adalah emulator Android lengkap untuk Windows. Emulator ini sangat unggul dan mudah digunakan walaupun oleh pengguna kasual sekali pun, yang semestinya tidak mengalami masalah dalam menelusuri dan mengendalikan program. Anda dapat menggunakan alat ini untuk mengemulasi seluruh peranti Android (uptodown, 2019).

2.2.16 Postman

Postman merupakan sebuah software yang memuat fungsi lengkap pengembangan sistem dalam mengirimkan dan menerima respon server. Software ini mendukung pengembangan sistem *REST API* dengan mengklasifikasi request berdasarkan request method, *URL* dan *parameter-parameter request* (Posdot, 2017).

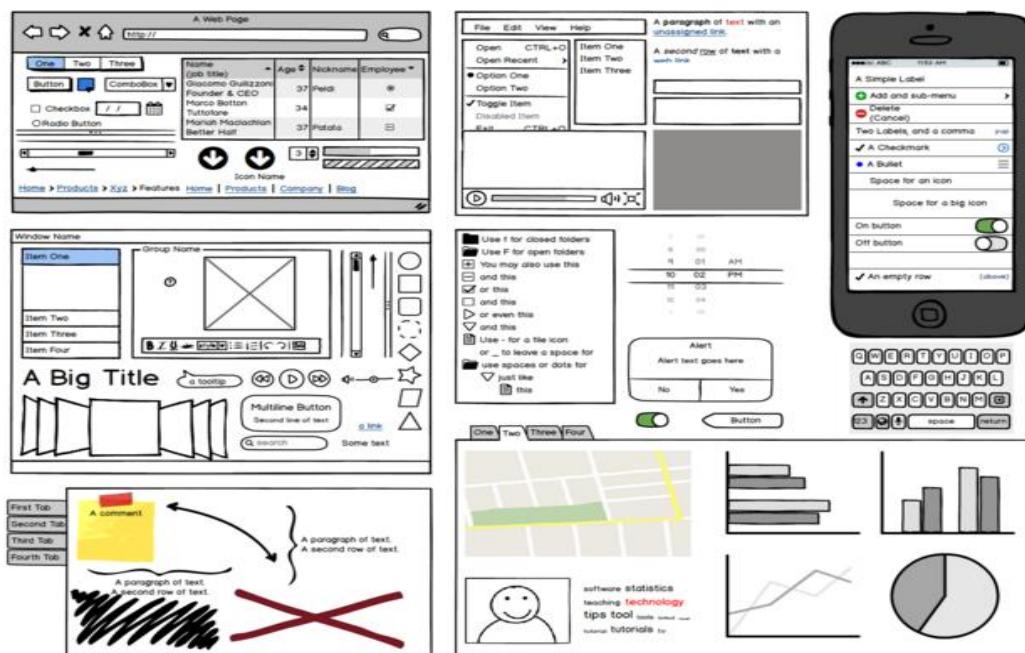
2.2.17 Balsamiq Mockup

Mockup artinya model atau replika mesin atau struktur, yang digunakan untuk tujuan instruksional atau eksperimental. *Balsamiq mockup* adalah program aplikasi yang digunakan dalam pembuatan tampilan user interface sebuah aplikasi. Software ini sudah menyediakan *tools* yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan kita buat. Software ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna (Balsamiq, 2019).

Alih-alih menggambar sketsa (*wireframe*) atau *prototype* rancangan desain website di atas kertas balsamiq mockups membantu seorang *web designer* membuat tampilan *web* dalam bentuk gambar di komputer. Tujuannya selain agar membuat tampilan (desain) *website* menarik juga dapat menyesuaikan dengan

kebutuhan *customer* (pelanggan). Dengan alat pembuat *mockup* maka seorang *web designer* dapat menganalisa tata letak, desain dan fungsi.

Konten yang terbuat seperti dari gambaran tangan, itu akan membuat Anda fokus pada pemecahan masalah *UI* yang lebih besar, daripada pada perincian *website*. Di websitenya sendiri ada dua pilihan untuk para pengguna, ada versi trial for desktop dan ada juga yang bisa kita download untuk versi desktop. Namun ada juga yang disediakan dalam versi berbayar. Aplikasi ini bisa digunakan untuk sistem operasi *Windows*, *Mac OS*, dan *Linux*.



Gambar 2.7 *Balsamic Mockup*

2.2.18 Start UML

StarUML adalah sebuah proyek *open source* untuk pengembangan secara cepat, fleksibel, *extensible*, *featureful*, dan bebas-tersedia. UML / platform MDA berjalan pada platform *Win32*. Tujuan dari proyek *StarUML* adalah untuk membangun sebuah alat pemodelan perangkat lunak dan juga platform yang menarik adalah pengganti alat UML komersial seperti Rational Rose, Together dan sebagainya. *Star UML* mendukung UML (*Unified Modeling Language*). Berdasarkan pada *UML* version 1.4 dan dilengkapi 11 macam diagram yang berbeda, selanjutnya mendukung notasi *UML* 2.0 dan juga mendukung

pendekatan *MDA (Model Driven Architecture)* dengan dukungan konsep UML. StarUML dapat memaksimalkan produktivitas dan kualitas dari suatu software project.

UML 2.0 itu sendiri adalah *UML* standar yang terus berkembang dan dikelola oleh *OMG (Object Management Group)*. Baru-baru ini, UML 2.0 direlease dan *StarUML* dukungan UML 2.0 yang akan mendukung standar terbaru *UML*. Selanjutnya *MDA (Model Driven Architecture)*. *MDA* adalah teknologi baru yang diperkenalkan oleh *OMG*. Untuk mendapatkan keuntungan dari *MDA*, perangkat lunak alat pemodelan harus mendukung variabel banyak kustomisasi. *StarUML* dirancang untuk mendukung *MDA* dan menyediakan kustomisasi banyak variabel seperti profil *UML* sebagai, Pendekatan, Model Kerangka, NX (ekstensi notasi), kode *MDA* dan dokumen template dan sebagainya. Ini akan membantu Anda dalam alat pemasangan ke dalam budaya organisasi, proses, dan proyek Anda.

StarUML menyediakan secara sederhana dan kuat arsitektur plug-in sehingga siapapun dapat mengembangkan plug-in modul dalam bahasa *COM-kompatibel (C + +, Delphi, C #, VB, dll)*. Hal yang paling penting dalam pengembangan perangkat lunak adalah Usability. *StarUML* diimplementasikan untuk memberikan berbagai fitur yang user-friendly seperti dialog Cepat, manipulasi Keyboard, ikhtisar Diagram, dll. *StarUML* sebagian besar ditulis dalam Delphi. Namun, *StarUML* adalah proyek multi-bahasa dan tidak terikat dengan bahasa pemrograman tertentu, sehingga setiap bahasa pemrograman dapat digunakan untuk mengembangkan *StarUML*. (Misalnya, *C / C + +, Java, Visual Basic, Delphi, JScript, VBScript, C #, VB.NET, dll*).

2.2.19 Tipe Diagram UML

Tipe Diagram UML (*Unified Modeling Language*) adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisaikan artifak dari proses analisis dan desain berorientasi objek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu sistem. UML dikembangkan oleh 3 orang “berorientasi objek”, yaitu *Grady Booch*, *Jim Rumbaugh*, dan *Ivar Jacobson*. *UML* menjadi bahasa yang bisa digunakan untuk

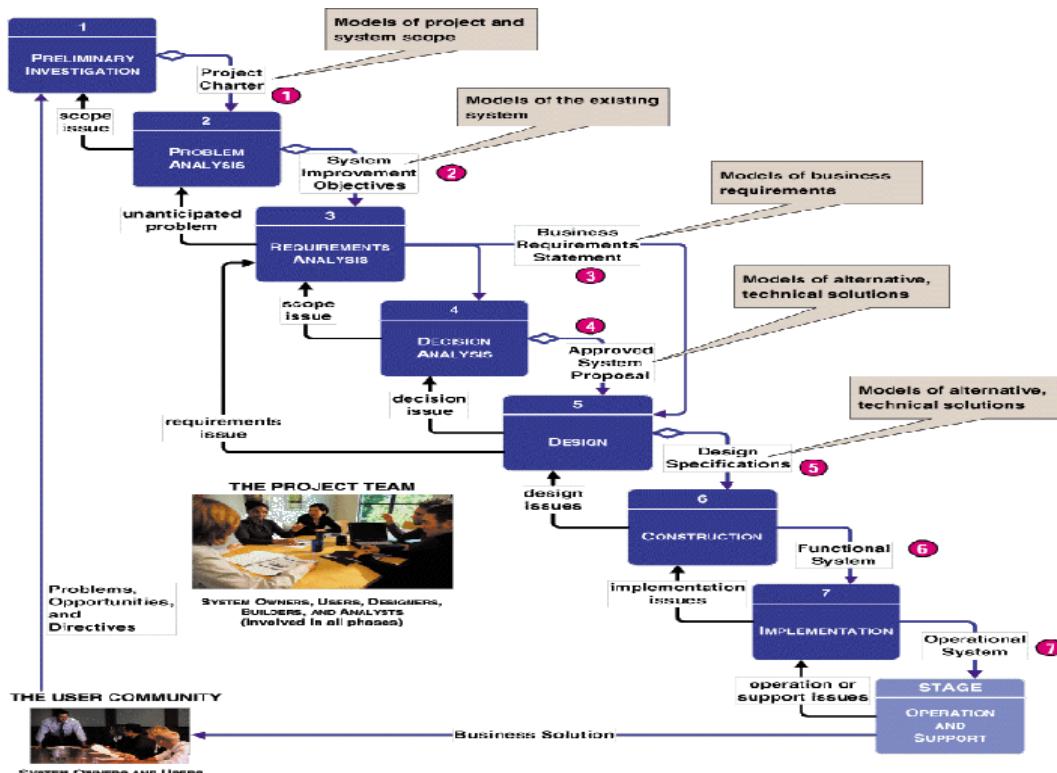
berkomunikasi dalam perspektif objek antara *user* dengan *developer*, antara *developer* dengan *developer*, antara *developer* analis dengan developer disain, dan antara developer desain dengan developer pemrograman. *UML* memungkinkan *developer* melakukan permodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Permodelan visual membantu untuk menangkap struktur dan kelaukan dari objek, mempermudah penggambaran interaksi antara elemen dalam system, dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman (Widianto, 2012).

Tabel 2.4 Tipe *Diagram UML*

| Nama | Keterangan |
|------------------------------------|--|
| <i>Use Case Diagram</i> | Menunjukkan pelaku (orang atau pengguna lain dari sistem), menggunakan kasus (skenario ketika mereka menggunakan sistem), dan hubungan pelaku. |
| <i>Class Diagram</i> | Menunjukkan kelas dan hubungan antara kelas. |
| <i>Sequence Diagram</i> | Menunjukkan objek dan urutan pemanggilan metode yang mereka buat untuk objek lain. |
| <i>Collaboration Diagram</i> | Menunjukkan objek dan hubungannya, menempatkan penekanan pada objek yang berpartisipasi dalam pertukaran pesan. |
| <i>State Diagram</i> | Menyatakan state, perubahan state dan peristiwa dalam suatu objek atau bagian dari sistem. |
| <i>Activity Diagram</i> | Menunjukkan kegiatan dan perubahan dari satu aktivitas ke aktivitas lainnya dengan peristiwa yang terjadi di beberapa bagian dari sistem. |
| <i>Component Diagram</i> | Menunjukkan komponen pemrograman tingkat tinggi. |
| <i>Deployment Diagram</i> | Menunjukkan contoh dari komponen dan hubungan-nya. |
| <i>Entity Relationship Diagram</i> | Menunjukkan data , hubungan dan kendala antara data. |

2.2.20 Model Driven Development (MDD)

Model Driven Development (MDD) memanfaatkan model grafis dan komponen aplikasi yang sudah dibangun sehingga pengguna dapat membangun aplikasi yang kompleks secara visual. Ketertarikan pada pengembangan yang didorong oleh model terus meningkat ketika individu mengalami kemudahan dan kecepatan yang dengannya pengguna teknis dan bisnis mengatur dan menggunakan aplikasi bisnis. Platform pengembangan yang digerakkan oleh model yang tepat memberikan keunggulan produktivitas yang signifikan dibandingkan metode pengembangan tradisional dan memungkinkan pengiriman proyek oleh tim yang lebih kecil.



Gambar 2.8 Model Driven Development (MDD) Route

Model Driven Development (MDD) telah muncul sebagai salah satu pendekatan terkemuka untuk memungkinkan pengembangan aplikasi kolaboratif yang cepat. Karena pengembangan berbasis model menggunakan teknik pemodelan visual untuk menentukan hubungan data, logika proses, dan membangun antarmuka pengguna, pengembangan perangkat lunak berbasis model memberdayakan baik pengembang maupun pengguna bisnis untuk secara cepat

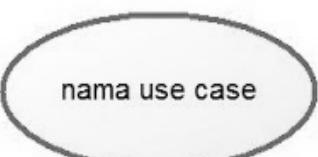
memberikan aplikasi tanpa perlu kode. Akibatnya, pengembangan *model-driven* secara signifikan lebih cepat daripada bahasa pemrograman tradisional seperti *C #* dan *Java*.

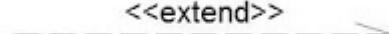
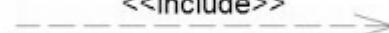
2.2.21 Use Case Diagram

Diagram *Use Case* atau *Use Case Diagram* adalah pemodelan untuk menggambarkan behavior / kelakuan sistem yang akan dibuat. Use case diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, diagram use case digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut. Menurut (Rosa & Shalahuddin, 2011) *use case* diagram tidak menjelaskan secara detail tentang penggunaan tiap *use case*, namun hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Melalui *use case* diagram kita dapat mengetahui fungsi-fungsi apa saja yang ada pada sistem. Adapun syarat penamaan pada *use case* diagram sendiri adalah nama didefinisikan sesederhana mungkin sehingga bisa dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
2. Aktor adalah orang atau system lain yang berinteraksi dengan system yang akan dibuat, jadi meskipun simbol dari aktor adalah gambar orang tapi aktor belum tentu merupakan orang.

Tabel 2.5 Simbol *Use Case Diagram*

| Simbol | Deskripsi |
|--|---|
| <i>Use Case</i>  | <i>Use case</i> adalah fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar <i>unit</i> atau <i>actor</i> . biasanya <i>use case</i> diberikan penamaan dengan menggunakan kata kerja di awal frase nama use case |

| | |
|--|--|
| Aktor / actor  nama aktor | Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi meskipun simbol dari aktor ialah gambar orang, tapi aktor belum tentu merupakan orang. biasanya penamaan aktor dinamakan menggunakan kata benda di awal frase nama aktor |
| Asosiasi / association  | Asosiasi adalah komunikasi antara aktor dan use case yang berpartisipasi pada use case diagram atau use case yang memiliki interaksi dengan aktor. Asosiasi merupakan simbol yang digunakan untuk menghubungkan link antar element. |
| Ekstend / extend  | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri meski tanpa use case tambahan itu arah panah mengarah pada use case yang ditambahkan |
| Include  | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan membutuhkan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini arah panah include mengarah pada use case yang dipakai (dibutuhkan) atau mengarah pada use case tambahan. |
| Generalisasi / generalization  | Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah use case dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya. arah panah mengarah pada use case yang menjadi generalisasinya (umum) |

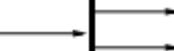
2.2.22 Activity Diagram

Activity Diagram atau Diagram aktivitas adalah bentuk visual dari alur kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, atau pengulangan. Dalam *Unified Modeling Language*(UML), diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun alur aktivitas dalam organisasi. Selain itu diagram aktivitas juga menggambarkan alur kontrol secara garis besar.

Diagram aktivitas memiliki komponen dengan bentuk tertentu, dihubungkan dengan tanda panah. Panah tersebut mengarahkan urutan aktivitas yang terjadi, dari awal sampai akhir. Yang perlu diperhatikan yaitu diagram aktivitas bukan menggambarkan aktivitas sistem yang dilakukan aktor, tetapi menggambarkan aktivitas yang dapat dilakukan oleh sistem.

Activity diagram pada beberapa waktu dianggap sama seperti flowchart (diagram alur), namun meskipun diagram terlihat seperti sebuah diagram alur, tetapi sebenarnya berbeda. *Activity diagram* menunjukkan aliran yang berbeda seperti paralel, bercabang, bersamaan dan tunggal.

Tabel 2.6 Simbol *Activity Diagram*

| Simbol | Deskripsi |
|---|---|
| Start Point  | Menggambarkan awal dari aktivitas. |
| End Point  | Menggambarkan akhir dari aktivitas. |
| Activity  | Menggambarkan suatu proses atau kegiatan bisnis. |
| Fork  | Menggambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan. |

| | |
|-------------------------|---|
| Join | untuk menggabungkan kembali activity atau action yang paralel. |
| Decision | Menggambarkan keputusan atau pilihan. |
| State Transition | Menggambarkan aliran perpindahan kontrol antara state. |
| Swimlane | Menggambarkan pemisahan atau pengelompokan aktivitas berdasarkan aktor. |

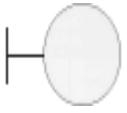
2.2.23 Sequence Diagram

Sequence Diagram adalah salah satu dari diagram – diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Sequence Diagram merupakan *Interaction Diagram* yang digunakan untuk menjelaskan eksekusi sebuah skenario semantik. *Sequence Diagram* juga digunakan untuk menjelaskan interaksi antar objek dalam urutan waktu.

Tabel 2.7 Simbol *Sequence Diagram*

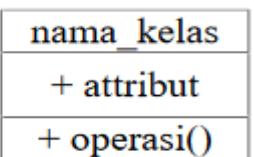
| Simbol | Keterangan |
|---------------|--|
| Actor | Menggambarkan orang yang sedang berinteraksi dengan sistem. |
| Entity | Menggambarkan informasi yang harus disimpan oleh sistem(struktur data dari sebuah sistem). |

| | |
|---|--|
| Boundary | Menghubungkan antara user dengan sistem. |
|  | |
| Control | Menggambarkan “perilaku mengatur”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem. |
|  | |
| Object Message <i>Message()</i> | Menggambarkan pengiriman pesan. |
|  | |
| Message to Self  | Sebuah objek yang mempunyai sebuah pesan kepada dirinya sendiri. |
| Loop  | Menggambarkan perulangan dalam <i>sequence</i> . |

2.2.24 Class Diagram

Pengertian *Class Diagram* adalah visual dari struktur sistem program pada kelompok-kelompok yang dibentuk. *Class diagram* merupakan alur jalannya database pada sistem. *Class diagram* adalah gambaran alur database dalam program. Dalam sebuah laporan sistem maka class diagram wajib ada. *Sequence diagram* adalah bagian dari *UML (Unified Modeling Language)*. Dalam sebuah laporan sequence diagram wajib ada karena sangat penting.

Tabel 2.8 Simbol *Class Diagram*

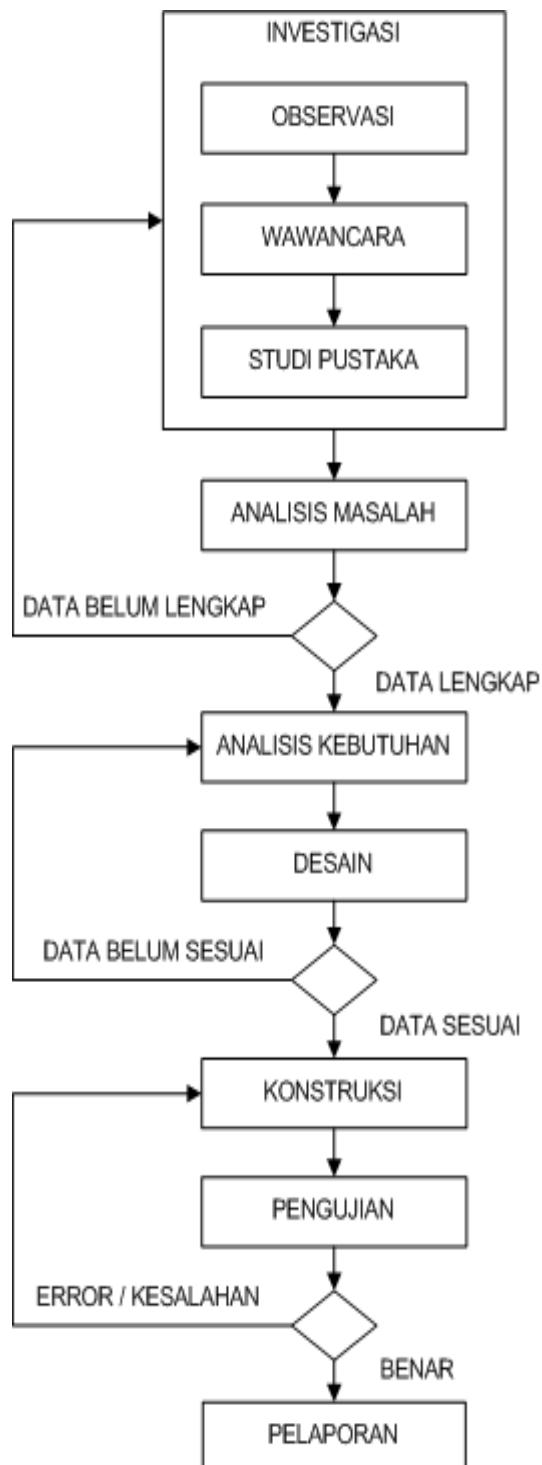
| Simbol | Deskripsi |
|---|----------------------------|
| Kelas  | Kelas pada struktur sistem |

| | |
|---|--|
| Antarmuka / interface  nama interface | Sama dengan konsep interface dalam pemrograman berorientasi objek |
| Asosiasi / association  | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity |
| Asosiasi berarah / directed association  | Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity |
| Generalisasi  | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus) |
| Kebergantungan / dependency  | Relasi antarkelas dengan makna kebergantungan antarkelas |
| Agregasi / aggregation  | Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>) |

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Pikir



Gambar 3.1 Gambar Kerangka Pikir

3.2 Deskripsi

3.2.1 Investigasi

Investigasi dilakukan di FTI UNIBBA beralamat pada Jl. R.A.A Wiranatakusumah No.7 Baleendah. FTI UNIBBA ini terbagi dalam dua program studi, IF (Teknik Informatika) dan SI (Sistem Informasi). Dalam melakukan investigasi dibagi dalam tiga metode berikut uraiannya:

1. Observasi

Observasi dilakukan dengan mengamati sistem yang sedang berjalan saat ini bertujuan untuk mengetahui secara langsung bagaimana proses pengumpulan data alumni.

2. Wawancara (*Interview*)

Wawancara diterapkan dalam melakukan penelitian ini, dimulai dari meminta izin kepada dekan FTI UNIBBA bapak Yudi Herdiana S.T.,MT setelah diizinkan lanjut menemui bagian kepala staf tata usaha bapak Yusuf Muhamram, S.Kom, penulis melakukan beberapa pertanyaan mengenai tracer study alumni adalah sebagai berikut:

- a. Apakah kegiatan *tracer study* alumni masih manual?

Tidak, karena untuk saat ini sistem yang sudah berjalan memanfaatkan layanan *google form* tapi masih memiliki keterbatasan fungsional dan mengolah data kembali untuk disajikan dalam bentuk laporan.

- b. Apakah semua lulusan merespon kegiatan ini dan memperbarui data terbaru?

Sebagian besar alumni merespon dengan baik untuk kekurangannya terletak pada perbarui data terbaru alumni, karena fitur *google form* hanya melakukan input data saja, tidak bisa memperbarui data maupun login ke sistem untuk melihat hasil respon alumni dan hasilnya ketika admin mengecek hanya berupa daftar isi *tracer* alumni dalam format excel.

- c. Hambatan apa saja yang menjadi kendala saat melakukan kegiatan tracer alumni?

Hambatan yang terjadi kurangnya alumni meluangkan waktu serta pengingat dalam melakukan pengisian *tracer* alumni.

3. Studi Pustaka

Studi pustaka atau pengumpulan data yang dilakukan dengan mempelajari pustaka baik dari dokumentasi dan gambar-gambar yang berhubungan dengan tracer study alumni di FTI UNIBBA maupun informasi dari internet yang digunakan untuk bahan-bahan penelitian yang dibutuhkan.

3.2.2 Analisis Masalah

Berdasarkan hasil dari investigasi analisis masalah yang ada dalam kegiatan *tracer study* alumni FTI UNIBBA adalah sebagai berikut:

1. Sistem yang berjalan saat ini dalam mengumpulkan data *tracer study* alumni masih menggunakan formulir isian dari layanan *google form/docs*.
2. Bagian Kemahasiswaan memproses formulir isian *tracer study* dengan cara merekap data alumni satu persatu menggunakan aplikasi *Microsoft office excel*.
3. Sistem yang berjalan saat ini masih memiliki keterbatasan fungsional dalam pembaharuan data alumni dan pelaporan hasil *tracer study* alumni

Melihat dari adanya permasalahan yang sering terjadi pada kegiatan *tracer study*, penulis memberikan batasan masalah yang akan ada pada sistem baru yang akan dibangun, terdiri dari:

1. Aplikasi ini dikhusruskan untuk pengguna lulusan Fakultas Teknologi Informasi Universitas Bale Bandung
2. Aplikasi memiliki dua akses yaitu hak akses admin dan user
3. Aplikasi yang dikembangkan adalah aplikasi mobile berbasis android
4. Database firebase menggunakan versi tidak berbayar atau gratis

Jika terjadi kesalahan atau kekurangan pada tahapan ini maka akan kembali ke tahapan investigasi sampai mendapatkan hasil yang tepat dan sesuai kemudian lanjut ke tahap berikutnya.

3.2.3 Analisis Kebutuhan

Analisis kebutuhan tracer study alumni di FTI UNIBBA sebagai berikut:

1. Membangun sistem informasi data *tracer study* alumni dalam pengumpulan data dokumen yang berupa berkas maupun file excel yang harus diolah

kembali menjadi dokumen digital yang siap digunakan dalam melihat perkembangan hasil *tracer study* alumni

2. Membangun sistem informasi *tracer study* alumni untuk mempermudah akses pengelolaan kegiatan, pelaporan data alumni dan terintegrasi kebagian akademik.

Setelah Melakukan analisis kebutuhan secara umum, penulis menyampaikan kebutuhan untuk aplikasi *Tracer Alumni Berbasis Android Studi Kasus Di Fakultas Teknologi Informasi*, sebagai berikut:

3. Kebutuhan fungsional:
 - a. Aplikasi dapat diakses online
 - b. Aplikasi dapat melakukan login dengan akses admin maupun user
 - c. Aplikasi dapat melakukan logout dengan akses admin maupun user
 - d. Aplikasi dapat melakukan pengelolaan laporan
 - 1) Dapat menampilkan data rekapitulasi tracer study alumni kedalam grafik maupun tabel.
 - 2) Dapat menampilkan data serapan dengan kategori pekerjaan pegawai negeri sipil (PNS), wiraswasta dan wirausaha.
 - 3) Dapat menampilkan data tracer study alumni yang diakses oleh bagian akademik
 - e. Aplikasi dapat melakukan komunikasi antara sesama pengguna
 - f. Aplikasi dapat melihat forum informasi lowongan pekerjaan bagi alumni yang belum bekerja
 - g. Aplikasi dapat melihat daftar nama alumni
 - h. Aplikasi dapat melihat lokasi keberadaan alumni
 - i. Aplikasi dapat melakukan pengelolaan data user alumni
 - 1) Dapat melakukan pencarian data alumni
 - 2) Dapat melakukan penambahan data alumni
 - 3) Dapat melakukan pembaharuan data alumni
 - 4) Dapat melakukan penghapusan data alumni yang tidak sesuai
 - 5) Dapat melakukan input pengumuman
 - 6) Dapat melakukan pencarian, penambahan dan penghapusan dalam menu forum, informasi lowongan kerja dan pengumuman

4. Kebutuhan non fungsional:
 - a. *Smartphone android dengan spesifikasi minimum sistem operasi 4.1 jelly bean*
 - b. *Windows 10 dan linux mint versi 18/19*
 - c. *Firebase database realtime*
 - d. *Android Studio*
 - e. *Genymotion emulator*
 - f. *Web browser*
 - g. *Koneksi Internet*

3.2.4 Desain

Desain atau perancangan dalam membangun aplikasi tracer study alumni berbasis android yang akan dilakukan adalah sebagai berikut:

1. Perancangan Aplikasi

Pembuatan perancangan menggunakan standar *Unified Modeling Language (UML)* dengan teknik pemrograman berorientasi objek dilakukan agar dalam pengembangan aplikasi akan lebih mudah dikerjakan maupun dikembangkan.

2. Perancangan Tabel

Perancangan tabel pada database nosql untuk menampung data yang akan terkumpul, perancangan tabel harus dilakukan untuk penyimpanan data dari sebuah aplikasi, berikut tabel yang akan dipakai dalam penelitian ini:

- a. Tabel *users_register* berfungsi untuk menyimpan data ketika user melakuka registrasi baru yang nantinya akan di validasi oleh admin data yang berhubungan dengan user_register tersebut seperti : nim, nama, email dan password.
- b. Tabel *users* berfungsi untuk menyimpan data user login sistem yang nantinya aplikasi akan selalu mengecek ke tabel users setiap alumni maupun admin login, seperti : nim, nama, password, akses(0=user, 1=admin) dan color(jika tanpa foto profil).

- c. Tabel alumni berfungsi untuk menyimpan data yang berhubungan dengan data user pengguna alumni. Seperti : nim, nama, angkatan, prodi, email, no_hp, status_bekerja, alama
 - d. Tabel *chat* berfungsi untuk menyimpan data yang berhubungan dengan data user pengguna alumni. Seperti :
 - e. Tabel flp (Forum, Lowongan kerja dan Pengumuman) berfungsi untuk menyimpan data yang berhubungan dengan data user pengguna alumni. Seperti : judul, deskripsi, tgl, tgl_exp, tgl_update, uid, uname
 - f. Tabel *tracerstudy* berfungsi untuk menyimpan data yang berhubungan dengan data hasil kuesioner pengguna. Seperti : nim, kode_jenis, nama_jenis, jawaban, jawaban_lain, keterangan
3. Perancangan *mockup*
- a. *Mockup* tampilan halaman registrasi akses user
 - b. *Mockup* tampilan halaman login akses admin maupun user
 - c. *Mockup* tampilan halaman lupa password akses admin maupun user
 - d. *Mockup* tampilan halaman dashboard
 - 1)*Mockup* tampilan halaman dashboard
 - 2)*Mockup* tampilan halaman data grafik dan pencarian
 - 3)*Mockup* tampilan menu item administrator
 - e. *Mockup* tampilan halaman chat
 - 1)*Mockup* tampilan halaman input dan edit chatting
 - 2)*Mockup* tampilan menu tombol hapus chatting
 - f. *Mockup* tampilan halaman informasi
 - 1)*Mockup* tampilan halaman Angkatan Alumni
 - 2)*Mockup* tampilan halaman Tempat Bekerja Alumni
 - 3)*Mockup* tampilan halaman forum
 - 4)*Mockup* tampilan halaman loker
 - 5)*Mockup* tampilan halaman pengumuman
 - g. *Mockup* tampilan halaman lokasi

- 1) *Mockup* tampilan menu pencarian berdasarkan tahun angkatan dan program studi
- 2) *Mockup* tampilan titik marker jika di klik muncul pesan informasi nim,nama dan angkatan
- h. *Mockup* tampilan halaman profil
 - 1) *Mockup* tampilan menu tombol edit atau perbaharui data
 - 2) *Mockup* tampilan menu tombol set lokasi
 - 3) *Mockup* tampilan menu tombol logout
- i. *Mockup* tampilan halaman administrator
 - 1) *Mockup* tampilan pengelolaan user
 - 2) *Mockup* tampilan data alumni
 - 3) *Mockup* tampilan rekap tracer alumni
 - 4) *Mockup* tampilan fakultas
 - 5) *Mockup* tampilan database backup dan restore

3.2.5 Konstruksi

Setelah tahap perancangan selesai maka tahap selanjutnya adalah pembuatan aplikasi. Konstruksi dalam pembuatan program *tracer* alumni di Fakultas Teknologi Informasi Universitas Bale Bandung adalah sebagai berikut:

1. Firebase Database

Firebase Database digunakan untuk penyimpanan data beserta *plugin* yang sudah disediakan untuk komunikasi antara aplikasi dengan *database* secara *realtime*, notifikasi pada fitur menu pengumuman, chat, forum, dan informasi lowongan kerja.

2. Kotlin

Kotlin adalah *syntax* yang digunakan dalam pembangunan aplikasi android yang terdiri dari hubungan antara *syntax class* dengan tampilan dan *syntax class* dengan database adalah sebagai berikut:

- a. Daftar Aplikasi
- b. Masuk Aplikasi
- c. *Reset Password* Akun Masuk Aplikasi

- d. *Dashboard*
- e. *Chat*
- f. Informasi Angkatan
- g. Informasi Tempat Bekerja Alumni
- h. Informasi Forum
- i. Informasi Loker
- j. Informasi Pengumuman
- k. Lokasi
- l. Profil
- m. *Administrator* Pengelolaan user
- n. *Administrator* Data alumni
- o. *Administrator* Rekap tracer alumni
- p. *Administrator* Fakultas
- q. *Administrator* Database

3. *Xml*

Xml digunakan untuk membuat *layout* tampilan pada aplikasi android sebagai berikut:

- a. Daftar Aplikasi
- b. Masuk Aplikasi
- c. *Reset Password* Akun Masuk Aplikasi
- d. Dashboard
- e. Chat
- f. Informasi Angkatan
- g. Informasi Tempat Bekerja Alumni
- h. Informasi Forum
- i. Informasi Loker
- j. Informasi Pengumuman
- k. Lokasi
- l. Profil
- m. *Administrator* Pengelolaan user
- n. *Administrator* Data alumni
- o. *Administrator* Rekap tracer alumni

- p. *Administrator* Fakultas
- q. *Administrator* Database

3.2.6 Pengujian

Pada umumnya, dimanapun ada hasil implementasi, maka terdapat sebuah pengujian atau testing. Pengujian ini dilakukan pada setiap pembangunan, yaitu : pengujian dilakukan dengan prosedur *Black-box*. kemungkinan kesalahan jika hal tersebut terjadi akan diperbaiki dan kembali lagi pada tahapan konstruksi.

3.2.7 Pelaporan

Pelaporan hasil merupakan tahapan terakhir dimana semua tahapan yang telah dilakukan mulai dari investigasi, analisis masalah, analisis kebutuhan, desain, konstruksi, dan pengujian. Sistematika penulisan skripsi ini yaitu:

Tahap terakhir ini adalah pembuatan laporan menggunakan sistematika penulisan yang terdiri dari 6 bab, sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang informasi umum yaitu latar belakang penelitian, perumusan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, waktu dan tempat penelitian, metodologi penelitian, dan sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini memuat tentang penelitian terkait sebelumnya dan juga dasar teori yang berfungsi sebagai sumber atau alat dalam memahami permasalahan yang berkaitan dengan konsep dasar sistem informasi, dan tentang sistem informasi tracer study.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang pengembangan metodologi yang terdiri dari kerangka pemikiran dan deskripsi.

BAB IV ANALISIS PERANCANGAN DAN HASIL

Bab ini menjelaskan analisis sistem yang sedang berjalan, menjelaskan perancangan sistem yang akan dibuat seperti perancangan struktur proses,

perancangan basis data, dan perancangan aplikasi yang akan dibuat, serta berisi hasil dari pengujian aplikasi yang telah dibuat.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dan pengujian serta hasil pengamatan secara keseluruhan terhadap aplikasi yang akan dibangun.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang memuat tentang jawaban dari pada rumusan masalah pada penelitian. Saran yang ditujukan kepada para pengguna hasil penelitian yang bersangkutan dan kepada peneliti selanjutnya.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

Analisis yang dilakukan pada penelitian ini adalah Analisis masalah, Analisis Kebutuhan dan Perancangan.

4.1.1 Analisis Masalah

Analisis Masalah di Fakultas Teknologi Informasi Universitas Bale Bandung hasil dari wawancara dan observasi mengenai kegiatan *tracer study* alumni yang dilakukan diperoleh dari sistem yang sudah berjalan saat ini adalah sebagai berikut:

1. Kegiatan *tracer study* alumni yang sedang berjalan saat ini menggunakan layanan *google form*.
2. Alumni mengakses link <http://unibba.ac.id/fti/mahasiswa/alumni> untuk memulai mengisi *tracer study* dengan pertanyaan-pertanyaan yang sudah disediakan pada aplikasi.
3. Setelah alumni mengisi *tracer study* di *google form* tersebut bagian kemahasiswaan mengecek hasil dari *tracer study* alumni tersebut.
4. Setelah data sudah divalidasi maka akan diserahkan kebagian akademik untuk digunakan dalam memenuhi salah satu syarat kelengkapan akreditas

Dalam tahap analisis ini memerlukan ketelitian karena apabila dalam tahap ini terdapat kesalahan, maka akan menyebabkan kesalahan pada tahap selanjutnya. Adapun dari hasil analisis sistem yang sudah berjalan terdapat permasalahan atau hambatan yang terjadi adalah sebagai berikut:

1. Setelah alumni mengisi *tracer study* data yang dihasilkan harus diolah kembali oleh bagian staf administrasi FTI UNIBBA jika data sudah valid kemudian diserahkan ke bagian akademik.
2. Memerlukan waktu dan koneksi internet yang stabil untuk mengakses *google form* dalam pengisian *tracer study* alumni yang sudah ada di

lakukan tes pengisian terhadap aplikasi tersebut sesuai dengan tabel berikut:

Tabel 4.1 Waktu pengisian *tracer study*

| NO | Nama User | Waktu pengisian |
|-----------|-----------|-----------------|
| 1 | User 01 | 12 Menit |
| 2 | User 02 | 7 Menit |
| 3 | User 03 | 10 Menit |
| 4 | User 04 | 8 Menit |
| 5 | User 05 | 5 Menit |
| 6 | User 06 | 3 Menit |
| 7 | User 07 | 2.5 Menit |
| Rata-rata | | 6.7 Menit |

4.1.2 Analisis Kebutuhan

Analisi *hardware* dan *software* yang digunakan dalam membantu pembuatan laporan maupun pembangunan terhadap aplikasi ini. Analisis Kebutuhan dari pengembangan sistem informasi *tracer study* alumni ini adalah sebagai berikut:

1. Analisis *Hardware*

Berikut ini merupakan instrumen penelitian pada perangkat keras dengan spesifikasi standar minimum *requirements* yang sudah ditetapkan pada saat awal sebelum membangun aplikasi ini berlangsung dan yang dapat menjadi penunjang dalam proses penelitian ini.

Tabel 4.2 Analisis *Hardware*

| NO | Nama | Spesifikasi |
|----|------------------------------------|--|
| 1 | Laptop | <i>RAM</i> min 4GB <i>HDD</i> min 250 GB |
| 2 | Smartphone | <i>Smartphone</i> android os <i>minimum</i> 4.1 <i>jelly bean</i> |
| 3 | Modem / Wifi / Koneksi Internet | |

2. Analisis Software

Berikut ini merupakan instrumen penelitian pada perangkat lunak yang dapat menjadi penunjang dalam penelitian.

Tabel 4.3 Analisis *software library android dependencies*

| NO | <i>Dependencies</i> | Fungsi |
|----|--|---|
| 1 | <i>implementation fileTree(dir: 'libs', include: ['*.jar'])</i> | Libray bawaan android studio untuk <i>include file jar</i> |
| 2 | <i>implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:\$kotlin_version"</i> | Libray pendukung <i>syntax kotlin</i> pada <i>java android studio</i> |
| 3 | <i>implementation "org.jetbrains.kotlin:kotlin-reflect:\$kotlin_version"</i> | Libray pendukung <i>kotlin</i> |
| 4 | <i>implementation 'com.android.support:appcompat-v7:28.0.0'</i> | Libray untuk <i>actionbar, toolbar</i> dan <i>material design</i> layout aplikasi |
| 5 | <i>implementation 'com.android.support:support-v4:28.0.0'</i> | Libray bawaan android studio pendukung untuk android v4 |
| 6 | <i>implementation 'com.android.support:design:28.0.0'</i> | Libray pendukung desain android |
| 7 | <i>implementation 'com.android.support:recyclerview-v7:28.0.0'</i> | Libray untuk menampilkan data seperti listview |
| 8 | <i>implementation 'com.android.support:cardview-v7:28.0.0'</i> | Libray untuk membuat panel / groupbox |
| 11 | <i>implementation 'com.google.firebaseio:firebase-firebase:11.8.0'</i> | Libray pendukung untuk koneksi aplikasi ke database <i>firebase firestore</i> |
| 12 | <i>implementation 'com.google.firebaseio:firebase-</i> | Libray pendukung untuk notifikasi <i>realtime</i> pada |

| | | |
|----|--|--|
| | <i>messaging:11.8.0'</i> | aplikasi |
| 13 | <i>implementation 'com.firebaseioui:firebase-ui-firebase:3.1.0'</i> | <i>Libray pendukung untuk memanggil fungsi database firebase</i> |
| 14 | <i>implementation 'com.google.android.gms:play-services-maps:11.8.0'</i> | <i>Libray pendukung play service map google</i> |
| 15 | <i>implementation 'com.google.android.gms:play-services-auth:11.8.0'</i> | <i>Libray pendukung play service map authentikasi</i> |
| 16 | <i>implementation 'com.google.android.gms:play-services-location:11.8.0'</i> | <i>Libray pendukung play service lokasi</i> |
| 17 | <i>implementation 'com.google.android.gms:play-services:11.8.0'</i> | <i>Libray pendukung play service berjalan di android</i> |
| 18 | <i>implementation 'com.karumi:dexter:4.2.0'</i> | <i>Libray pendukung untuk memberikan izin akses aplikasi</i> |
| 20 | <i>implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'</i> | <i>Libray untuk tampilan grafik</i> |
| 21 | <i>implementation 'com.android.support:multidex:1.0.3'</i> | <i>Libray pendukung untuk mengatasi Batasan limit file dalvik executable(DEX) 64k method</i> |
| 22 | <i>implementation 'com.github.bumptech.glide:glide:4.9.0' annotationProcessor 'com.github.bumptech.glide:compiler:4.9.0'</i> | <i>Libray tambahan untuk menampilkan gambar pada imageview</i> |
| 24 | <i>implementation 'com.github.chrisbanes:PhotoView:2.0.0'</i> | <i>Libray untuk melihat foto Sperbesar dan perkecil</i> |

| | | |
|----|---|---|
| 25 | <i>implementation 'com.toptooche.searchablespinner:searchable spinnerlibrary:1.3.1'</i> | <i>Libray untuk pencarian di spinner / dropdown box</i> |
|----|---|---|

Tabel 4.4 Analisis Software

| NO | Nama | Fungsi |
|----|---------------------------------------|--|
| 1 | <i>Windows 10, Linux Mint 18 / 19</i> | Operating sistem |
| 2 | <i>Firebase</i> | <i>Server Cloud</i> |
| 3 | <i>Android Studio</i> | <i>Teks editor dan pembuatan layout</i> |
| 4 | <i>Genymotion</i> | <i>Emulator untuk menjalankan aplikasi android</i> |
| 5 | <i>Balsamiq Mockup 3</i> | Pembuatan <i>User Interface</i> |
| 6 | <i>Start UML 2.6</i> | Pembuatan <i>UML</i> |
| 7 | <i>Postman</i> | Untuk melakukan pengetesan ke <i>database</i> dengan method <i>GET, POST</i> dan <i>DELETE</i> |
| 8 | <i>Mozilla Firefox</i> | <i>Web Browser</i> |
| 9 | <i>Microsoft Office 2016</i> | Pembuatan Laporan |
| 10 | <i>Microsoft Visio 2007</i> | Pembuatan Pemodelan Gambar Sistem |

3. Analisis Pengguna

Analisis pengguna dalam sistem aplikasi *tracer study* memiliki hak akses masing-masing fitur / fungsional disesuaikan dengan kebutuhan *user* adalah sebagai berikut:

a. Admin

Admin adalah pengguna aplikasi memiliki hak akses penuh pada fitur aplikasi, baik *input, edit* maupun *delete*, validasi *user* baru yang mendaftar pada aplikasi, melihat laporan data alumni FTI UNIBBA dan melihat hasil rekap *tracer study* alumni yang telah dilaksanakan oleh mahasiswa FTI UNIBBA.

b. Alumni

Alumni adalah pengguna aplikasi memiliki hak akses terbatas pada fitur aplikasi hanya bisa akses untuk registrasi, mengisi kuesioner *tracer study*, posting forum, posting lowongan pekerjaan, perbaharui informasi biodata dan status pekerjaan.

Hak akses antara admin dengan alumni dibedakan dari fungsional fitur pada aplikasi berikut tabel klasifikasinya:

Tabel 4.5 Klasifikasi Pengguna pada aplikasi

| No | Menu | Akses | |
|----|--|-------|--------|
| | | Admin | Alumni |
| 1 | Dashboard | ✓ | ✓ |
| 2 | Chat | ✓ | ✓ |
| 3 | Informasi Angkatan | ✓ | ✓ |
| 4 | Informasi Tempat Bekerja Alumni | ✓ | ✓ |
| 5 | Informasi Forum | ✓ | ✓ |
| 6 | Informasi Loker | ✓ | ✓ |
| 7 | Informasi Pengumuman | ✓ | ✓ |
| 8 | Lokasi | ✓ | ✓ |
| 9 | Profil | ✓ | ✓ |
| 10 | Administrator Pengelolaan <i>user</i> | ✓ | ✗ |
| 11 | Administrator Data Alumni | ✓ | ✗ |
| 12 | Administrator Rekap <i>Tracer</i> Alumni | ✓ | ✗ |
| 13 | Administrator Fakultas | ✓ | ✗ |
| 14 | Administrator Database (<i>Backup & Restore</i>) | ✓ | ✗ |

4. Analisis User Interface

Fitur-fitur yang terdapat pada tracer study alumni bertujuan untuk mempermudah dalam melakukan pengisian *tracer* alumni. Untuk akses maupun fungsi dalam aplikasi tersebut dibedakan antara admin dan user adalah sebagai berikut:

1. Fungsi akses aplikasi pada admin

a. Fungsi untuk melakukan login

- b. Fungsi untuk melakukan logout
- c. Fungsi untuk mengelola data alumni
- d. Fungsi mengelola data alumni adalah fungsi untuk admin untuk mengelola semua data yang berhubungan dengan data alumni. Pada Fungsi mengelola data alumni terdapat beberapa bagian yaitu:
 - 1) Fungsi menambahkan data alumni yang diinputkan oleh admin melalui *form* pengelolaan *user* kemudian di simpan kedalam *database*.
 - 2) Fungsi mengubah data alumni berfungsi untuk melakukan perubahan data email login alumni yang bertujuan untuk *reset password*.
 - 3) Fungsi menghapus data alumni berfungsi menghapus dari *database* berdasarkan data yang dipilih.
 - 4) Fungsi pencarian data alumni berfungsi untuk melakukan pencarian data alumni yang terdapat pada *database* berdasarkan kriteria yang diinputkan.
 - 5) Fungsi validasi data alumni berfungsi untuk validasi alumni jika data sesuai dan jika tidak maka admin akan menolak data tersebut.
 - 6) Fungsi cetak data ke pdf berfungsi untuk menampilkan laporan data alumni dan rekapitulasi hasil *tracer* alumni.
 - 7) Fungsi tampilan informasi *dashboard* jumlah berdasarkan kategori pembelajaran dan pekerjaan dalam bentuk grafik.
- 2. Fungsi akses aplikasi pada alumni
 - a. Fungsi untuk melakukan registrasi
 - b. Fungsi untuk pengisian maupun perubahan data kuesioner
 - c. Fungsi untuk melakukan chat sesama pengguna aplikasi
 - d. Fungsi untuk melakukan aktifitas yang terdiri dari input, ubah dan hapus pada informasi forum, lowongan pekerjaan dan melihat list data alumni
 - e. Fungsi Lokasi melakukan pencarian alumni berdasarkan posisi lokasi *maps google*

- f. Fungsi untuk melakukan perbaharui profil data alumni dan set lokasi alumni untuk menampilkan data tersebut pada menu lokasi.

5. Analisis Biaya

Analisis biaya yang diperlukan untuk membuat aplikasi *tracer study* alumni adalah sebagai berikut:

Tabel 4.6 Analisis Biaya

| NO | Keperluan | Durasi | Biaya |
|----|------------------------------|----------|-------------|
| 1 | Akun <i>Google Playstore</i> | Lifetime | Rp. 450,000 |
| 2 | Kuota Internet Telkomsel | 60 Hari | Rp. 130,000 |

4.2 Perancangan

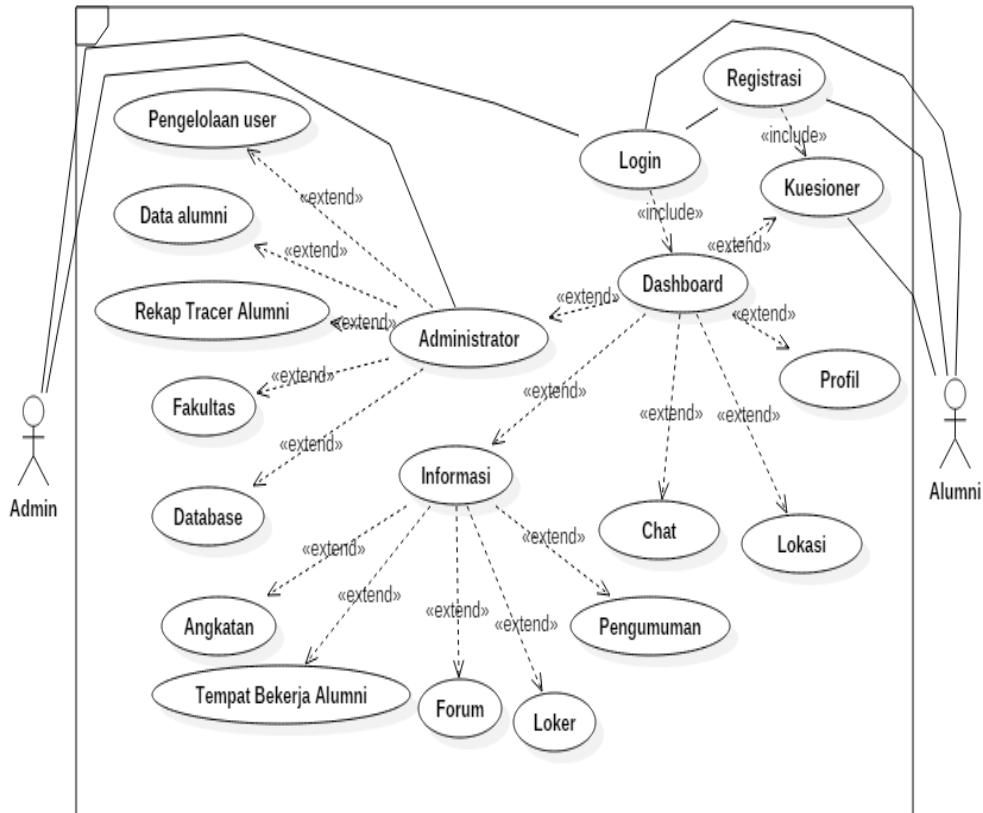
Perancangan penelitian ini merupakan tahapan setelah analisis masalah dan analisis kebutuhan untuk mendefinisikan kebutuhan sistem fungsional dan *user interface* yang menggambarkan bagaimana suatu aplikasi akan dikembangkan dan dibentuk sesuai dengan gambar pemodelan yang dibuat. Perancangan model aplikasi yang digunakan dalam sistem pemodelan yang akan dibangun menggunakan UML (*Unified Modeling Language*) dan untuk memberikan gambaran pada penelitian ini meliputi menu *Diagram model*, *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Sequence Diagram* dan *flowmap*.

4.2.1 Use Case Diagram

Diagram *use case* dari aplikasi *tracer study* alumni FTI UNIBBA bertujuan untuk memberikan gambaran interaksi aktor dengan sistem aplikasi tracer alumni. Berdasarkan *use case* yang diusulkan dapat dilihat pada Gambar 4.1 *Use Case Diagram*, admin dapat melakukan login ke sistem dan mengakses semua fitur menu untuk mengelola data *tracer* alumni , pada menu *dashboard* menampilkan halaman monitoring data alumni dalam bentuk grafik, melihat melihat user yang terdaftar disistem dan laporan data alumni beserta rekap *tracer* alumni. Pengelolaan user digunakan admin untuk mengetahui user baru yang mendaftar

kemudian admin akan memproses validasi dan jika data tidak sesuai admin juga dapat melakukan penolakan.

Sedangkan untuk *user* dapat melakukan registrasi, melakukan pengisian kuesioner *tracer study* dan jika sudah melakukan registrasi atau daftar user dapat langsung melakukan login masuk aplikasi.



Gambar 4.1 Use Case Diagram

Skenario dari aktor admin dalam *use case* diagram aplikasi *tracer alumni* adalah sebagai berikut:

1. Skenario *Use Case Login – Admin*

Pada Skenario *Use Case Login – Admin* yaitu admin jika belum punya dapat mengunduh aplikasi tersebut di *playstore* dengan kata kunci nama “*Tracer Alumni Unibba*” untuk mendownload aplikasi tersebut sampai berhasil terinstal, admin dapat melakukan login dengan *email* dan *password* yang sudah tersimpan dalam *database* jika belum harus diregistrasi oleh admin dan di validasi dengan akses level admin.

Reaksi sistem setelah melakukan validasi jika data benar, data akan dikirim ke database kemudian melakukan verifikasi pencocokan data, jika data

sesuai sistem akan memperbolehkan masuk yang nantinya akan diarahkan pada halaman dashboard.

Gambar 4.2 Skenario *Use Case Login - Admin*

| Identifikasi | |
|---|--|
| Nama | Login |
| Deskripsi | Tampilan Form Login |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi Awal | Tampilan Form Login |
| Aksi Aktor Reaksi Sistem | |
| 1. Menuju <i>google playstore</i> dan cari “Tracer Alumni Unibba” jika aplikasi belum terinstal | |
| 2. Memasukan <i>Email</i> dan <i>Password</i> | |
| | 3. Validasi data jika data sesuai akan dikirim dan diverifikasi ke tabel users untuk dicocokan |
| | 4. Masuk ke aplikasi |

2. Skenario *Use Case Administrator – Admin*

Pada Skenario Use Case Dashboard – Admin yaitu admin dapat melihat data grafik hasil *tracer* alumni dan menu pengguna *user* baru mendaftar, belum melakukan *tracer study*, admin dan *user* alumni dan pencarian alumni untuk memudahkan admin dalam mencari alumni yang ditampilkan bersama pada halaman *dashboard* di khususkan oleh admin yang telah berhasil *login* masuk pada sistem.

Reaksi sistem menampilkan halaman *dashboard* kemudian *database* menyiapkan data terbaru yang nantinya akan disajikan ke dalam bentuk grafik dan tabel pengumuman. Admin memiliki fitur tambahan di menu *dashboard* dengan menekan menu dan memilih lainnya didalamnya ada menu pengelolaan *user*, data alumni, rekap *tracer* alumni, fakultas dan *database backup restore*.

Tabel 4.7 Skenario *Use Case Administrator* – Admin

| Identifikasi | |
|----------------------------------|---|
| Nama | Administrator |
| Deskripsi | Tampilan Form Administrator |
| Aktor | Admin |
| Skenario Utama | |
| Kondisi Awal | Tampilan Form Administrator |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih menu Administrator | <p>2. Memeriksa status login</p> <p>3. Database menyiapkan data dan menampilkan halaman administrator yang berisi menu</p> <ul style="list-style-type: none"> a) Pengelolaan <i>User</i> b) Data Alumni c) Rekap <i>Tracer Alumni</i> d) Fakultas e) <i>Database</i> |

Skenario dari aktor Alumni dalam *use case diagram* aplikasi tracer alumni adalah sebagai berikut:

1. Skenario *Use Case* Registrasi – Alumni

Pada Skenario *Use Case* Registrasi – Alumni yaitu Alumni mengunduh aplikasi tersebut di *playstore* dengan mengakses nama “*Tracer Alumni Unibba*” setelah berhasil terinstal aksi aktor memilih menu registrasi.

Reaksi sistem akan menampilkan form registrasi, aktor mengisi form registrasi, jika *field* sudah diisi oleh aktor kemudian menekan tombol daftar, maka reaksi sistem akan validasi form registrasi jika data *field* sesuai maka sistem akan mengirim data tersebut kedatabase kemudian user dialihkan pada *form* menunggu validasi, jika admin validasi maka alumni akan dialihkan pada halaman untuk memulai dan melakukan kuesioner, jika admin menolak maka alumni akan dialihkan pada halaman gagal validasi.

Tabel 4.8 Skenario *Use Case* Registrasi – Alumni

| Identifikasi | |
|--|--|
| Nama | Registrasi |
| Deskripsi | Tampilan Form Registrasi |
| Aktor | Alumni |
| Skenario Utama | |
| Kondisi Awal | Tampilan Form Registrasi |
| Aksi Aktor | Reaksi Sistem |
| 1. Menuju <i>google playstore</i> dan cari “ <i>Tracer Alumni Unibba</i> ” jika aplikasi belum terinstal | |
| | 2. Menampilkan form registrasi |
| 3. Mengisi form registrasi | |
| | 4. Validasi form registrasi jika data field tidak diisi maka akan memberikan notifikasi pada field yang belum diisi tersebut |
| | 5. Mengirim data ke database untuk kemudian disimpan |
| | 6. Menampilkan halaman mengunggu divalidasi |

2. Skenario *Use Case* Kuesioner – Alumni

Pada Skenario *Use Case* Kuesioner – Alumni yaitu Alumni dapat melakukan pengisian kuesioner jika *user* tersebut sudah divalidasi oleh admin kemudian menuju ke menu kuesioner aplikasi.

Aksi aktor yaitu menekan tombol mulai dan sistem akan menampilkan halaman kuesioner yang terdiri dari pengisian data pribadi, pengalaman metode penekanan pembelajaran, kepuasan pengguna lulusan dan riwayat pekerjaan.

Reaksi sistem menampilkan menu kuesioner dan setiap kategori tersebut sistem melakukan validasi inputan jika data yang harusnya diisi tidak diisi

maka muncul notifikasi field harus diisi, memeriksa data valid tidaknya setiap pilihan atau inputan yang dilakukan user dalam pengisian kuesioner. Jika data valid maka data akan disimpan ke *database* dan jika pengisian data kuesioner semua kategori sudah diisi dan menampilkan *icon checklist* bertanda sudah diisi semua maka sistem akan menyimpan ke *database* dan akan memperbolehkan masuk yang nantinya akan diarahkan pada halaman dashboard.

Tabel 4.9 *Use Case* Kuesioner – Alumni

| Identifikasi | |
|---|--|
| Nama | Kuesioner |
| Deskripsi | Tampilan Form Kuesioner |
| Aktor | Alumni |
| Skenario Utama | |
| Kondisi Awal | Tampilan Form Kuesioner |
| Aksi Aktor | |
| 1. Menekan tombol mulai | |
| | 2. Menampilkan halaman kuesioner |
| 3. Memilih menu pengisian kuesioner: <ul style="list-style-type: none"> a) Data pribadi b) Pengalaman Metode Penekanan Pembelajaran c) Kepuasan Pengguna Lulusan d) Riwayat Pekerjaan | |
| | 4. Menampilkan halaman sesuai yang dipilih |
| | 5. Validasi data pilihan atau inputan |
| | 6. Menyimpan data kedatabase |
| | 7. Menampilkan halaman <i>dashboard</i> |

3. Skenario *Use Case Login – Alumni*

Pada Skenario *Use Case Login – Alumni* yaitu Alumni jika belum punya aplikasi dapat mengunduh aplikasi tersebut di *playstore* dengan mengakses kata kunci nama “*Tracer Alumni Unibba*” setelah berhasil terinstal aksi aktor memilih menu login pada aplikasi kemudian user mengisi email dan *password* pada form login tersebut, setelah data inputan sudah terisi kemudian *user* menekan tombol *login*.

Reaksi sistem setelah melakukan validasi terhadap *field* yang diisi kemudian verifikasi pencocokan data *kedatabase* jika data benar sistem akan memperbolehkan masuk yang nantinya akan diarahkan pada halaman *dashboard* dan jika tidak maka sistem memunculkan notifikasi *error*.

Tabel 4.10 Skenario *Use Case Login – Alumni*

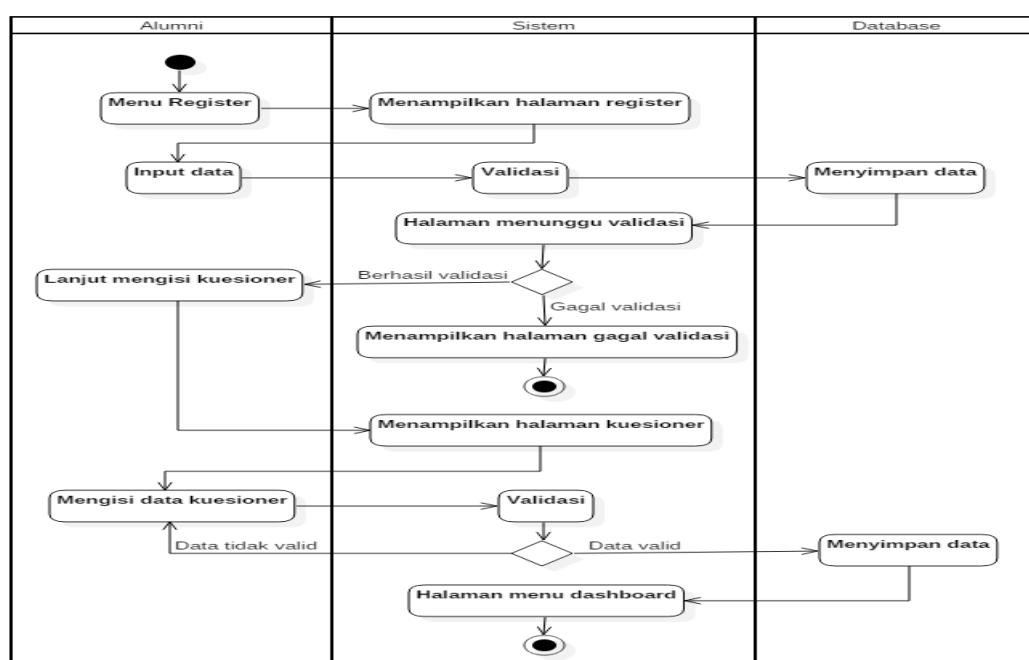
| Identifikasi | |
|---|---|
| Nama | Login |
| Deskripsi | Tampilan Form Login |
| Aktor | Alumni |
| Skenario Utama | |
| Kondisi Awal | Tampilan Form Login |
| Aksi Aktor | Reaksi Sistem |
| 1. 1. Menuju <i>google playstore</i> dan cari “ <i>Tracer Alumni Unibba</i> ” jika aplikasi belum terinstal | |
| 2. Memilih menu login dan memasukan <i>Email</i> dan <i>Password</i> | |
| | 3. Validasi data jika data tidak sesuai akan menampilkan pesan gagal <i>login</i> dan jika data sesuai akan dikirim kemudian diverifikasi ke tabel <i>users</i> untuk dicocokan |
| | 4. Masuk ke aplikasi |

4.2.2 Activity Diagram

Pada sistem Berikut *activity diagram tracer* alumni yang menunjukan aktivitas yang dilakukan *user* dan timbal balik dari sistem terhadap *user* secara sistematis. *Activity Diagram* dibagi menjadi 2 bagian yaitu admin dan alumni berikut adalah *Activity Diagram* Alumni:

1. *Activity Diagram* Alumni Registrasi

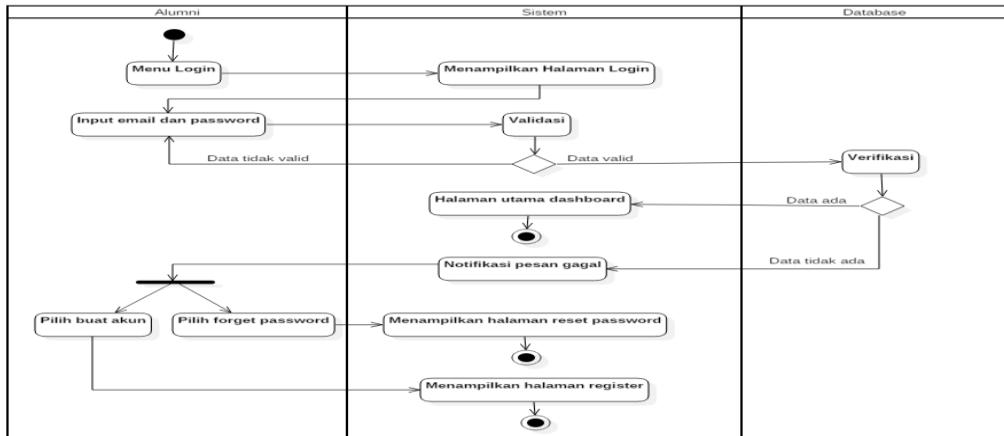
Aktifitas Registrasi di mulai dengan alumni masuk ke menu registrasi atau daftar aplikasi merespon dengan menampilkan *form* registrasi, dan user mengisi nim, nama, email, *password* dan ulangi *password*. Jika Setelah itu di simpan ke *database* dan menampilkan Jika *user* sudah terdaftar maka menampilkan notifikasi *user* sudah terdaftar atau salah inputan sistem akan menampilkan notifikasi gagal.



Gambar 4.3 *Activity Diagram* Alumni Registrasi

2. *Activity Diagram* Alumni Login

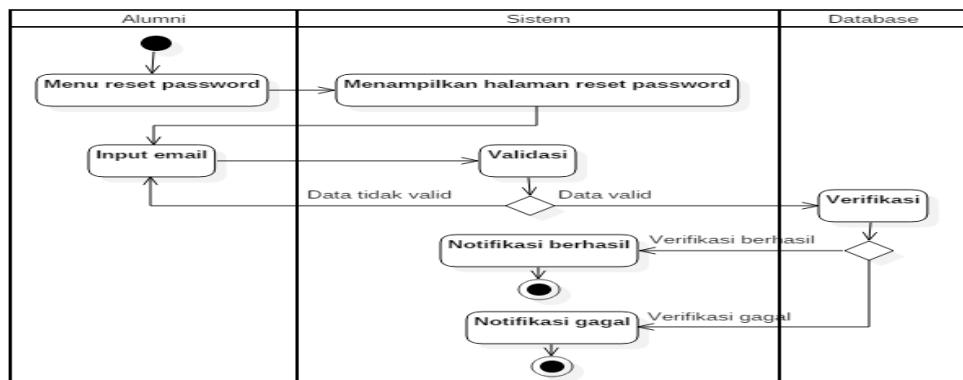
Aktifitas ini dimulai dengan alumni melakukan interaksi dengan Aplikasi dengan melakukan aktifitas login. Form login menginputkan *email* dan *password* setelah itu di validasi jika tidak valid user kembali menginputkan *email* dan *password* , jika valid langsung di verifikasi *database*, jika ditemukan berarti langsung masuk ke form dashboard. Jika tidak di temukan menampilkan notifikasi *email / password* salah atau *email / password* tidak sesuai.



Gambar 4.4 Activity Diagram Alumni Login

3. Activity Diagram Alumni Reset Password

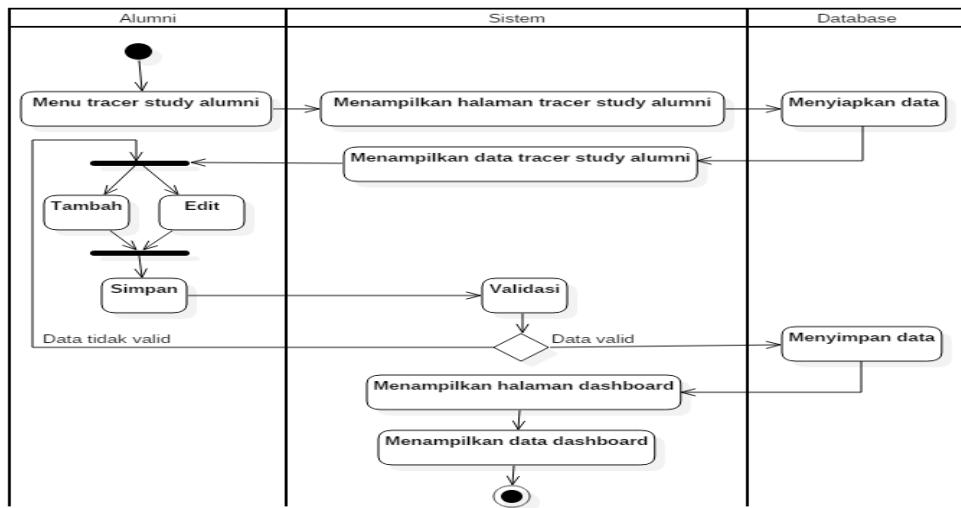
Aktifitas *Reset Password* di mulai dengan Alumni masuk ke menu login dan memilih *forgot password* tersebut meliputi data *email*. Aplikasi merespon dengan menampilkan halaman *reset password*, dan alumni dapat mengisi email melakukan *reset password* jika data benar maka sistem akan validasi dan mengirim ke email tersebut, Jika tidak maka akan menampilkan notifikasi pesan gagal email tidak sesuai atau tidak terdaftar.



Gambar 4.5 Activity Diagram Alumni Reset Password

4. Activity Diagram Alumni Kuesioner

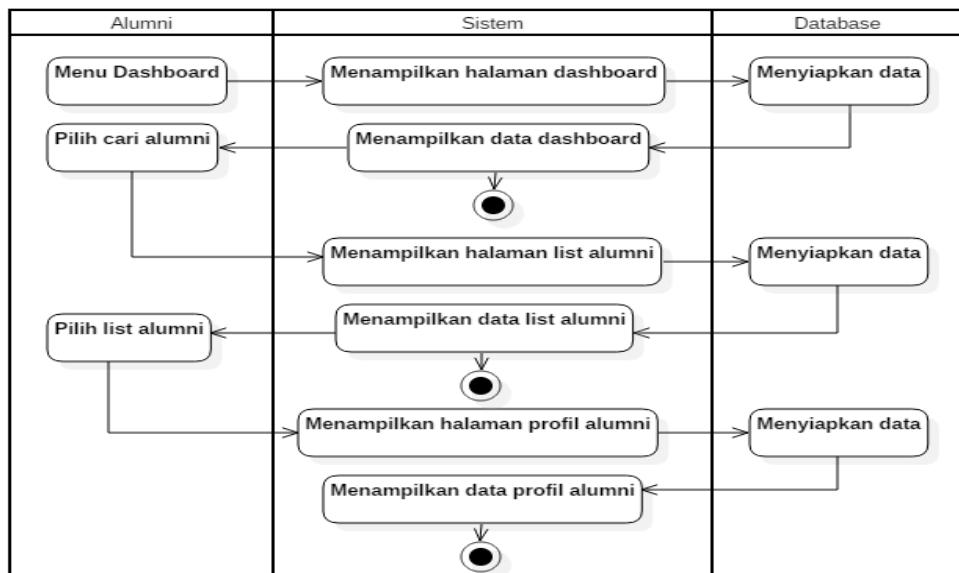
Aktifitas Kuesioner di mulai dengan alumni masuk ke menu kuesioner terdapat sub menu dalam menu kuesioner meliputi data pribadi, pembelajaran dan pekerjaan. Aplikasi merespon dengan menampilkan halaman kuesioner, dan Alumni dapat melakukan pengisian kuesioner dengan pertanyaan – pertanyaan yang sudah disediakan jika data benar maka sistem akan validasi dan menyimpan data kedatabase. Jika tidak maka akan notifikasi pesan pengisian yang terlewatkan harus diisi.



Gambar 4.6 Activity Diagram Alumni Kuesioner

5. Activity Diagram Alumni Dashboard

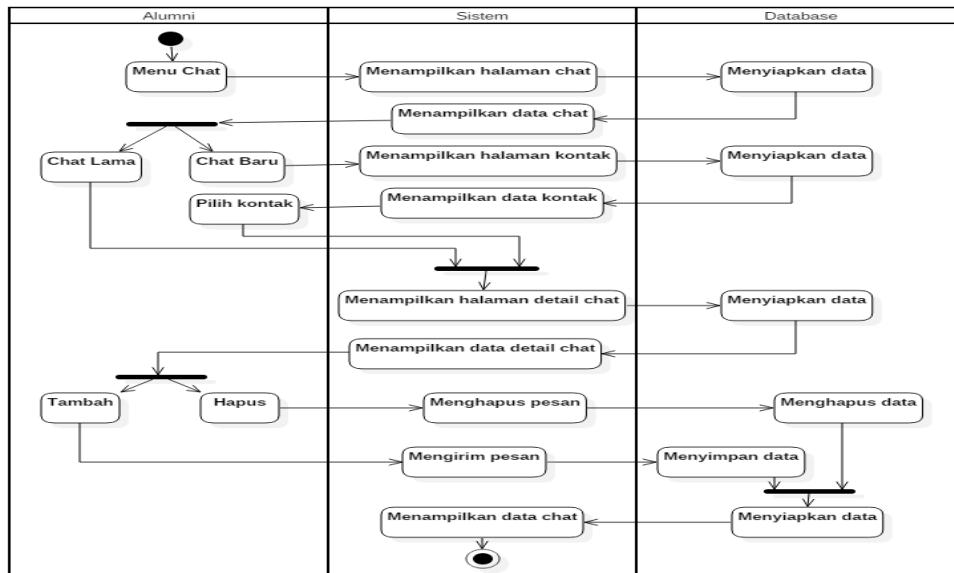
Aktifitas *Dashboard* di mulai dengan *user* masuk ke menu *dashboard* aplikasi mengirimkan *request* ke *database*, *database* merespon dengan menyiapkan data kemudian oleh sistem ditampilkan dalam tampilan pencarian alumni dan pengumuman.



Gambar 4.7 Activity Diagram Alumni Dashboard

6. Activity Diagram Alumni Chat

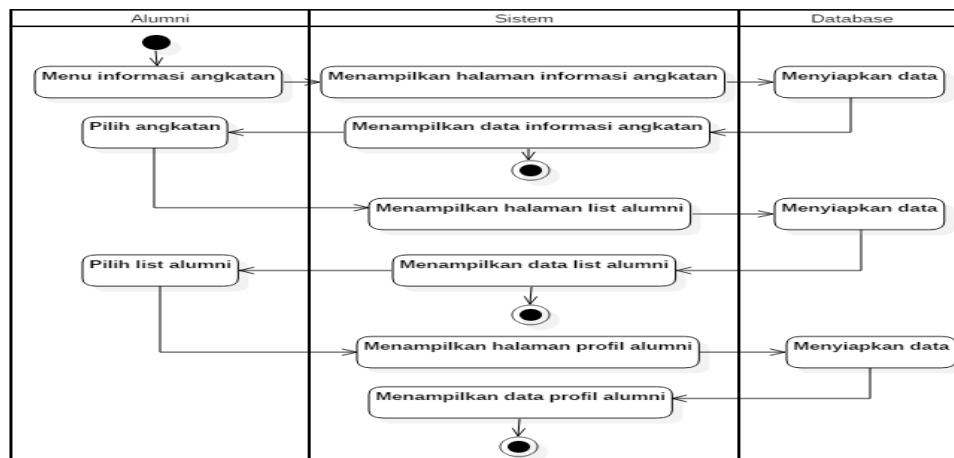
Aktifitas Kuesioner di mulai dengan *user admin* dan *alumni* masuk ke menu chat. Aplikasi merespon dengan menampilkan halaman chat, dapat mengirim atau menerima pesan sesama pengguna aplikasi. Jika ada pesan baru masuk maka aplikasi menampilkan notifikasi pemberitahuan chat atau pesan baru.



Gambar 4.8 Activity Diagram Alumni Chat

7. Activity Diagram Alumni Informasi Angkatan

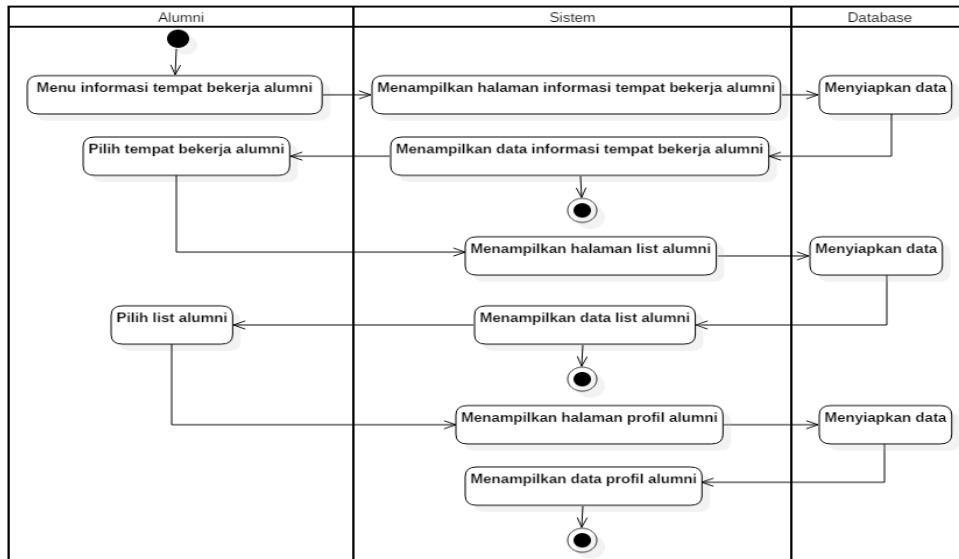
Aktifitas Informasi Angkatan di mulai dengan alumni masuk ke menu informasi dan memilih tab angkatan. Aplikasi merespon dengan menampilkan halaman angkatan dan list tahun angkatan alumni ketika dipilih list tersebut akan menampilkan list alumni sesuai list yang dipilih.



Gambar 4.9 Activity Diagram Alumni Angkatan

8. Activity Diagram Alumni Informasi Tempat Bekerja Alumni

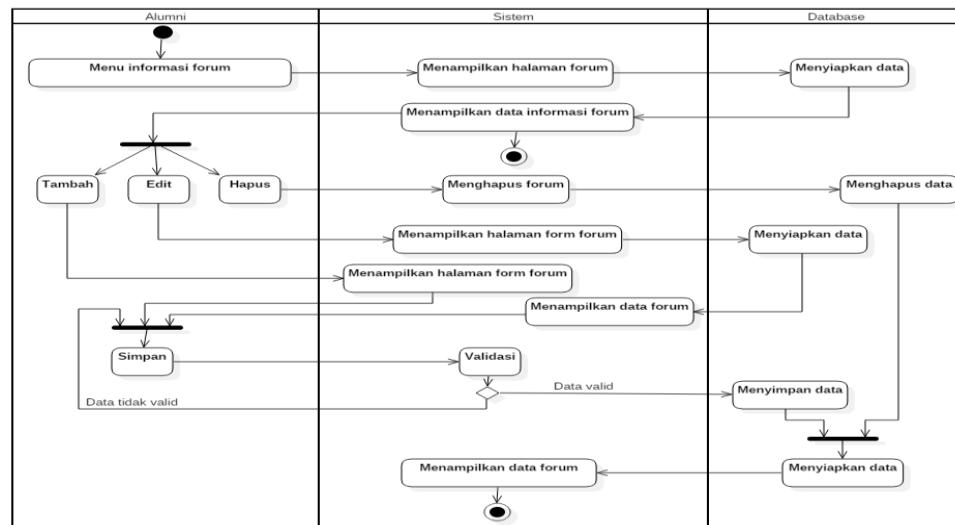
Aktifitas Informasi Tempat Bekerja Alumni di mulai dengan alumni masuk ke menu informasi dan memilih tab tempat bekerja alumni. Aplikasi merespon dengan menampilkan halaman tempat bekerja alumni dan list tempat bekerja alumni ketika dipilih list tersebut akan menampilkan list alumni sesuai list yang dipilih.



Gambar 4.10 Activity Diagram Alumni Tempat Bekerja Alumni

9. Activity Diagram Alumni Informasi Forum

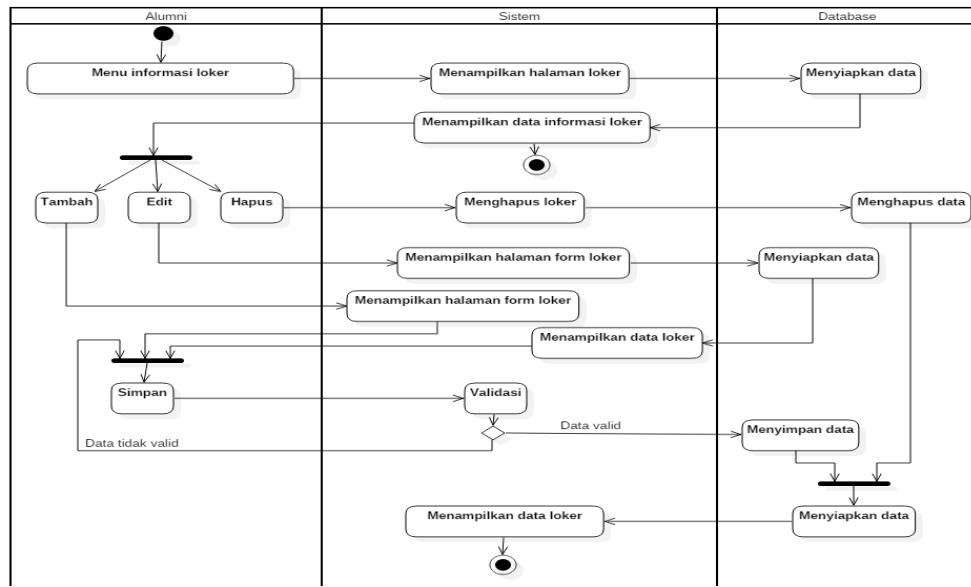
Aktifitas Informasi Forum di mulai dengan alumni masuk ke menu informasi dan memilih tab forum. Aplikasi merespon dengan menampilkan halaman forum, isi list forum terbaru kemudian ketika dipilih akan ditampilkan ke *form* detail dan *user* dapat melakukan tambah, edit dan hapus sesuai *user* login.



Gambar 4.11 Activity Diagram Alumni Forum

10. Activity Diagram Alumni Informasi Loker

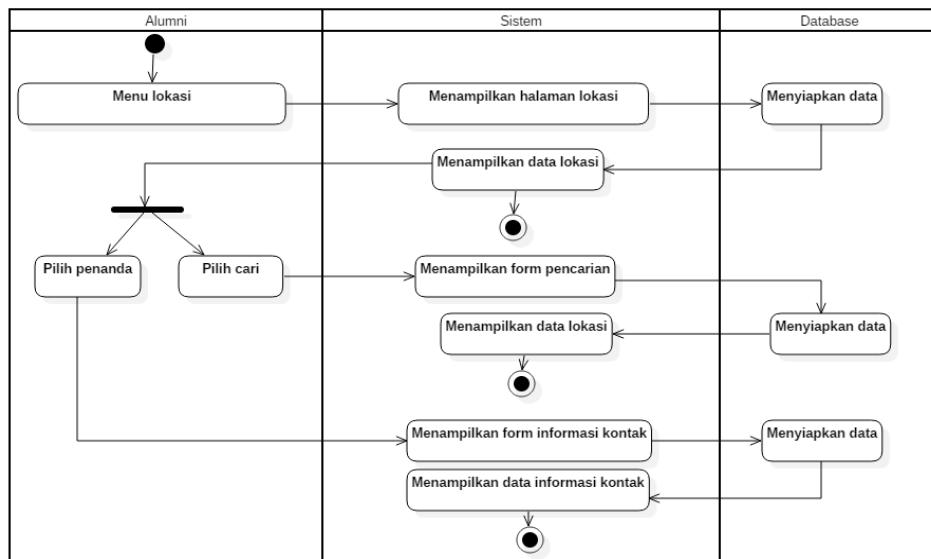
Aktifitas Informasi Loker di mulai dengan alumni masuk ke menu informasi dan memilih tab Loker. Aplikasi merespon dengan menampilkan halaman Loker, isi list Loker terbaru kemudian ketika dipilih akan ditampilkan ke *form* detail dan *user* dapat melakukan tambah, edit dan hapus sesuai *user* login.



Gambar 4.12 Activity Diagram Alumni Loker

11. Activity Diagram Alumni Lokasi

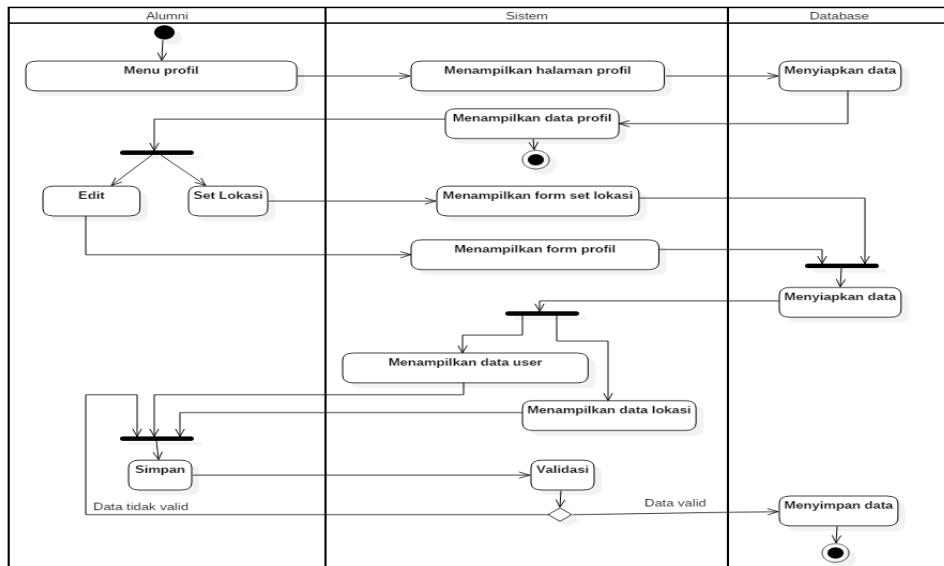
Aktifitas Lokasi dimulai dengan alumni masuk ke menu lokasi terdapat menu pencarian data sesuai kriteria angkatan dan program studi. Aplikasi merespon dengan menampilkan halaman map lokasi keberadaan alumni jika sudah perbaharui data dan set lokasi.



Gambar 4.13 Activity Diagram Alumni Lokasi

12. Activity Diagram Alumni Profil

Aktifitas Profil dimulai dengan admin maupun user masuk ke menu profil aplikasi merespon dengan menampilkan halaman profil, dan dapat melakukan pembaharuan informasi data alumni dan kuesioner.

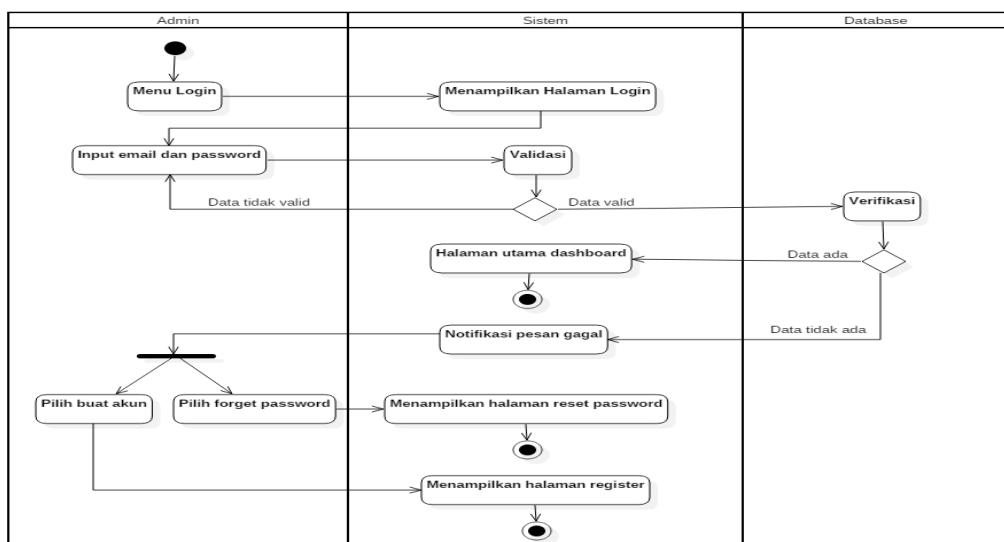


Gambar 4.14 Activity Diagram Alumni Profil

Sedangkan untuk *Activity Diagram Admin* yang berjalan dalam sistem adalah sebagai berikut:

1. *Activity Diagram Admin Login*

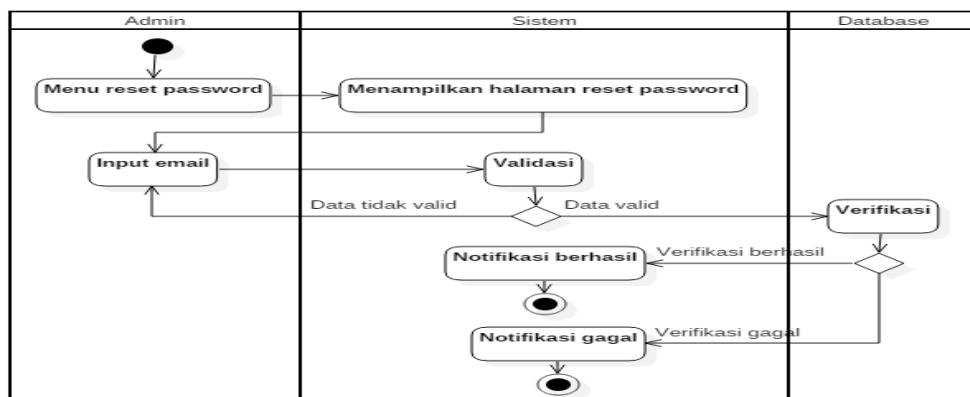
Aktifitas ini dimulai dengan admin melakukan interaksi dengan Aplikasi dengan melakukan aktifitas login. *Form* login menginputkan *email* dan *password* setelah itu di validasi jika tidak valid user kembali menginputkan email dan password , jika valid langsung di verifikasi database, jika ditemukan berarti langsung masuk ke form dashboard. Jika tidak di temukan menampilkan notifikasi *email / password* salah atau *email / password* tidak sesuai dan admin dapat melakukan *reset password* dengan cara klik *forgot password*.



Gambar 4.15 Activity Diagram Admin Login

2. Activity Diagram Admin Reset Password

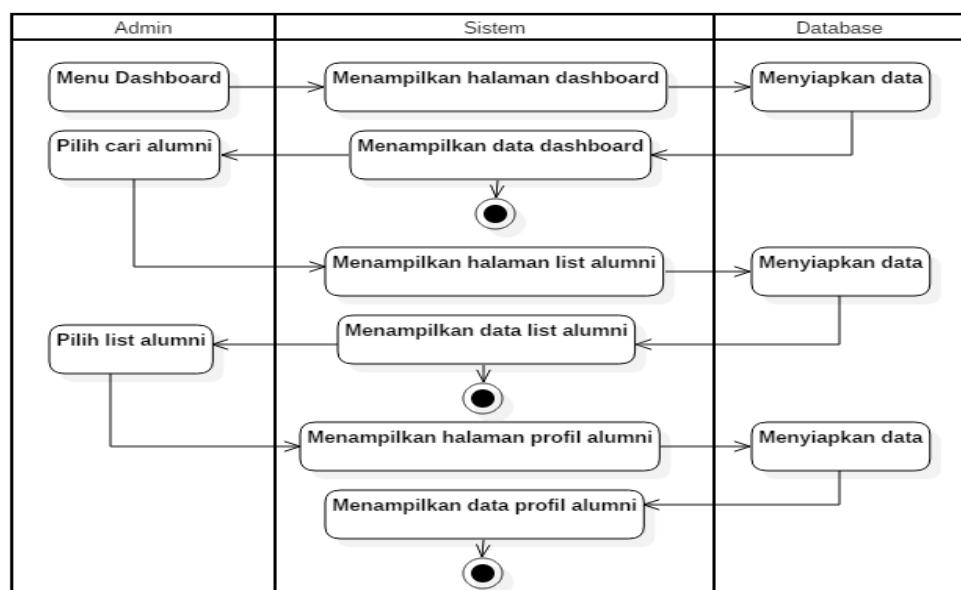
Aktifitas Admin *Reset Password* di mulai dengan Admin masuk ke menu *login* dan memilih *forgot password* tersebut meliputi data *email*. Aplikasi merespon dengan menampilkan halaman *reset password*, dan alumni dapat mengisi *email* melakukan *reset password* jika data benar maka sistem akan validasi dan mengirim ke email tersebut, Jika tidak maka akan menampilkan notifikasi pesan gagal email tidak sesuai atau tidak terdaftar.



Gambar 4.16 Activity Diagram Admin Reset Password

3. Activity Diagram Admin Dashboard

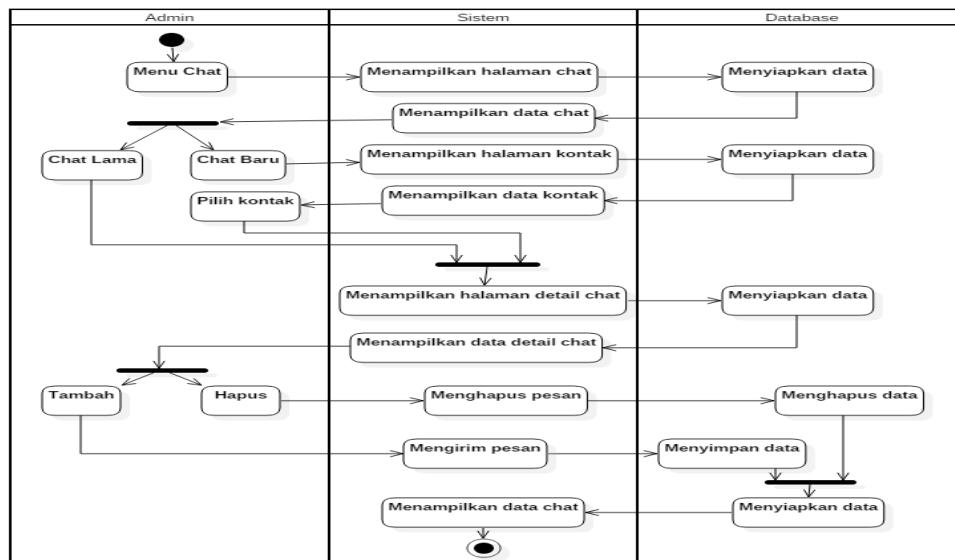
Aktifitas Dashboard di mulai dengan user masuk ke menu *dashboard* aplikasi mengirimkan request ke database, database merespon dengan menyiapkan data kemudian oleh sistem ditampilkan dalam tampilan pencarian alumni dan grafik hasil dari pengisian *tracer* alumni.



Gambar 4.17 Activity Diagram Admin Dashboard

4. Activity Diagram Admin Chat

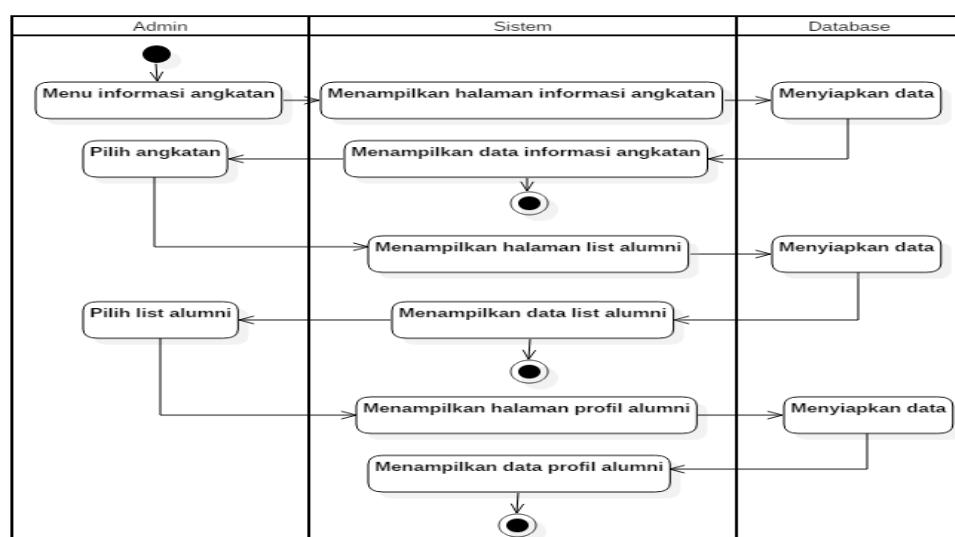
Aktifitas Kuesioner di mulai dengan admin masuk ke menu chat. Aplikasi merespon dengan menampilkan halaman chat, dapat mengirim atau menerima pesan sesama pengguna aplikasi. Jika ada pesan baru masuk maka aplikasi menampilkan notifikasi pemberitahuan chat atau pesan baru.



Gambar 4.18 Activity Diagram Admin Chat

5. Activity Diagram Admin Angkatan

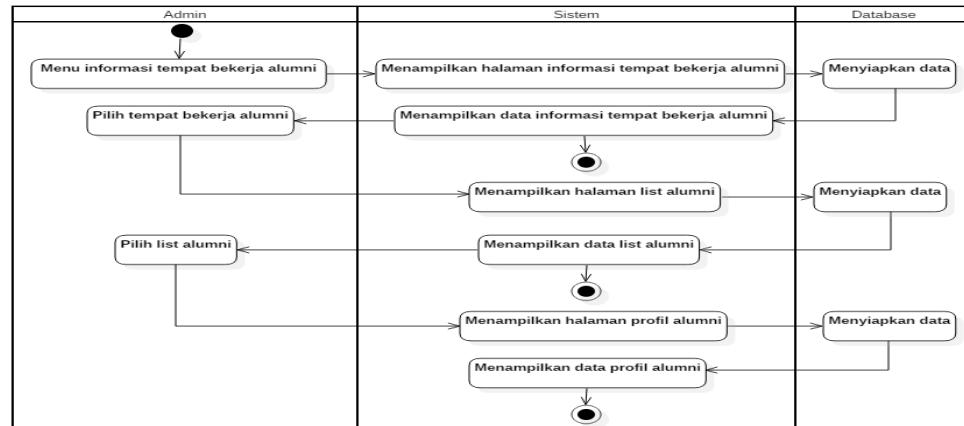
Aktifitas Informasi Angkatan di mulai dengan admin masuk ke menu informasi dan memilih tab angkatan. Aplikasi merespon dengan menampilkan halaman angkatan dan list tahun angkatan alumni ketika dipilih list tersebut akan menampilkan list alumni sesuai list yang dipilih.



Gambar 4.19 Activity Diagram Admin Angkatan

6. Activity Diagram Admin Informasi Tempat Bekerja Alumni

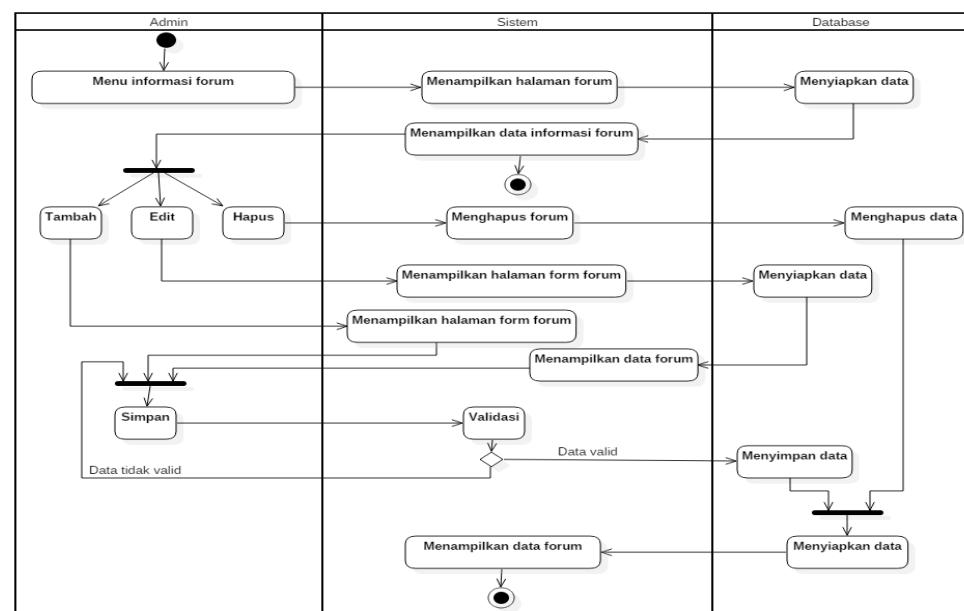
Aktifitas Informasi Tempat Bekerja Alumni di mulai dengan admin masuk ke menu informasi dan memilih tab tempat bekerja alumni. Aplikasi merespon dengan menampilkan halaman tempat bekerja alumni dan list tempat bekerja alumni ketika dipilih list tersebut akan menampilkan list alumni sesuai list yang dipilih.



Gambar 4.20 Activity Diagram Admin Tempat Bekerja Alumni

7. Activity Diagram Admin Informasi Forum

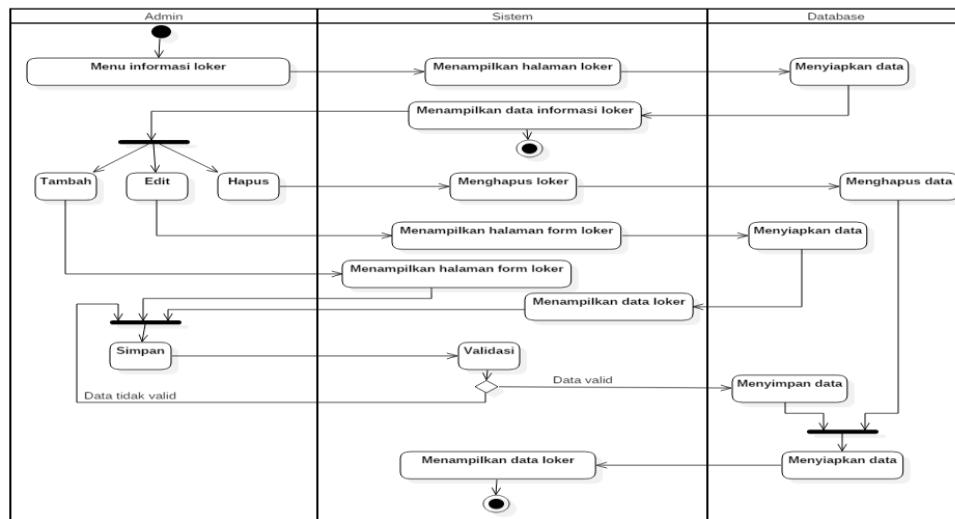
Aktifitas Informasi Forum di mulai dengan admin masuk ke menu informasi dan memilih tab forum. Aplikasi merespon dengan menampilkan halaman forum, isi list forum terbaru kemudian ketika dipilih akan ditampilkan ke form detail dan user dapat melakukan tambah, edit dan bisa menghapus semua.



Gambar 4.21 Activity Diagram Admin Forum

8. *Activity Diagram* Admin Informasi Loker

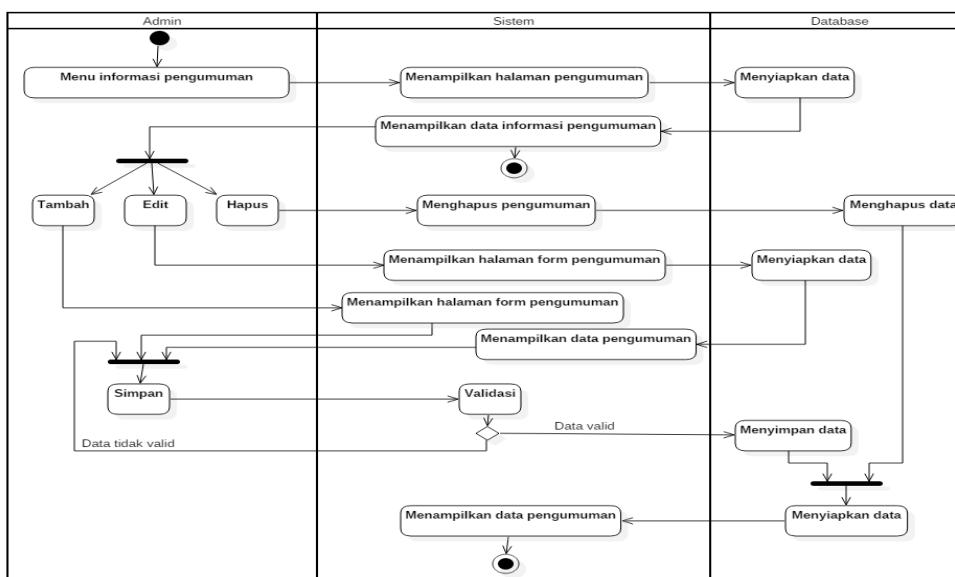
Aktifitas Informasi Loker di mulai dengan admin masuk ke menu informasi dan memilih tab Loker. Aplikasi merespon dengan menampilkan halaman loker, isi list loker terbaru kemudian ketika dipilih akan ditampilkan ke form detail dan user dapat melakukan tambah, edit dan bisa menghapus semua.



Gambar 4.22 *Activity Diagram Admin Loker*

9. *Activity Diagram Admin Informasi Pengumuman*

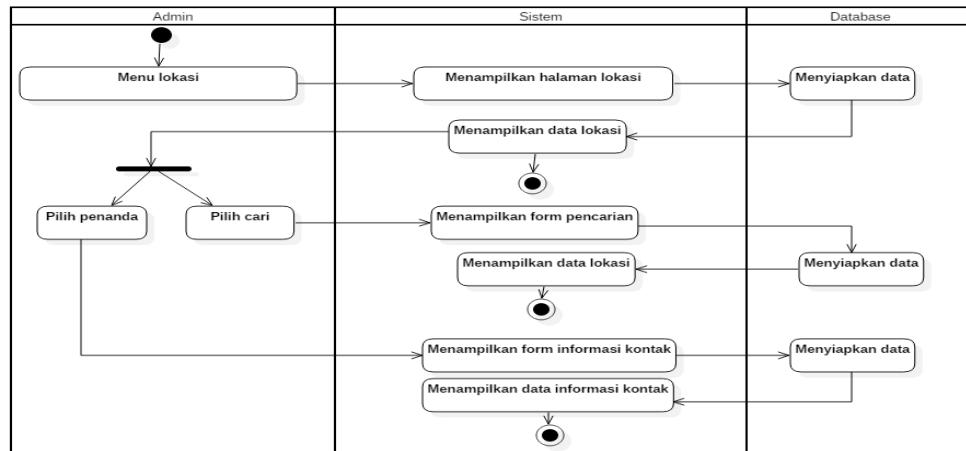
Aktifitas Informasi Pengumuman di mulai dengan admin masuk ke menu informasi dan memilih tab Loker. Aplikasi merespon dengan menampilkan halaman Loker, isi list Loker terbaru kemudian ketika dipilih akan ditampilkan ke form detail dan user dapat melakukan tambah, edit dan bisa menghapus semua.



Gambar 4.23 *Activity Diagram* Admin Pengumuman

10. Activity Diagram Admin Lokasi

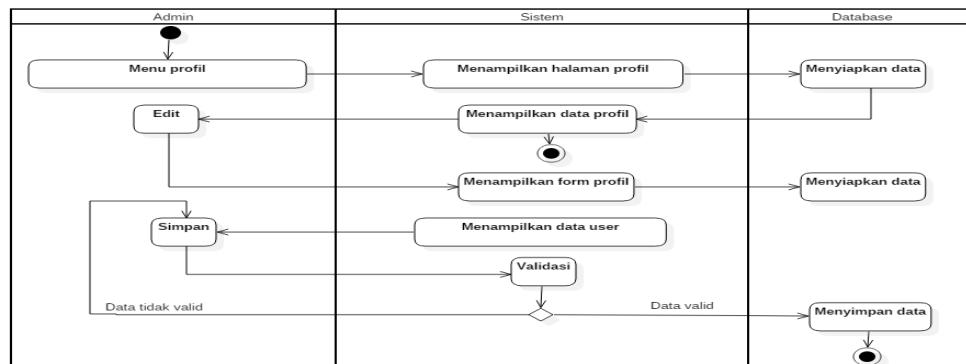
Aktifitas Lokasi di mulai dengan admin masuk ke menu lokasi terdapat menu pencarian data sesuai kriteria angkatan dan program studi. Aplikasi merespon dengan menampilkan halaman map lokasi keberadaan alumni jika alumni sudah perbarui data dan set lokasi.



Gambar 4.24 Activity Diagram Admin Lokasi

11. Activity Diagram Admin Profil

Aktifitas Profil di mulai dengan admin maupun user masuk ke menu profil aplikasi merespon dengan menampilkan halaman profil, dan dapat melakukan pembaharuan informasi data alumni dan kuesioner.

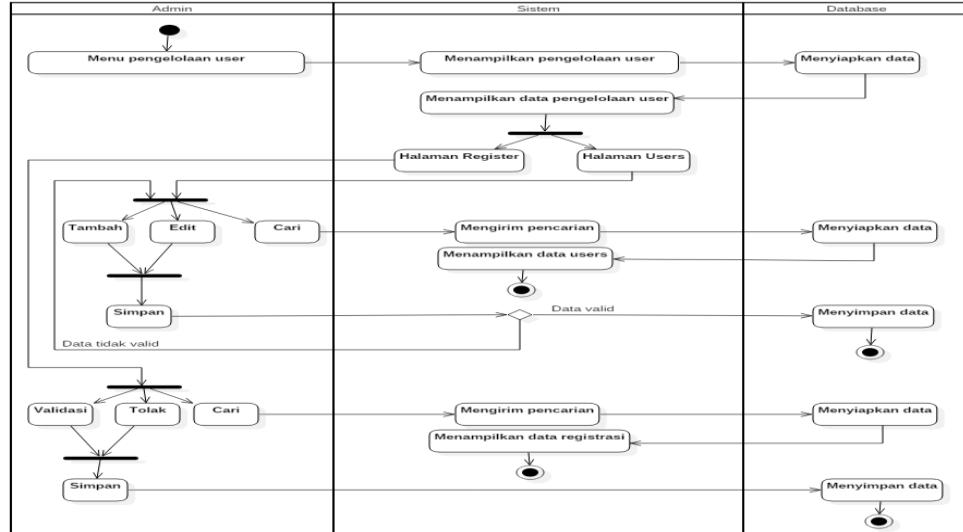


Gambar 4.25 Activity Diagram Admin Profil

12. Activity Diagram Admin Pengelolaan User

Aktifitas Pengelolaan User di mulai dengan Admin masuk ke menu Pengelolaan user terdapat sub menu registrasi dan users. Aplikasi merespon dengan menampilkan halaman yang dipilih. Pada pengelolaan user registrasi, dapat melakukan validasi user baru dengan memilih validasi atau tolak dan data

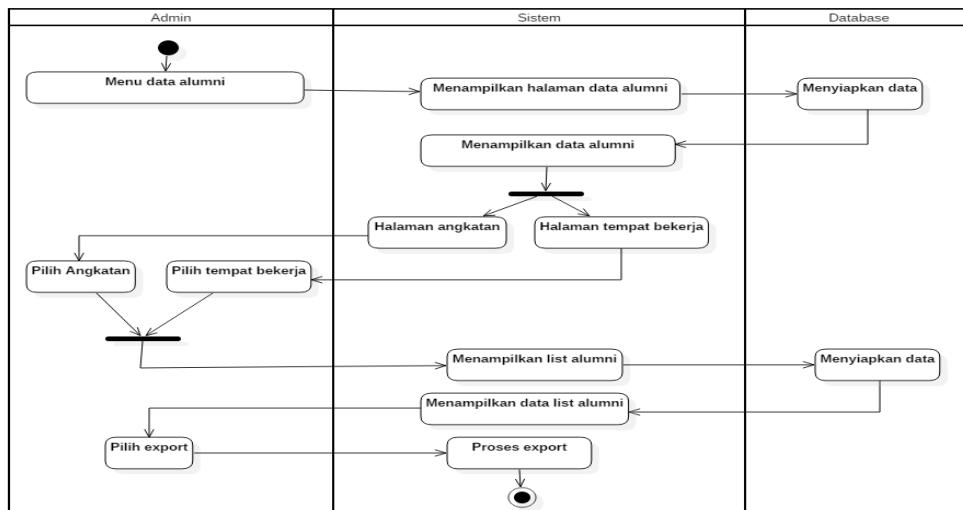
tersebut meliputi nim,nama,email dan *password*. Pada Pengelolaan user users, dapat melakukan penambahan dan perubahan data login tersebut meliputi nim, nama, email, *image*, dan akses.



Gambar 4.26 Activity Diagram Pengelolaan user

13. Activity Diagram Admin Data Alumni

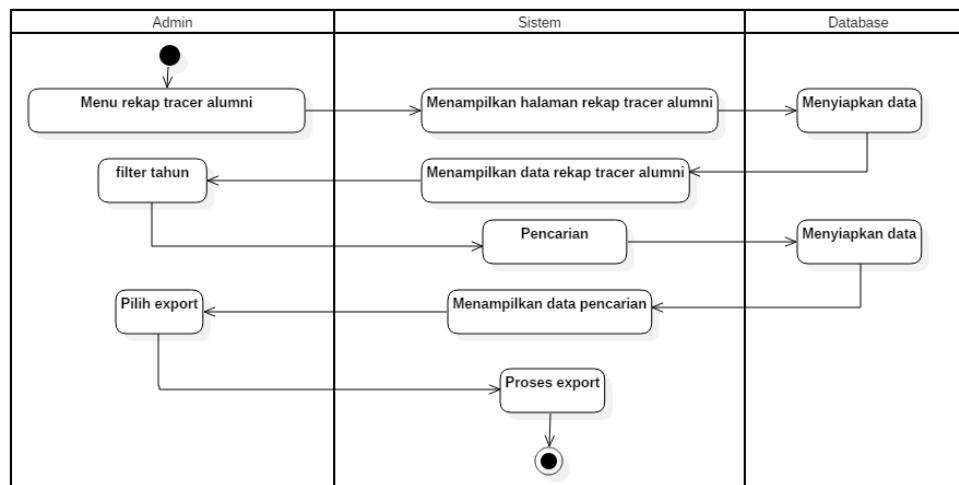
Aktifitas Admin Data Alumni di mulai dengan Admin masuk ke menu Data Alumni terdapat sub menu angkatan dan tempat bekerja. Aplikasi merespon dengan menampilkan halaman yang dipilih. Pada data alumni angkatan, dapat melihat angkatan dan memilih kemudian menampilkan alumni sesuai dengan angkatan tahun yang dipilih. Pada data alumni tempat bekerja, dapat melihat tempat bekerja dan memilih kemudian menampilkan alumni sesuai dengan angkatan tahun yang dipilih.



Gambar 4.27 Activity Diagram Data Alumni

14. Activity Diagram Admin Rekap Tracer Alumni

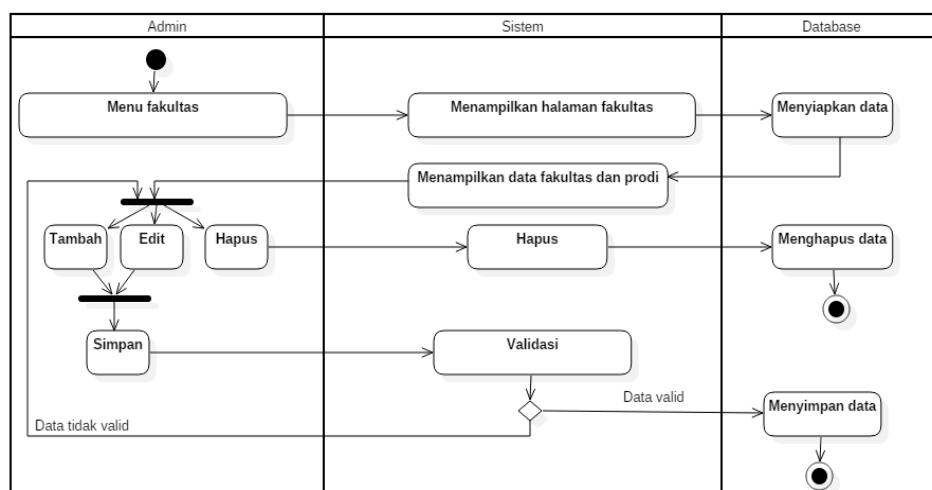
Aktifitas Admin Rekap *Tracer Alumni* di mulai dengan Admin masuk ke menu Rekap *Tracer Alumni*. Aplikasi merespon dengan menampilkan halaman yang dipilih. Pada Rekap *Tracer Alumni*, dapat melihat hasil kuesioner tracer study alumni dan memfilter berdasarkan tahun awal kemudian menampilkan data tracer alumni sesuai dengan angkatan tahun yang dipilih.



Gambar 4.28 Activity Diagram Rekap *Tracer Alumni*

15. Activity Diagram Admin Fakultas

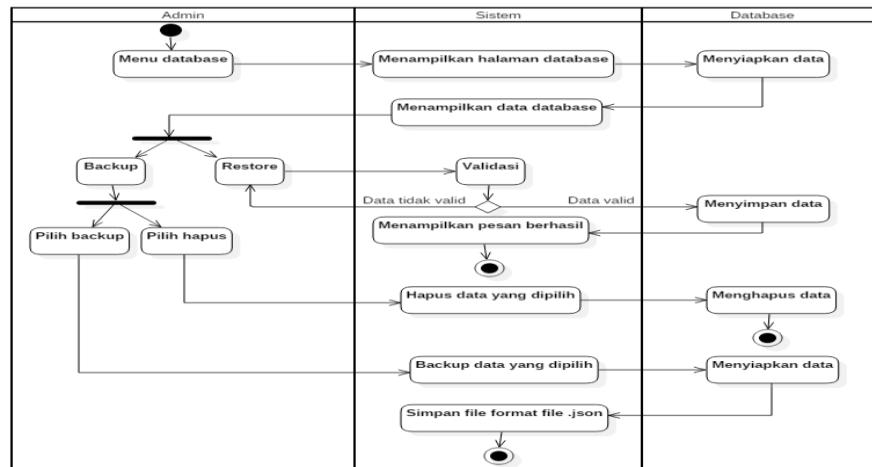
Aktifitas Admin Fakultas di mulai dengan Admin masuk ke menu Admin Fakultas. Aplikasi merespon dengan menampilkan halaman yang dipilih. Pada menu Fakultas, dapat melihat program studi sesuai dengan angkatan tahun yang dipilih dan dapat menambah, mengedit dan menghapus data fakultas maupun program studi.



Gambar 4.29 Activity Diagram Fakultas

16. Activity Diagram Admin Database

Aktifitas Admin *Database* di mulai dengan Admin masuk ke menu Admin *Database*. Aplikasi merespon dengan menampilkan halaman yang dipilih. Pada menu *Backup*, dapat melakukan *backup* data kemudian menyimpan kedalam format *file .json* dan menghapus data sesuai yang dipilih. Pada menu *Restore*, dapat melakukan restore data sesuai dengan yang dipilih.

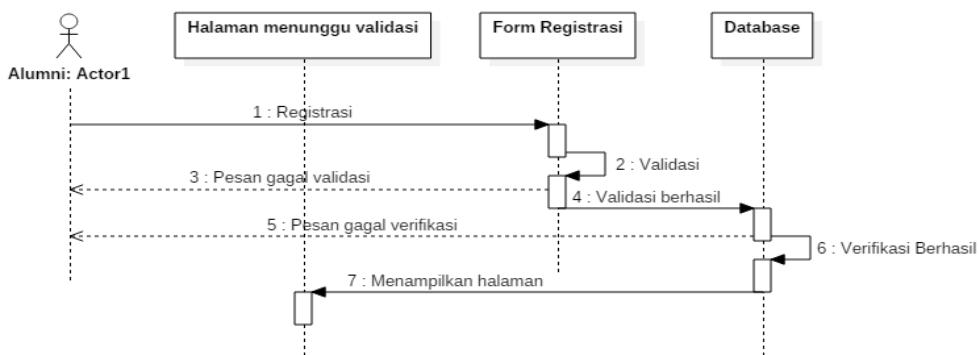


Gambar 4.30 Activity Diagram Database

4.2.3 Sequence Diagram

Sequence Diagram merupakan diagram yang menggambarkan interaksi antar objek pada sistem yang dibangun. *Sequence Diagram* dibagi menjadi 2 bagian yaitu admin dan alumni berikut adalah *Sequence Diagram* Alumni:

1. Sequence Diagram Alumni Registrasi



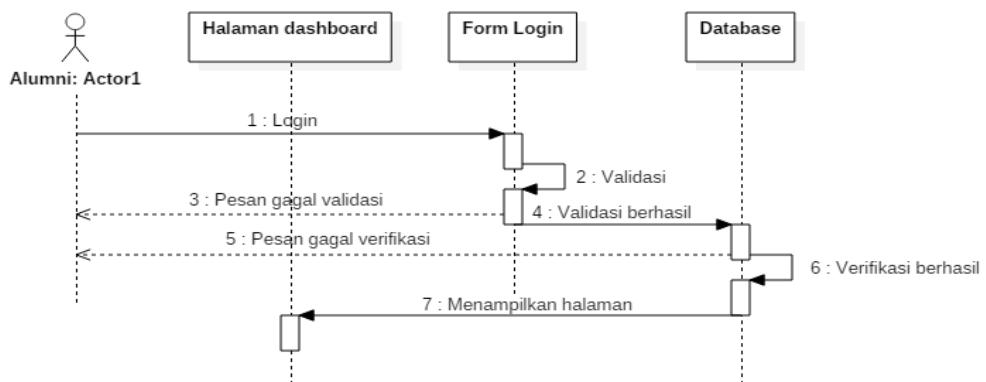
Gambar 4.31 Sequence Diagram Alumni Registrasi

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Registrasi:

1. Alumni melakukan registrasi

2. Aplikasi melakukan validasi terhadap inputan registrasi
3. Menampilkan pesan gagal validasi
4. Validasi berhasil mengirim data ke database
5. Registrasi tidak dapat dilakukan karena data sudah terdaftar
6. Verifikasi berhasil menyimpan data ke *database*
7. Menampilkan halaman tunggu validasi

2. Sequence Diagram Alumni Login

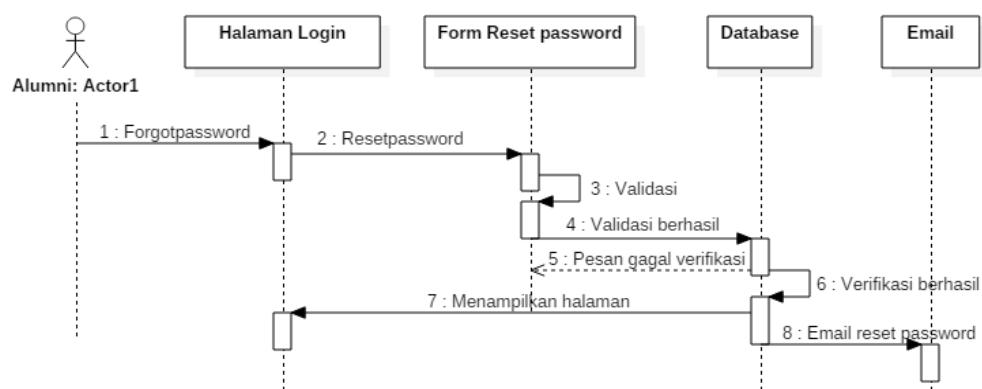


Gambar 4.32 Sequence Diagram Alumni Login

Berikut ini adalah penjelasan dari Sequence Diagram Alumni Login:

1. Alumni melakukan login
2. Aplikasi melakukan validasi terhadap inputan login
3. Menampilkan pesan gagal validasi
4. Validasi berhasil mengirim data ke *database*
5. Menampilkan pesan gagal verifikasi atau data tidak cocok
6. Verifikasi berhasil atau pencocokan data valid
7. Menampilkan halaman *dashboard*.

3. Sequence Diagram Alumni Reset Password

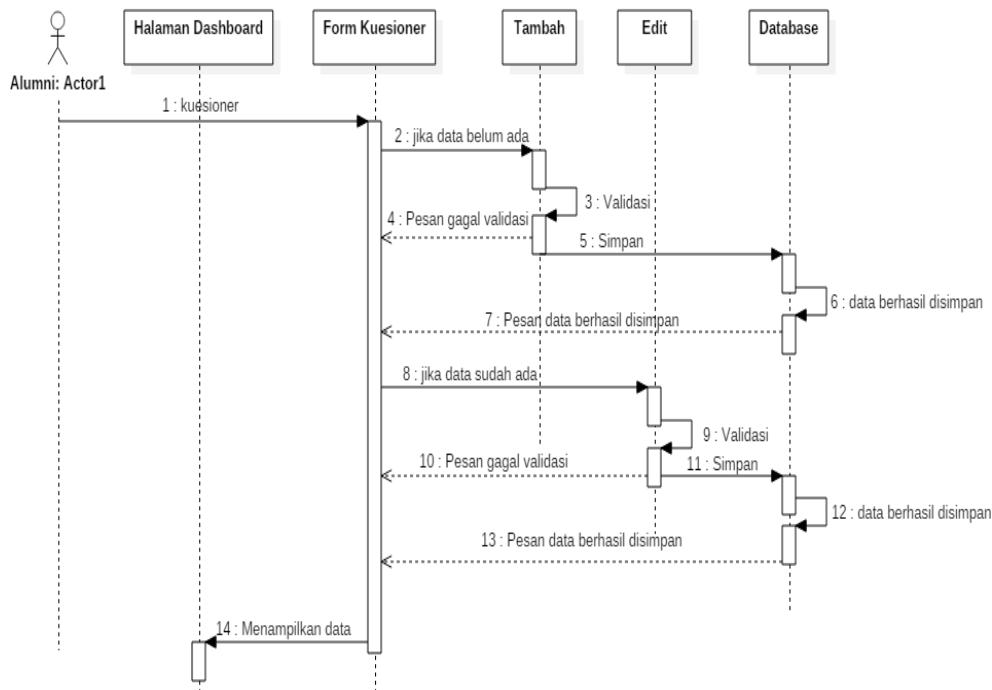


Gambar 4.33 Sequence Diagram Alumni Reset Password

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni *Reset Password*:

1. Alumni memilih *forgot password* dihalaman login
2. Alumni melakukan inputan *reset password* dengan mengisi *email* yang sudah terdaftar
3. Aplikasi melakukan validasi terhadap inputan *reset password*
4. Validasi berhasil mengirim data ke database
5. Menampilkan pesan gagal verifikasi atau data tidak cocok
6. Verifikasi berhasil atau pencocokan data valid
7. Menampilkan halaman login.
8. Aplikasi mengirim perintah *reset password* ke *email*

4. *Sequence Diagram* Alumni Kuesioner



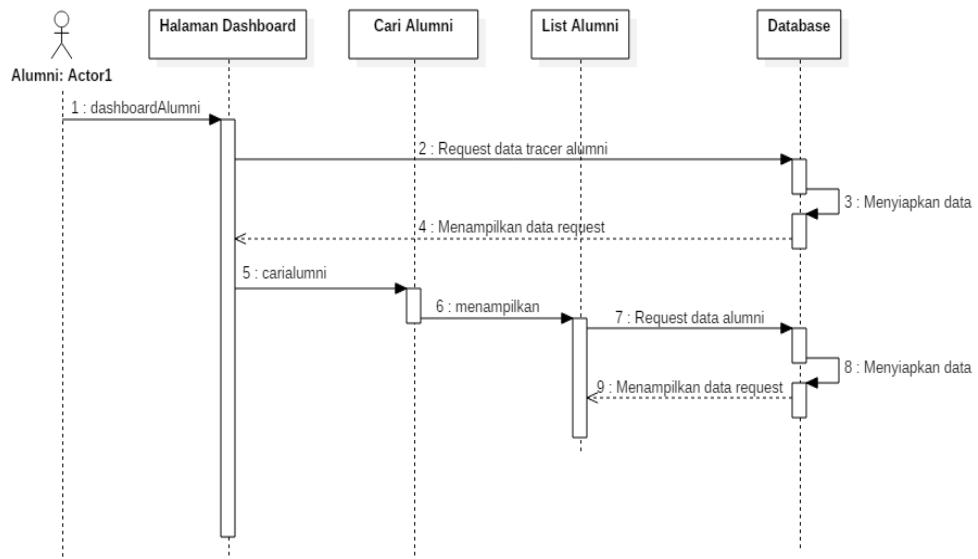
Gambar 4.34 *Sequence Diagram* Alumni Kuesioner

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Kuesioner:

1. Alumni memilih menu kuesioner, dan melakukan pengisian kuesioner
2. Jika data belum ada maka akan melakukan perintah tambah
3. Aplikasi melakukan validasi terhadap inputan
4. Menampilkan pesan gagal validasi

5. Validasi berhasil menyimpan data ke *database*
6. Data berhasil disimpan
7. Menampilkan pesan berhasil disimpan
8. Jika data sudah ada maka akan melakukan perintah edit
9. Aplikasi melakukan validasi terhadap inputan
10. Menampilkan pesan gagal validasi
11. Validasi berhasil menyimpan data ke *database*
12. Data berhasil disimpan
13. Menampilkan pesan berhasil disimpan
14. Menampilkan data dan halaman *dashboard*

5. Sequence Diagram Alumni Dashboard

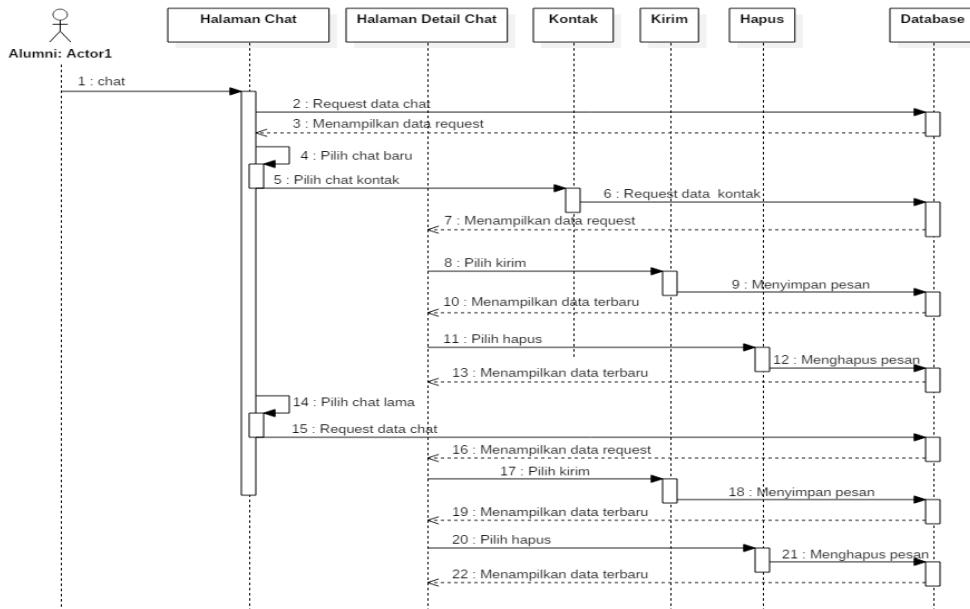


Gambar 4.35 Sequence Diagram Alumni Dashboard

Berikut ini adalah penjelasan dari *Sequence Diagram Dashboard*:

1. Alumni memilih menu *dashboard*
2. Aplikasi melakukan permintaan data ke *database*
3. Database menyiapkan data
4. Menampilkan data dari database ke halaman *dashboard*
5. Pencarian alumni
6. Menampilkan list alumni
7. Aplikasi melakukan permintaan data ke *database*
8. Database menyiapkan data
9. Menampilkan data dari database ke halaman list alumni

6. Sequence Diagram Alumni Chat



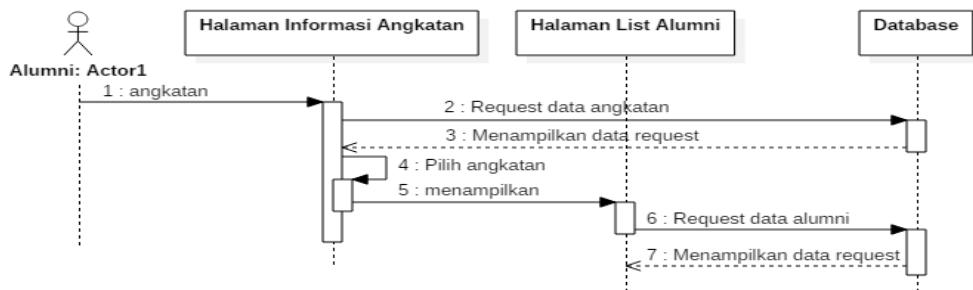
Gambar 4.36 Sequence Diagram Alumni Chat

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Chat:

1. Alumni memilih menu chat
2. Aplikasi melakukan permintaan data chat
3. Aplikasi menampilkan data permintaan chat
4. Alumni memilih chat baru
5. Alumni memilih kontak chatting
6. Aplikasi melakukan permintaan data kontak chatting
7. Aplikasi menampilkan data permintaan data kontak chatting
8. Alumni memilih menu kirim pesan
9. Aplikasi mengirim perintah menyimpan pesan ke database
10. Aplikasi menampilkan data terbaru
11. Alumni memilih menu hapus pesan
12. Aplikasi mengirim perintah hapus pesan kedatabase
13. Aplikasi menampilkan data terbaru
14. Memilih chat lama
15. Aplikasi melakukan permintaan data chat
16. Aplikasi menampilkan data permintaan chat
17. Alumni memilih kirim
18. Aplikasi mengirim perintah menyimpan pesan ke database

19. Aplikasi menampilkan data terbaru
20. Alumni memilih hapus
21. Aplikasi mengirim perintah menghapus pesan ke database
22. Aplikasi menampilkan data terbaru

7. Sequence Diagram Alumni Informasi Angkatan

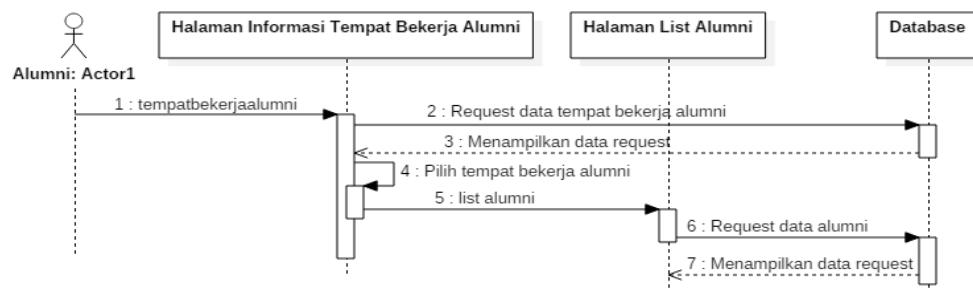


Gambar 4.37 Sequence Diagram Alumni Informasi Angkatan

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Informasi Angkatan:

1. Alumni memilih menu informasi angkatan
2. Aplikasi melakukan permintaan data angkatan
3. Aplikasi menampilkan data angkatan
4. Alumni memilih angkatan
5. Menampilkan halaman list alumni
6. Aplikasi melakukan permintaan data alumni sesuai dengan angkatan yang dipilih
7. Menampilkan data alumni sesuai angkatan yang dipilih

8. Sequence Diagram Alumni Informasi Tempat Bekerja Alumni



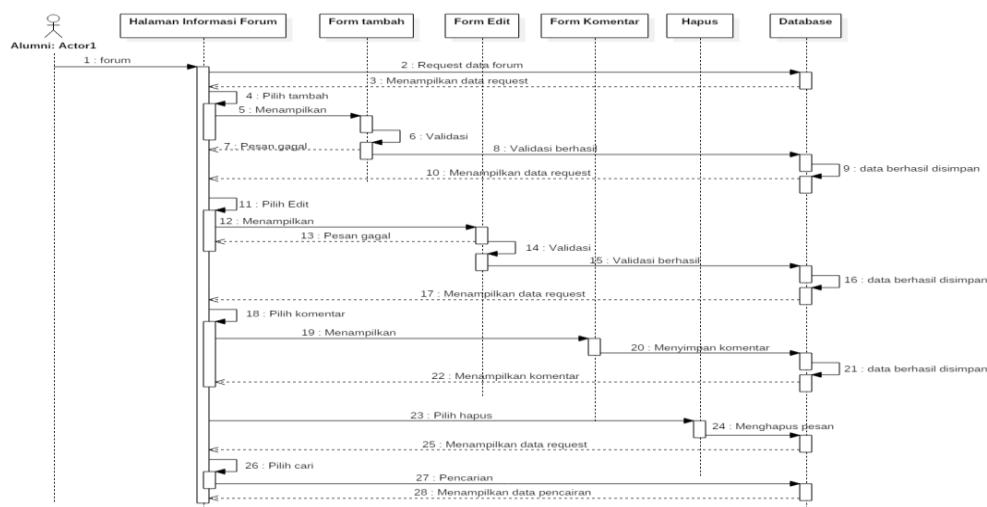
Gambar 4.38 Sequence Diagram Alumni Informasi Tempat Bekerja

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Informasi Tempat Bekerja Alumni:

1. Alumni memilih menu informasi tempat bekerja alumni

2. Aplikasi melakukan permintaan data tempat bekerja alumni
3. Aplikasi menampilkan data tempat bekerja alumni
4. Alumni memilih tempat bekerja alumni
5. Menampilkan halaman list alumni
6. Aplikasi melakukan permintaan data alumni sesuai dengan tempat bekerja alumni yang dipilih
7. Menampilkan data alumni sesuai tempat bekerja alumni yang dipilih

9. Sequence Diagram Alumni Informasi Forum



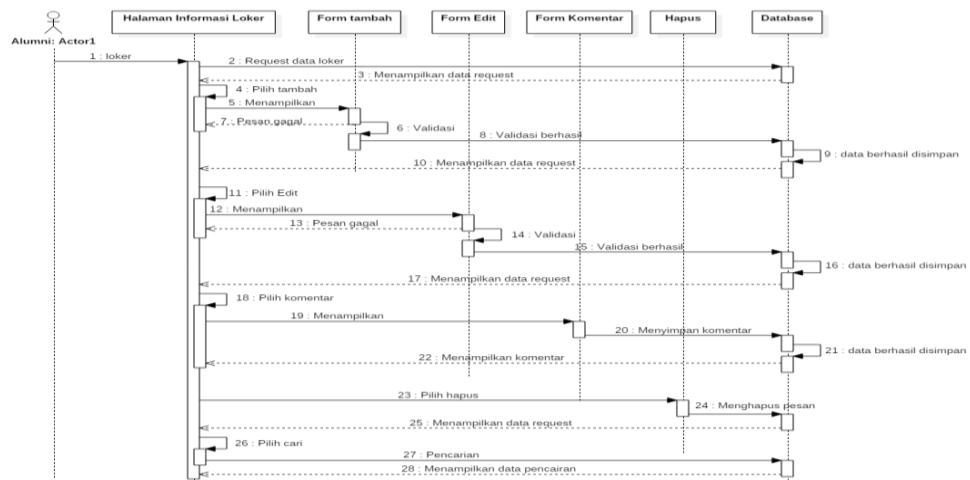
Gambar 4.39 Sequence Diagram Alumni Informasi Forum

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Informasi Forum:

1. Alumni memilih menu informasi forum
2. Aplikasi melakukan permintaan data forum
3. Aplikasi menampilkan data forum
4. Alumni memilih tambah forum
5. Menampilkan halaman forum tambah
6. Validasi terhadap inputan forum
7. Menampilkan pesan gagal
8. Validasi berhasil menyimpan data ke database
9. Data berhasil disimpan
10. Aplikasi menampilkan data forum terbaru
11. Alumni memilih edit
12. Menampilkan halaman forum edit

13. Validasi terhadap inputan forum
14. Menampilkan pesan gagal validasi
15. Validasi berhasil menyimpan data ke database
16. Data berhasil disimpan
17. Aplikasi menampilkan data forum terbaru
18. Alumni memilih komentar
19. Menampilkan halaman komentar
20. Menyimpan komentar ke database
21. Data berhasil disimpan
22. Menampilkan data komentar
23. Alumni memilih hapus komentar
24. Aplikasi mengirim perintah menghapus pesan
25. Menampilkan data komentar
26. Alumni memilih cari
27. Alumni melakukan pencarian
28. Menampilkan data pencarian

10. Sequence Diagram Alumni Informasi Loker



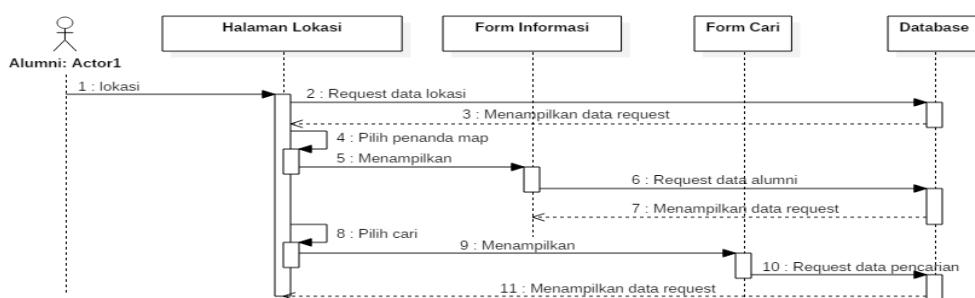
Gambar 4.40 Sequence Diagram Alumni Informasi Loker

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Informasi Loker:

1. Alumni memilih menu informasi loker
2. Aplikasi melakukan permintaan data loker
3. Aplikasi menampilkan data loker

4. Alumni memilih tambah loker
5. Menampilkan halaman loker tambah
6. Validasi terhadap inputan loker
7. Menampilkan pesan gagal
8. Validasi berhasil menyimpan data ke database
9. Data berhasil disimpan
10. Aplikasi menampilkan data loker terbaru
11. Alumni memilih edit
12. Menampilkan halaman loker edit
13. Validasi terhadap inputan loker
14. Menampilkan pesan gagal validasi
15. Validasi berhasil menyimpan data ke database
16. Data berhasil disimpan
17. Aplikasi menampilkan data loker terbaru
18. Alumni memilih komentar
19. Menampilkan halaman komentar
20. Menyimpan komentar ke database
21. Data berhasil disimpan
22. Menampilkan data komentar
23. Alumni memilih hapus komentar
24. Aplikasi mengirim perintah menghapus pesan
25. Menampilkan data komentar
26. Alumni memilih cari
27. Alumni melakukan pencarian
28. Menampilkan data pencarian

11. Sequence Diagram Alumni Lokasi

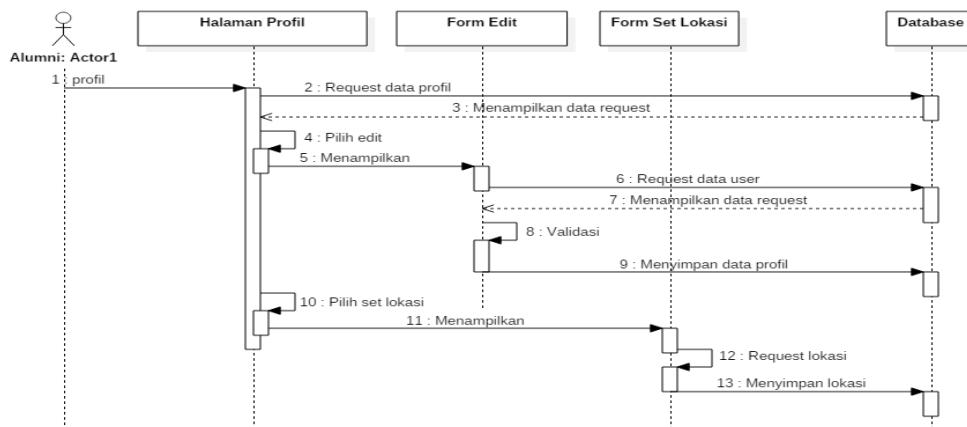


Gambar 4.41 Sequence Diagram Alumni Lokasi

Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Lokasi:

1. Alumni memilih menu lokasi
2. Aplikasi melakukan permintaan data lokasi ke database
3. Menampilkan data lokasi alumni
4. Alumni memilih penanda map
5. Menampilkan halaman informasi
6. Aplikasi melakukan permintaan data alumni sesuai penanda map yang dipilih
7. Menampilkan data alumni yang dipilih pada halaman informasi
8. Alumni memilih cari
9. Menampilkan halaman pencarian
10. Aplikasi melakukan pencarian data lokasi ke database
11. Menampilkan data alumni sesuai dengan pencarian

12. Sequence Diagram Alumni Profil



Gambar 4.42 Sequence Diagram Alumni Profil

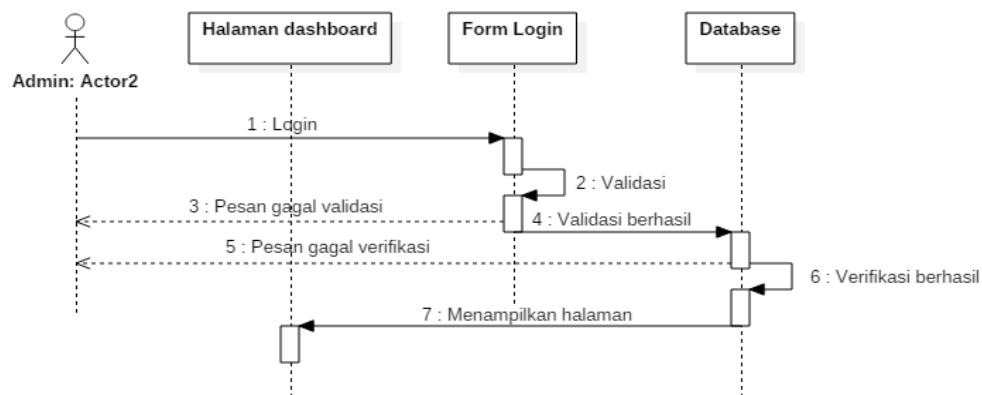
Berikut ini adalah penjelasan dari *Sequence Diagram* Alumni Profil:

1. Alumni memilih menu profil dan menampilkan halaman profil
2. Aplikasi melakukan permintaan data profil
3. Menampilkan data profil
4. Alumni memilih edit
5. Menampilkan halaman edit
6. Aplikasi melakukan permintaan data profil
7. Menampilkan data profil
8. Validasi terhadap inputan edit profil

9. Aplikasi melakukan perintah menyimpan data profil
10. Alumni memilih set lokasi
11. Menampilkan halaman set lokasi
12. Aplikasi melakukan permintaan data lokasi
13. Aplikasi melakukan perintah menyimpan data lokasi

Sedangkan untuk *Sequence Diagram* Admin adalah sebagai berikut:

1. Sequence Diagram Admin Login

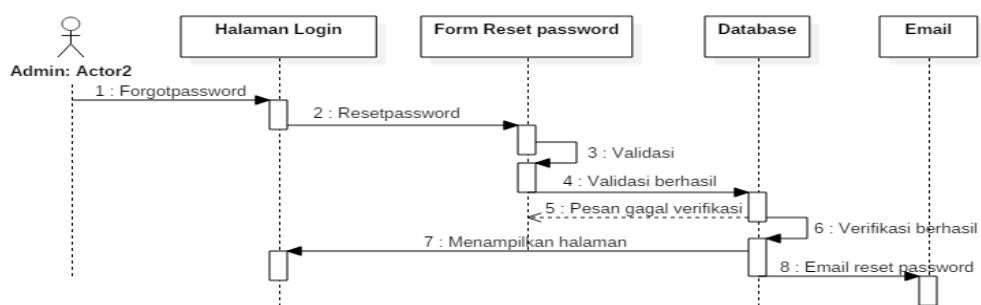


Gambar 4.43 Sequence Diagram Admin Login

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Login:

1. Admin melakukan login
2. Aplikasi melakukan validasi terhadap inputan login
3. Menampilkan pesan gagal validasi
4. Validasi berhasil mengirim data ke database
5. Menampilkan pesan gagal verifikasi atau data tidak cocok
6. Verifikasi berhasil atau pencocokan data valid
7. Menampilkan halaman dashboard.

2. Sequence Diagram Admin Reset Password

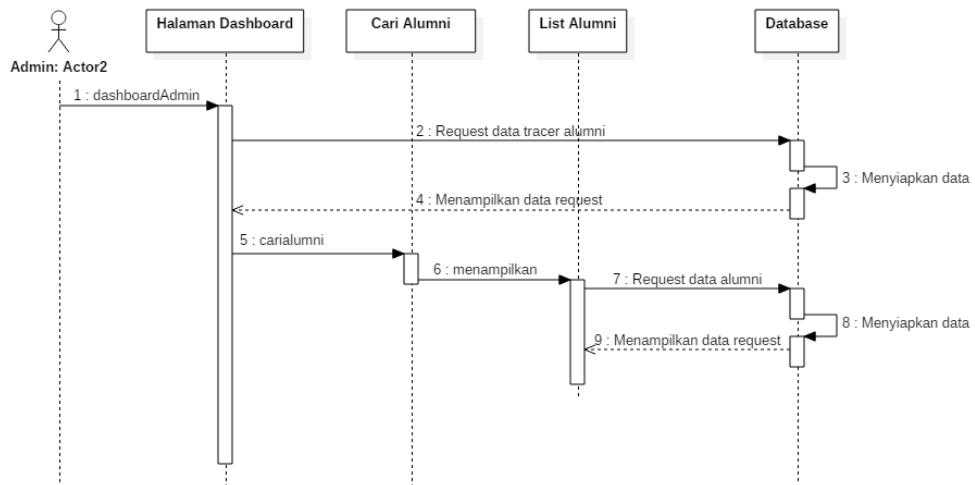


Gambar 4.44 Sequence Diagram Admin Reset Password

Berikut ini adalah penjelasan dari *Sequence Diagram Admin Reset Password*:

1. Admin memilih *forgot password* dihalaman login
2. Admin melakukan inputan *reset password* dengan mengisi *email* yang sudah terdaftar
3. Aplikasi melakukan validasi terhadap inputan *reset password*
4. Validasi berhasil mengirim data ke *database*
5. Menampilkan pesan gagal verifikasi atau data tidak cocok
6. Verifikasi berhasil atau pencocokan data valid
7. Menampilkan halaman login.
8. Aplikasi mengirim perintah *reset password* ke *email*

3. Sequence Diagram Admin Dashboard

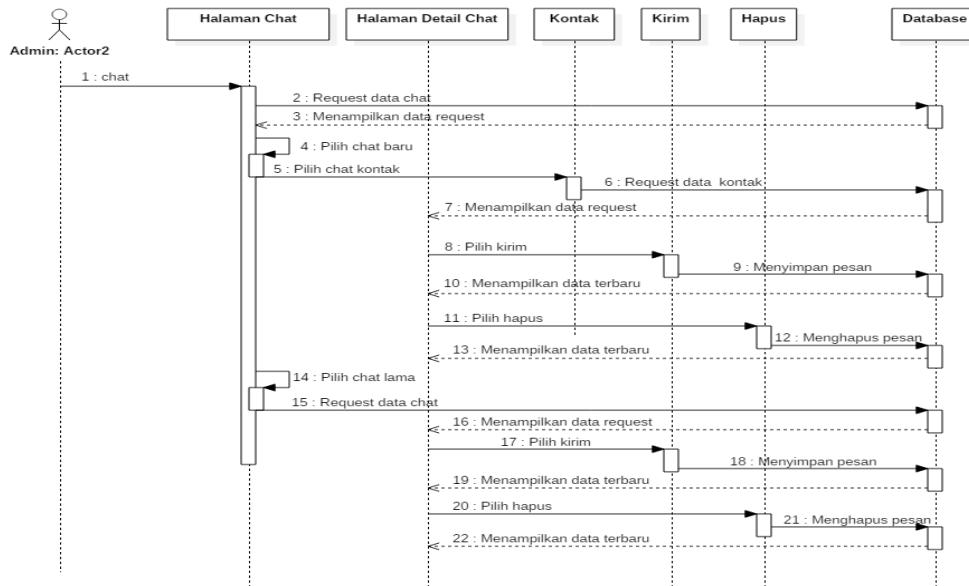


Gambar 4.45 Sequence Diagram Admin Dashboard

Berikut ini adalah penjelasan dari *Sequence Diagram Admin Dashboard*:

1. Admin memilih menu *dashboard*
2. Aplikasi melakukan permintaan data ke *database*
3. Database menyiapkan data
4. Menampilkan data dari database ke halaman *dashboard*
5. Pencarian alumni
6. Menampilkan list alumni
7. Aplikasi melakukan permintaan data ke *database*
8. Database menyiapkan data
9. Menampilkan data dari database ke halaman list alumni

4. Sequence Diagram Admin Chat



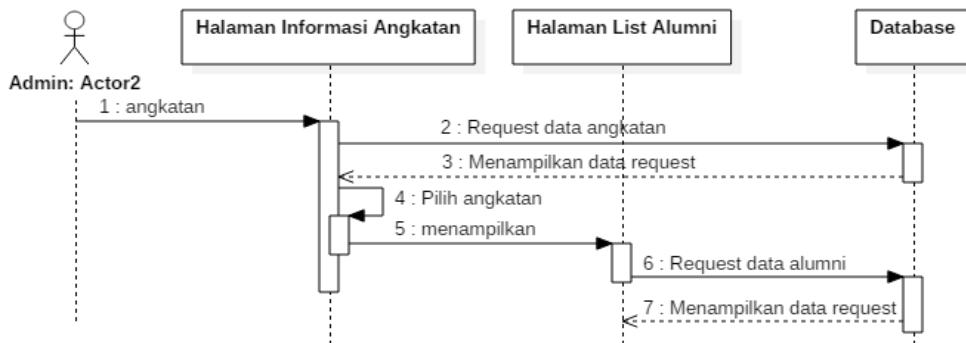
Gambar 4.46 Sequence Diagram Admin Chat

Berikut ini adalah penjelasan dari *Sequence Diagram Admin Chat*:

1. Admin memilih menu chat
2. Aplikasi melakukan permintaan data chat
3. Aplikasi menampilkan data permintaan chat
4. Admin memilih chat baru
5. Admin memilih kontak chatting
6. Aplikasi melakukan permintaan data kontak chatting
7. Aplikasi menampilkan data permintaan data kontak chatting
8. Admin memilih menu kirim pesan
9. Aplikasi mengirim perintah menyimpan pesan ke *database*
10. Aplikasi menampilkan data terbaru
11. Admin memilih menu hapus pesan
12. Aplikasi mengirim perintah hapus pesan ke *database*
13. Aplikasi menampilkan data terbaru
14. Memilih chat lama
15. Aplikasi melakukan permintaan data chat
16. Aplikasi menampilkan data permintaan chat
17. Admin memilih kirim
18. Aplikasi mengirim perintah menyimpan pesan ke *database*

19. Aplikasi menampilkan data terbaru
20. Alumni memilih hapus
21. Aplikasi mengirim perintah menghapus pesan ke database
22. Aplikasi menampilkan data terbaru

5. Sequence Diagram Admin Informasi Angkatan

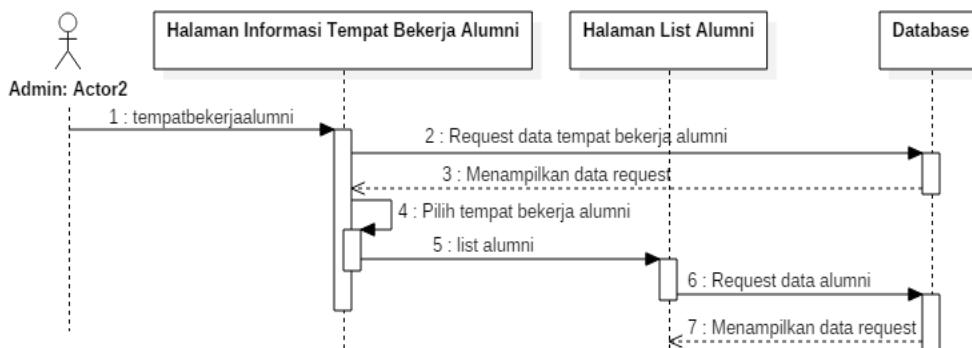


Gambar 4.47 Sequence Diagram Admin Informasi Angkatan

Berikut ini adalah penjelasan dari *Sequence Diagram Admin Informasi Angkatan*:

1. Admin memilih menu informasi angkatan
2. Aplikasi melakukan permintaan data angkatan
3. Aplikasi menampilkan data angkatan
4. Admin memilih angkatan
5. Menampilkan halaman list alumni
6. Aplikasi melakukan permintaan data alumni sesuai dengan angkatan yang dipilih
7. Menampilkan data alumni sesuai angkatan yang dipilih

6. Sequence Diagram Admin Informasi Tempat Bekerja Alumni

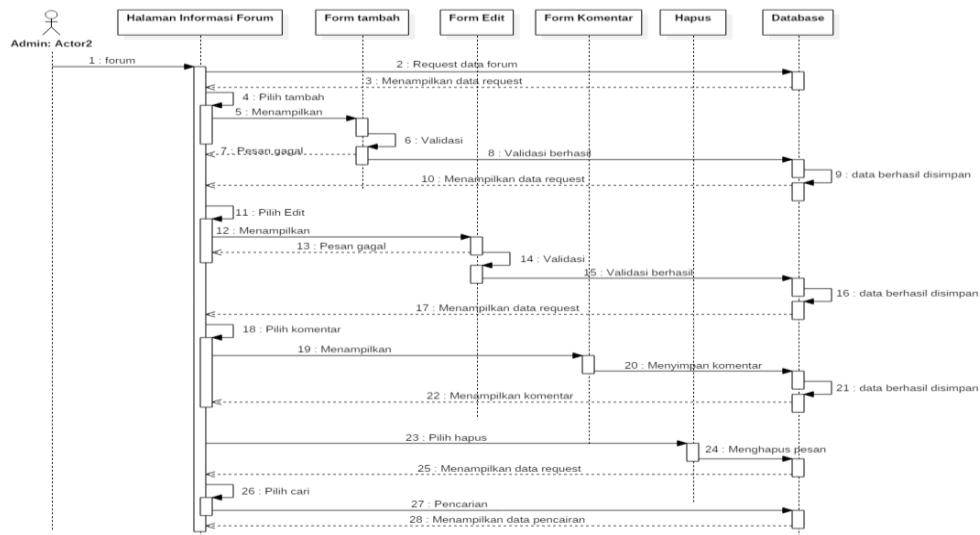


Gambar 4.48 Sequence Diagram Admin Tempat Bekerja Alumni

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Informasi Tempat Bekerja Alumni:

1. Admin memilih menu informasi tempat bekerja alumni
2. Aplikasi melakukan permintaan data tempat bekerja alumni
3. Aplikasi menampilkan data tempat bekerja alumni
4. Admin memilih tempat bekerja alumni
5. Menampilkan halaman list alumni
6. Aplikasi melakukan permintaan data alumni sesuai dengan tempat bekerja alumni yang dipilih
7. Menampilkan data alumni sesuai tempat bekerja alumni yang dipilih

7. Sequence Diagram Admin Informasi Forum



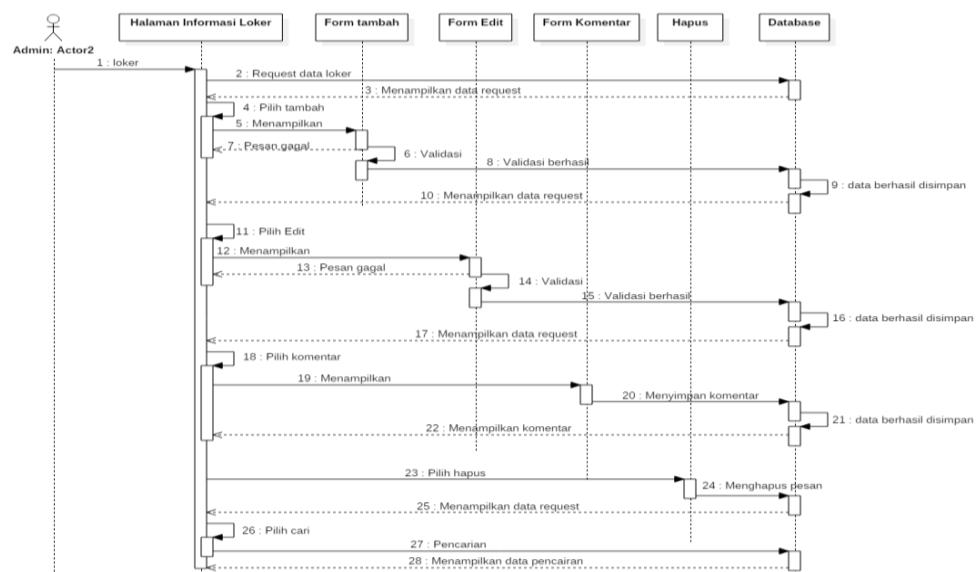
Gambar 4.49 Sequence Diagram Admin Informasi Forum

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Informasi Forum:

1. Admin memilih menu informasi forum
2. Aplikasi melakukan permintaan data forum
3. Aplikasi menampilkan data forum
4. Admin memilih tambah forum
5. Menampilkan halaman forum tambah
6. Validasi terhadap inputan forum
7. Menampilkan pesan gagal
8. Validasi berhasil menyimpan data ke *database*

9. Data berhasil disimpan
10. Aplikasi menampilkan data forum terbaru
11. Admin memilih edit
12. Menampilkan halaman forum edit
13. Validasi terhadap inputan forum
14. Menampilkan pesan gagal validasi
15. Validasi berhasil menyimpan data ke database
16. Data berhasil disimpan
17. Aplikasi menampilkan data forum terbaru
18. Admin memilih komentar
19. Menampilkan halaman komentar
20. Menyimpan komentar ke *database*
21. Data berhasil disimpan
22. Menampilkan data komentar
23. Admin memilih hapus komentar
24. Aplikasi mengirim perintah menghapus pesan
25. Menampilkan data komentar
26. Admin memilih cari
27. Admin melakukan pencarian
28. Menampilkan data pencarian

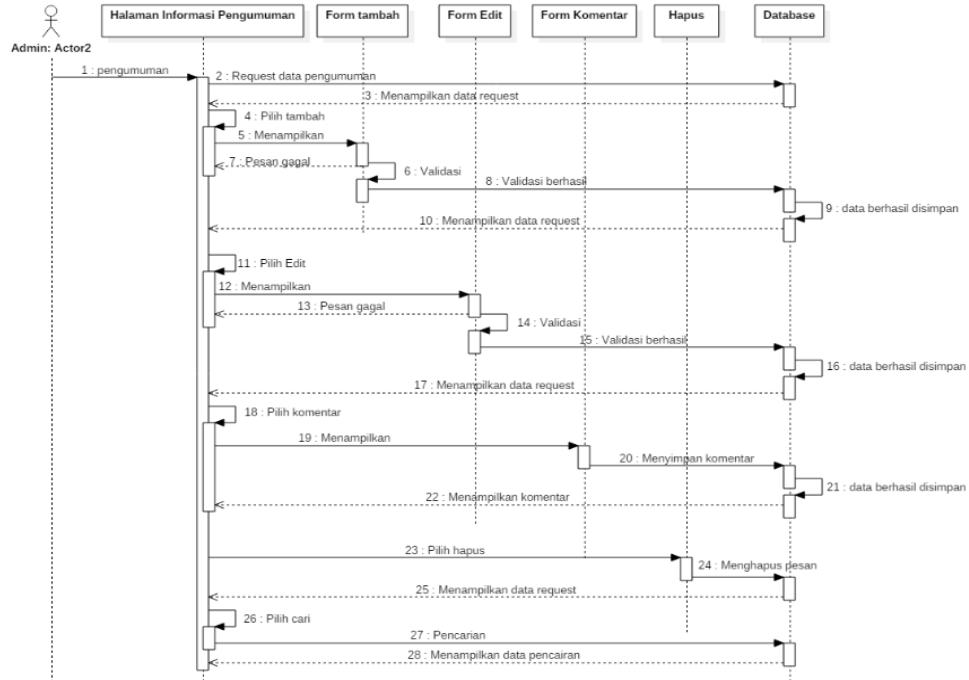
8. Sequence Diagram Admin Informasi Loker



Gambar 4.50 Sequence Diagram Admin Informasi Loker

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Informasi Loker:

1. Admin memilih menu informasi loker
 2. Aplikasi melakukan permintaan data loker
 3. Aplikasi menampilkan data loker
 4. Admin memilih tambah loker
 5. Menampilkan halaman loker tambah
 6. Validasi terhadap inputan loker
 7. Menampilkan pesan gagal
 8. Validasi berhasil menyimpan data ke database
 9. Data berhasil disimpan
 10. Aplikasi menampilkan data loker terbaru
 11. Admin memilih edit
 12. Menampilkan halaman loker edit
 13. Validasi terhadap inputan loker
 14. Menampilkan pesan gagal validasi
 15. Validasi berhasil menyimpan data ke *database*
 16. Data berhasil disimpan
 17. Aplikasi menampilkan data loker terbaru
 18. Admin memilih komentar
 19. Menampilkan halaman komentar
 20. Menyimpan komentar ke database
 21. Data berhasil disimpan
 22. Menampilkan data komentar
 23. Admin memilih hapus komentar
 24. Aplikasi mengirim perintah menghapus pesan
 25. Menampilkan data komentar
 26. Admin memilih cari
 27. Admin melakukan pencarian
 28. Menampilkan data pencarian
9. *Sequence Diagram* Admin Informasi Pengumuman



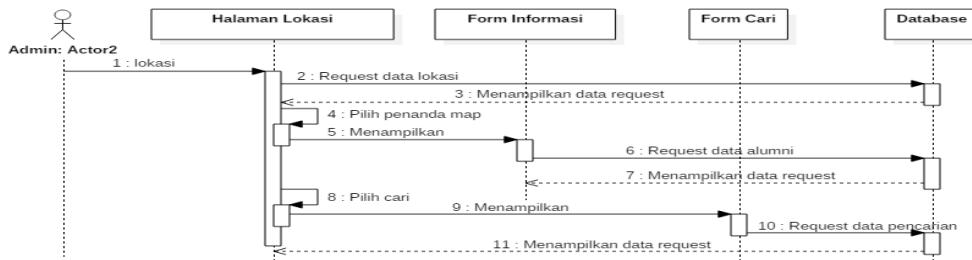
Gambar 4.51 *Sequence Diagram* Admin Informasi Pengumuman

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Informasi Pengumuman:

1. Admin memilih menu informasi pengumuman
2. Aplikasi melakukan permintaan data pengumuman
3. Aplikasi menampilkan data pengumuman
4. Admin memilih tambah pengumuman
5. Menampilkan halaman pengumuman tambah
6. Validasi terhadap inputan pengumuman
7. Menampilkan pesan gagal
8. Validasi berhasil menyimpan data ke database
9. Data berhasil disimpan
10. Aplikasi menampilkan data pengumuman terbaru
11. Admin memilih edit
12. Menampilkan halaman loker edit
13. Validasi terhadap inputan pengumuman
14. Menampilkan pesan gagal validasi
15. Validasi berhasil menyimpan data ke database
16. Data berhasil disimpan

17. Aplikasi menampilkan data pengumuman terbaru
18. Admin memilih komentar
19. Menampilkan halaman komentar
20. Menyimpan komentar ke database
21. Data berhasil disimpan
22. Menampilkan data komentar
23. Admin memilih hapus komentar
24. Aplikasi mengirim perintah menghapus pesan
25. Menampilkan data komentar
26. Admin memilih cari
27. Admin melakukan pencarian
28. Menampilkan data pencarian

10. Sequence Diagram Admin Lokasi

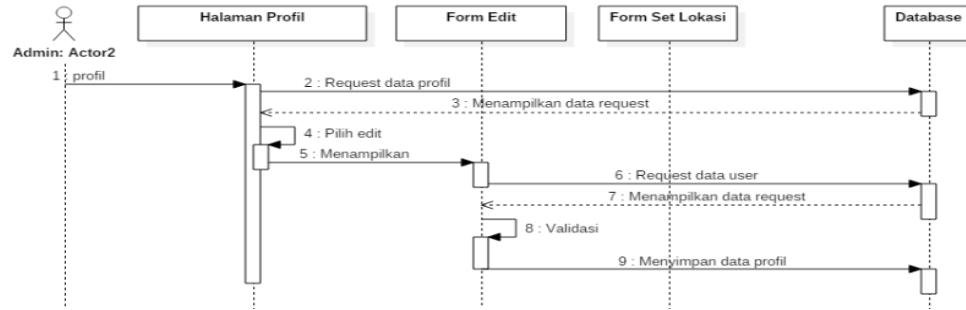


Gambar 4.52 Sequence Diagram Admin Lokasi

Berikut ini adalah penjelasan dari *Sequence Diagram Admin Lokasi*:

1. Admin memilih menu Lokasi
2. Aplikasi melakukan permintaan data lokasi ke database
3. Menampilkan data lokasi alumni
4. Admin memilih penanda map
5. Menampilkan halaman informasi
6. Aplikasi melakukan permintaan data alumni sesuai penanda map yang dipilih
7. Menampilkan data alumni yang dipilih pada halaman informasi
8. Admin memilih cari
9. Menampilkan halaman pencarian
10. Aplikasi melakukan pencarian data lokasi ke database
11. Menampilkan data alumni sesuai dengan pencarian

11. Sequence Diagram Admin Profil

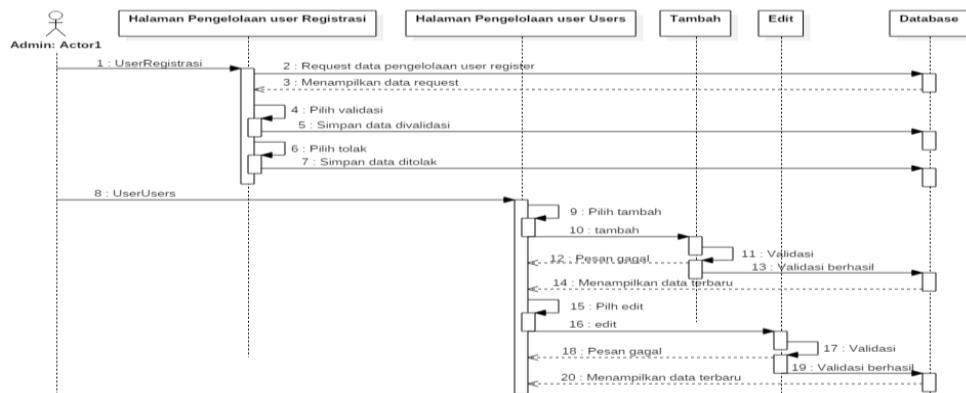


Gambar 4.53 Sequence Diagram Admin Profil

Berikut ini adalah penjelasan dari *Sequence Diagram Admin Profil*:

1. Admin memilih menu profil dan menampilkan halaman profil
2. Aplikasi melakukan permintaan data profil
3. Menampilkan data profil
4. Admin memilih edit
5. Menampilkan halaman edit
6. Aplikasi melakukan permintaan data profil
7. Menampilkan data profil
8. Validasi terhadap inputan edit profil
9. Aplikasi melakukan perintah menyimpan data profil

12. Sequence Diagram Admin Pengelolaan User



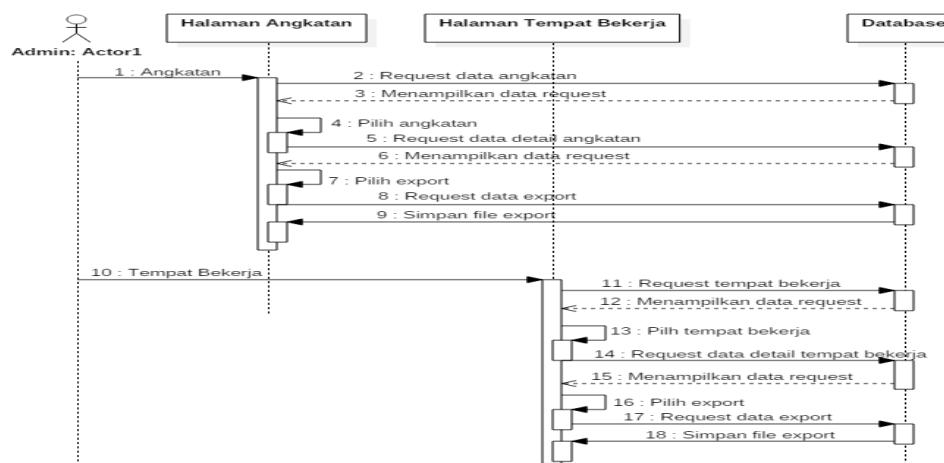
Gambar 4.54 Sequence Diagram Admin Pengelolaan User

Berikut ini adalah penjelasan dari *Sequence Diagram Admin Pengelolaan User*:

1. Admin memilih menu Pengelolaan User Tab Registrasi
2. Aplikasi melakukan *request* permintaan data user Registrasi
3. Database menyiapkan data user registrasi dan merespon mengirim data ke aplikasi

4. Aplikasi menyediakan fitur untuk mengolah data meliputi penambahan, perubahan, penghapusan dan pencarian
5. Admin dapat melakukan perubahan data
6. Aplikasi menampilkan data user Registrasi
7. Aplikasi akan melakukan validasi jika ada perubahan data
8. Aplikasi mengirim data tersebut ke database
9. Database melakukan verifikasi data
10. Database merespon mengirim data ke aplikasi dan aplikasi menampilkannya
11. Admin memilih Sub Menu Tab Login
12. Aplikasi melakukan *request* permintaan data user Login
13. Database menyiapkan data user login dan merespon mengirim data ke aplikasi
14. Aplikasi menyediakan fitur untuk mengolah data meliputi penambahan, perubahan, penghapusan dan pencarian
15. Admin dapat melakukan perubahan data
16. Aplikasi menampilkan data user Login
17. Aplikasi akan melakukan validasi jika ada perubahan data
18. Aplikasi mengirim data tersebut ke database
19. Database melakukan verifikasi data
20. Database merespon mengirim data ke aplikasi dan aplikasi menampilkannya

13. Sequence Diagram Admin Data Alumni

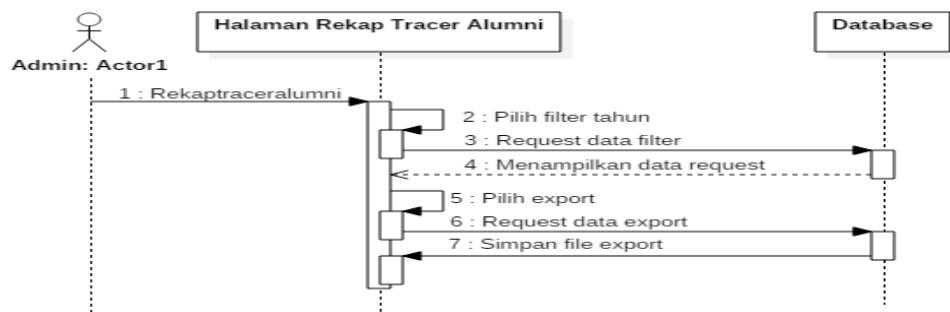


Gambar 4.55 Sequence Diagram Admin Data Alumni

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Data Alumni:

1. Admin memilih menu data alumni angkatan
2. Aplikasi melakukan permintaan data angkatan
3. Menampilkan data angkatan
4. Admin memilih angkatan
5. Aplikasi melakukan permintaan data detail angkatan
6. Menampilkan data alumni sesuai angkatan yang dipilih
7. Admin memilih export
8. Aplikasi melakukan permintaan data export
9. Menyimpan file export
10. Admin memilih tempat bekerja
11. Aplikasi melakukan permintaan data tempat bekerja
12. Menampilkan data tempat bekerja
13. Memilih tempat bekerja
14. Aplikasi melakukan permintaan data tempat bekerja
15. Menampilkan data tempat bekerja
16. Admin memilih export
17. Aplikasi melakukan permintaan data export
18. Menyimpan file export

14. *Sequence Diagram* Admin Rekap Tracer Alumni



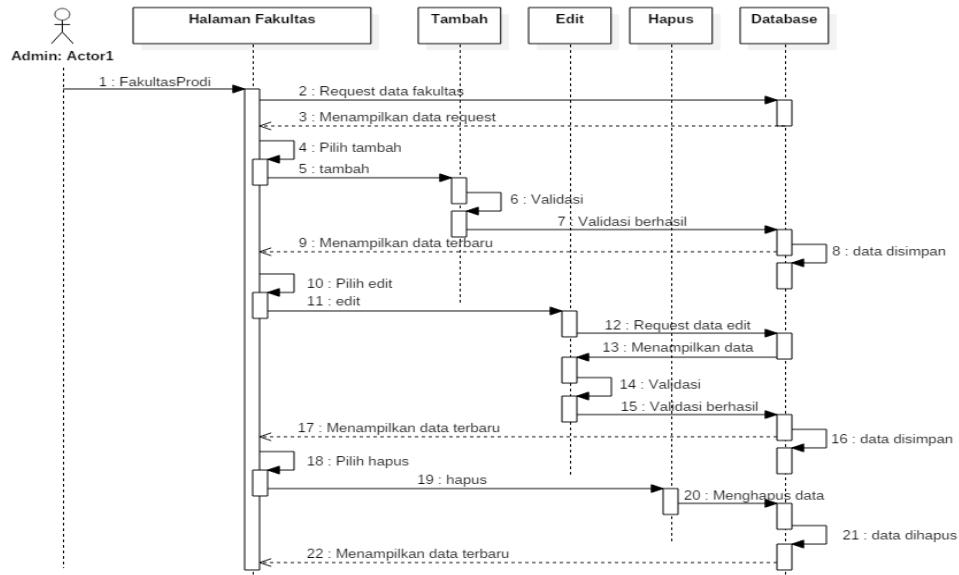
Gambar 4.56 *Sequence Diagram* Admin Rekap Tracer Alumni

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Rekap Tracer Alumni:

1. Admin memilih menu data rekap tracer alumni
2. Admin memilih filter tahun
3. Aplikasi melakukan permintaan data yang difilter

4. Menampilkan data yang difilter
5. Admin memilih export
6. Aplikasi melakukan permintaan data export
7. Menyimpan file export

15. Sequence Diagram Admin Fakultas



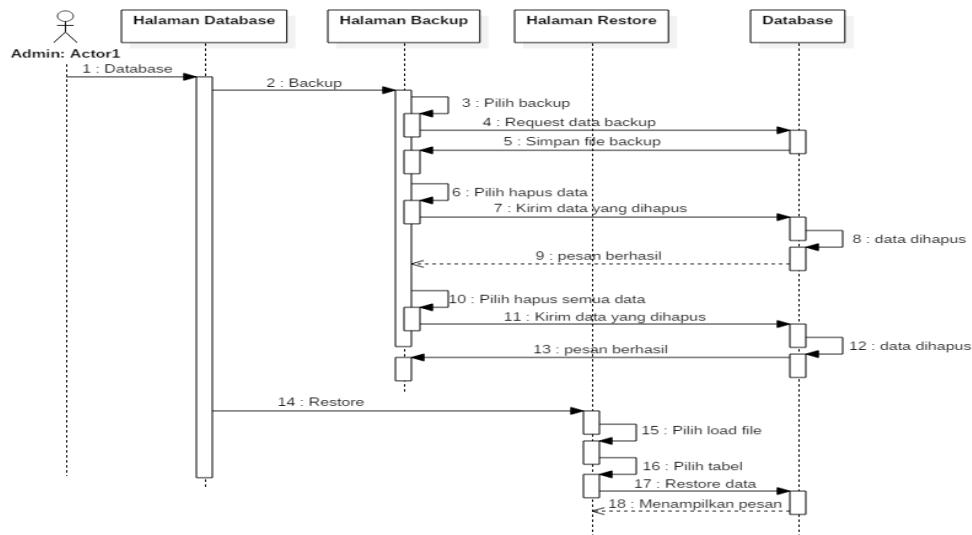
Gambar 4.57 Sequence Diagram Admin Fakultas

Berikut ini adalah penjelasan dari *Sequence Diagram* Admin Fakultas:

1. Admin memilih menu fakultas
2. Aplikasi melakukan permintaan data fakultas dan program studi
3. Menampilkan data fakultas dan program studi
4. Admin memilih menu tambah
5. Menampilkan halaman tambah
6. Validasi terhadap inputan tambah fakultas
7. Validasi berhasil menyimpan data ke database
8. Data disimpan
9. Menampilkan data terbaru
10. Admin memilih menu edit
11. Menampilkan halaman edit
12. Aplikasi melakukan permintaan data fakultas dan program studi
13. Menampilkan data fakultas dan program studi
14. Validasi terhadap inputan edit fakultas

15. Validasi berhasil menyimpan data ke database
16. Data disimpan
17. Menampilkan data terbaru
18. Admin memilih hapus
19. Menampilkan halaman hapus
20. Aplikasi mengirim perintah menghapus data fakultas
21. Data disimpan
22. Menampilkan data terbaru

16. Sequence Diagram Admin Database



Gambar 4.58 Sequence Diagram Admin Database

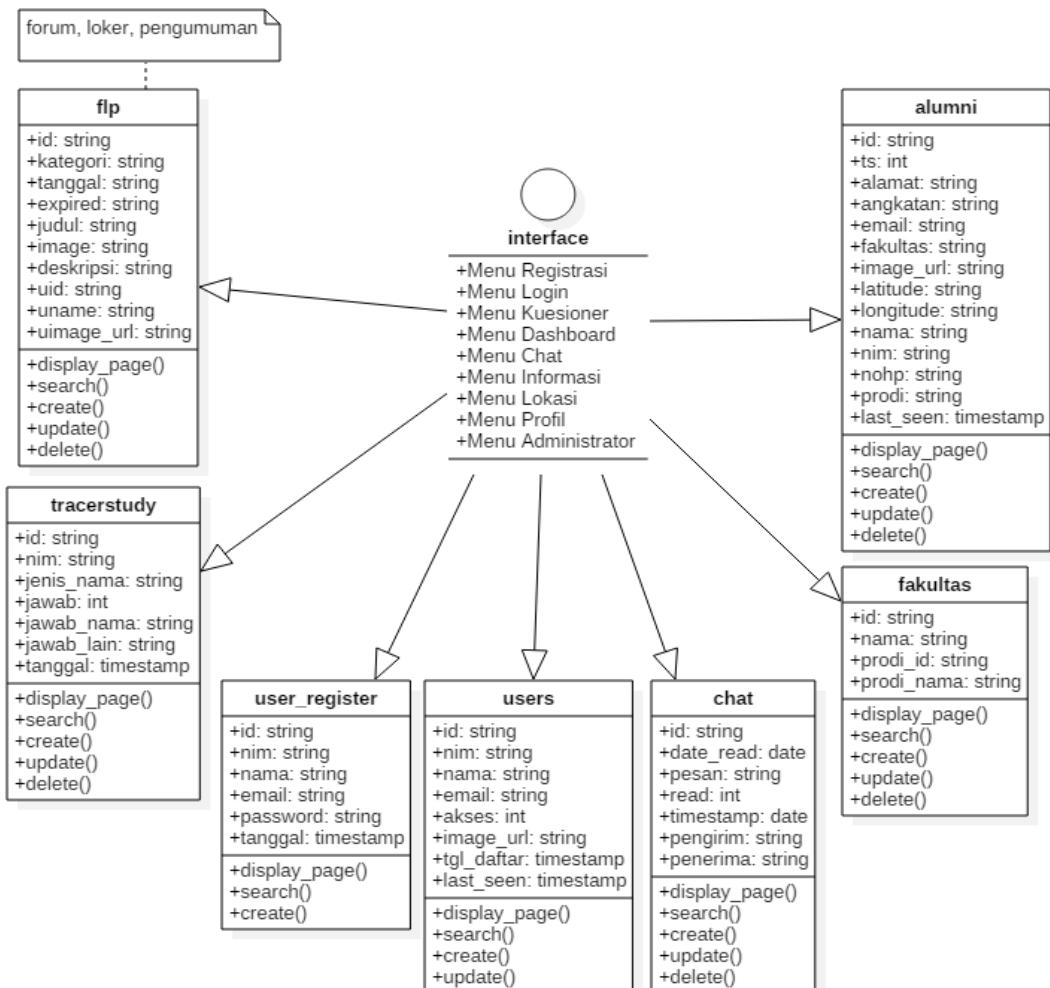
Berikut ini adalah penjelasan dari *Sequence Diagram Admin Database*:

1. Admin memilih menu *database*
2. Admin memilih tab *backup*
3. Admin memilih *backup*
4. Aplikasi melakukan permintaan data *backup*
5. Menyimpan *file backup*
6. Admin memilih hapus data
7. Aplikasi mengirim data yang akan dihapus
8. Data berhasil dihapus
9. Menampilkan pesan berhasil
10. Admin memilih hapus semua data
11. Aplikasi mengirim data yang akan dihapus

12. Data berhasil dihapus
13. Menampilkan pesan berhasil
14. Admin memilih tab *restore*
15. Admin memilih *load file*
16. Admin memilih list data tabel
17. Aplikasi mengirim data yang akan *diresetore*
18. Menampilkan pesan berhasil

4.2.4 Class Diagram

Class Diagram merupakan diagram untuk menampilkan beberapa kelas yang ada dalam sistem perangkat lunak yang akan dikembangkan. *Class diagram* menunjukkan hubungan antar kelas dalam aplikasi yang akan dibangun. Berikut digambar *Class Diagram*.



Gambar 4.59 Class Diagram

Class Diagram diatas ini akan menjelaskan tabel-tabel yang akan digunakan dalam perancangan *Unified Modelling Language* (UML) yaitu:

1. Tabel alumni

Nama Tabel : alumni

Primary Key : id,nim

Keterangan : tabel ini berisikan data-data alumni

Tabel 4.11 Tabel alumni

| No | Field | Type | Keterangan |
|----|-----------|-----------|--|
| 1 | id | String | Data unik |
| 2 | ts | Int | 0=belum, 1=sudah mengisi tracer alumni |
| 3 | alamat | String | Alamat alumni |
| 4 | angkatan | Int | Tahun Angkatan alumni |
| 5 | email | String | Email alumni |
| 6 | fakultas | String | Fakultas alumni |
| 7 | image_url | String | Link url gambar |
| 8 | latitude | String | Posisi map lokasi latitude |
| 9 | longitude | String | Posisi map lokasi longitude |
| 10 | nama | String | Nama alumni |
| 11 | nim | String | Nomor induk mahasiswa |
| 12 | nohp | String | Nomor handphone |
| 13 | prodi | String | Program studi |
| 14 | last_seen | Timestamp | Tanggal terakhir akses aplikasi |

2. Tabel chat

Nama Tabel : chat

Primary Key : id

Keterangan : tabel ini berisikan data-data chat pengguna aplikasi

Tabel 4.12 Tabel chat

| No | Field | Type | Keterangan |
|----|-----------|-----------|---|
| 1 | id | String | Data unik menggunakan penerima_pengirim |
| 2 | date_read | Timestamp | Tanggal saat membaca pesan |

| | | | |
|---|-----------|-----------|--|
| 3 | penerima | String | Nim penerima |
| 4 | pengirim | String | Nim pengirim |
| 5 | pesan | String | Pesan text chatting |
| 6 | timestamp | Timestamp | Tanggal saat mengirim pesan |
| 7 | read | Int | 0 = pesan belum dibaca, 1 = pesan sudah dibaca |

3. Tabel fakultas

Nama Tabel : flp

Primary Key : id

Keterangan : tabel ini berisikan data-data fakultas dan program studi

Tabel 4.13 Tabel fakultas

| No | Field | Type | Keterangan |
|----|------------|--------|-------------------------|
| 1 | id | String | Data unik Fakultas |
| 2 | nama | String | Nama Fakultas |
| 3 | prodi_id | String | Data Unik Program studi |
| 4 | prodi_nama | String | Nama Program studi |

4. Tabel flp

Nama Tabel : flp

Primary Key : id

Keterangan : tabel ini berisikan data-data informasi forum, lowongan pekerjaan dan pengumuman

Tabel 4.14 Tabel flp

| No | Field | Type | Keterangan |
|----|-----------|-----------|-------------------------------------|
| 1 | id | String | Data unik menggunakan random teks |
| 2 | kategori | String | Kategori (forum, loker, pengumuman) |
| 3 | tanggal | Timestamp | Tanggal saat input |
| 4 | expired | Timestamp | Tanggal kadaluarsa sesuai yg dpilih |
| 5 | judul | String | Berisi judul pada konten |
| 6 | image_url | String | Link url gambar |
| 7 | deskripsi | String | Text deskripsi pada konten |

| | | | |
|----|------------|--------|-----------------|
| 8 | uid | String | User input nim |
| 9 | uname | String | User input nama |
| 10 | uimage_url | String | Link url gambar |

5. Tabel *tracerstudy*

Nama Tabel : *tracerstudy*

Primary Key : id, nim

Keterangan : tabel ini berisikan data-data tracer study alumni

Tabel 4.15 Tabel tracerstudy

| No | Field | Type | Keterangan |
|----|------------|-----------|---------------------------------|
| 1 | id | String | Data unik menggunakan nim |
| 2 | nim | String | Nomor induk mahasiswa |
| 3 | jenis | String | Jenis pertanyaan kuesioner |
| 4 | jenis_nama | String | Nama pertanyaan kuesioner |
| 5 | jawab | Int | Jawab poin pertanyaan kuesioner |
| 6 | jawab_nama | String | Jawab nama pertanyaan kuesioner |
| 7 | jawab_lain | String | Jawab lain pertanyaan kuesioner |
| 8 | tanggal | Timestamp | Tanggal saat input kuesioner |

6. Tabel users

Nama Tabel : users

Primary Key : id, nim

Keterangan : tabel ini berisikan data-data user admin dan alumni

Tabel 4.16 Tabel users

| No | Field | Type | Keterangan |
|----|-----------|--------|---------------------------|
| 1 | id | String | Data unik menggunakan nim |
| 2 | nim | String | Nomor induk mahasiswa |
| 3 | nama | String | Nama pengguna |
| 4 | email | String | Email pengguna |
| 6 | akses | Int | 0 = user dan 1 = admin |
| 8 | image_url | String | Link url gambar |

| | | | |
|----|------------|-----------|--|
| 9 | tgl_daftar | Timestamp | Tanggal daftar pertama user melakukan registrasi |
| 10 | last_seen | Timestamp | Tanggal terakhir akses aplikasi |

7. Tabel users_register

Nama Tabel : users_register

Primary Key : id

Keterangan : tabel ini berisikan data-data user baru

Tabel 4.17 Tabel users_register

| No | Field | Type | Keterangan |
|----|----------|-----------|-----------------------|
| 1 | id | String | Data unik |
| 2 | nim | String | Nomor induk mahasiswa |
| 3 | nama | String | Nama pengguna |
| 4 | email | String | Email pengguna |
| 5 | password | String | Password |
| 6 | tanggal | Timestamp | Tanggal daftar |

4.2.5 Desain

1) Perancangan Antarmuka

Implementasi sistem aplikasi ini akan menjelaskan secara menyeluruh tentang aplikasi *Tracer Alumni* dan menjelaskan fitur-fitur yang dimiliki yaitu diagram model untuk user dan diagram model untuk admin. Berikut diagram modelnya:

1. Diagram Model Admin

Pada Diagram Model Admin memiliki fitur menu login, reset password, reset password, dashboard, chat, informasi angkatan, informasi tempat bekerja alumni, informasi forum, informasi loker, informasi pengumuman, lokasi, profil, administrator pengelolaan user, administrator data alumni, administrator rekap tracer alumni, administrator fakultas dan administrator database yang dapat dilihat pada gambar 4.20 Diagram Model Admin, yang dimiliki oleh seorang admin untuk mengelola data jika belum ada aplikasi tracer alumni langsung dengan

menginstal aplikasi akses di playstore dengan kata kunci “**Tracer Alumni Unibba**”. Admin harus melakukan login terlebih mengisi form di halaman login dan mengisi email dan *password*. Jika data form yang di isi sudah dilakukan dan melakukan dan admin menekan tombol login, maka reaksi sistem akan memvalidasi jika data pada form login yang di isikan benar maka akan menuju halaman dashboard, namun jika salah akan menampilkan informasi pesan salah. Fitur menu pada diagram model admin memiliki fungsi diantaranya Menu Login berfungsi sebagai tampilan awal dengan memilih untuk *login* untuk masuk ke aplikasi atau *forgot password* jika lupa *password* masuk aplikasi.

Menu Dashboard berfungsi sebagai tampilan awal dengan halaman data grafik dan pencarian alumni. pada saat admin berhasil melakukan proses login pada halaman login, maka akan di arahkan pada menu tersebut.

Menu Chat berfungsi pada saat admin memilih maka reaksi sistem akan menampilkan struktur dokumen pada halaman chat, yang dimana admin dapat melakukan komunikasi sesama pengguna aplikasi.

Menu Informasi Angkatan berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi angkatan, yang dimana admin dapat memilih angkatan dan melihat daftar alumni yang sesuai dengan angkatan yang dipilih.

Menu Informasi Tempat Bekerja Alumni berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi tempat bekerja alumni, yang dimana admin dapat memilih tempat bekerja dan melihat daftar alumni yang sesuai dengan tempat bekerja alumni yang dipilih.

Menu Informasi Forum berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi forum, yang dimana admin dapat melakukan posting dan dapat menghapus semua konten postingan user pada menu tersebut.

Menu Informasi Loker berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi loker, yang dimana admin dapat melakukan posting dan dapat menghapus semua konten postingan user pada menu tersebut.

Menu Informasi Pengumuman berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi pengumuman, yang dimana admin dapat melakukan posting dan dapat menghapus semua konten postingan user pada menu tersebut.

Menu Lokasi berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman lokasi, yang dimana admin dapat melakukan pencarian lokasi keberadaan alumni dengan tampilan map beserta koordinat yang terdapat pada menu tersebut.

Menu Profil berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman profil, yang dimana admin dapat melakukan perubahan informasi data pribadi, password dan logout keluar aplikasi yang terdapat pada menu tersebut.

Menu Administrator berfungsi sebagai tampilan menu admin yang didalamnya terdapat menu pengelolaan user, data alumni, rekap tracer alumni, fakultas dan database.

Menu Pengelolaan User berfungsi pada saat admin memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman pengelolaan user registrasi dan users yang dimana pada menu registrasi admin dapat mengelola data validasi dan menolak user yang baru mendaftar. pada menu users admin dapat melakukan penambahan, perubahan email dan perubahan akses sesuai dengan yang dipilih.

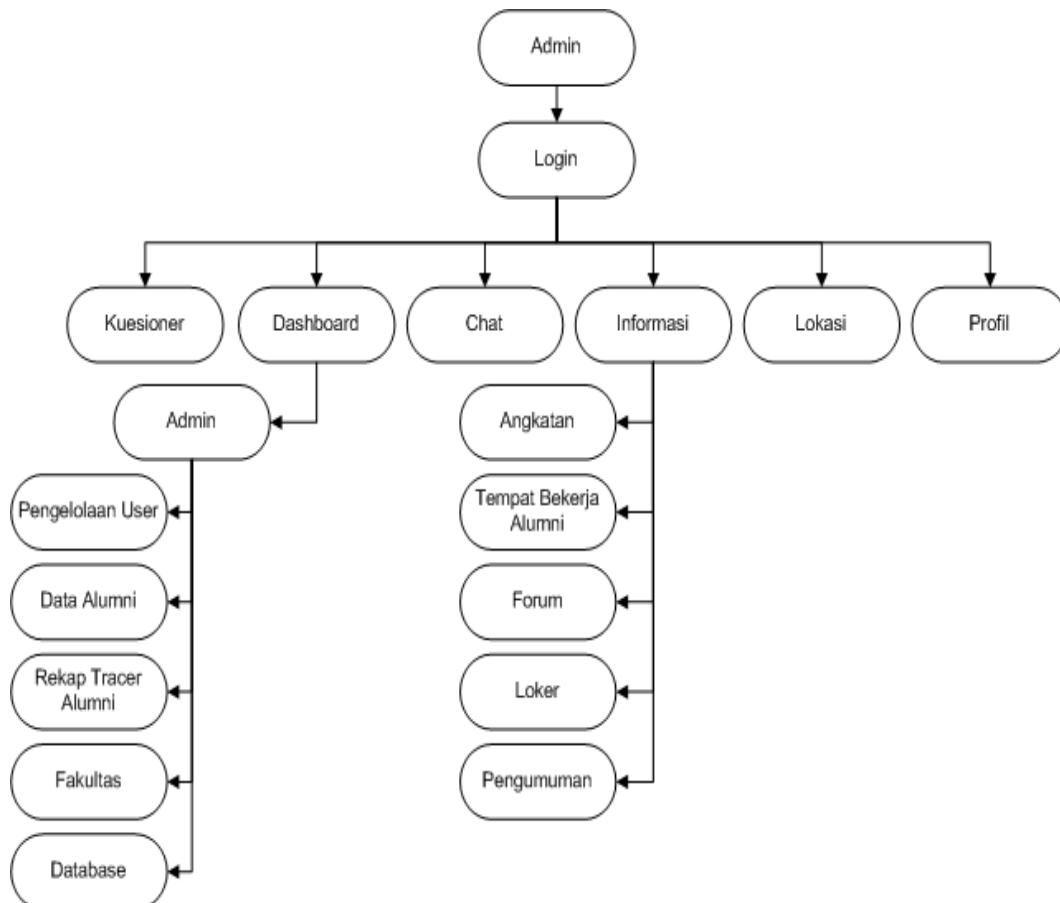
Menu Data Alumni berfungsi pada saat admin memilih menu data alumni, maka reaksi sistem akan menampilkan struktur dokumen pada halaman menu data alumni angkatan dan tempat bekerja alumni. pada halaman informasi angkatan admin dapat memilih angkatan dan melihat daftar alumni yang sesuai dengan angkatan yang dipilih kemudian admin bisa mengexport data tersebut ke format file csv, json dan pdf. Pada halaman tempat bekerja alumni admin dapat memilih tempat bekerja dan melihat daftar alumni yang sesuai dengan tempat bekerja alumni yang dipilih kemudian admin bisa mengexport data tersebut ke format file csv dan pdf.

Menu Rekap *Tracer* Alumni berfungsi pada saat admin memilih menu Rekap *Tracer* Alumni, maka reaksi sistem akan menampilkan struktur dokumen

pada halaman menu rekap *tracer* alumni, yang dimana admin dapat mengelola data dan cetak data hasil *tracer* alumni dalam format csv dan pdf.

Menu Fakultas berfungsi pada saat admin memilih maka reaksi sistem akan menampilkan struktur dokumen pada halaman fakultas. Halaman ini merupakan gabungan antara data fakultas sebagai induknya dan program studi sebagai data anaknya fakultas. yang dimana admin dapat melakukan penambahan, perubahan dan hapus data.

Menu Database berfungsi pada saat admin memilih maka reaksi sistem akan menampilkan struktur dokumen pada halaman database *backup* dan *restore*. Pada menu backup admin dapat melakukan backup data kemudian disimpan dalam format file .json pada penyimpanan *smartphone* tersebut dan hapus data sesuai dengan yang dipilih.



Gambar 4.60 Diagram model admin

2. Diagram Model Alumni

Pada Diagram model alumni ini memiliki fitur menu registrasi, login, reset password, dashboard, chat, informasi angkatan, informasi tempat bekerja alumni,

informasi forum, informasi loker, informasi pengumuman, lokasi dan profil yang dapat dilihat pada Gambar 4.30 Diagram model alumni, yang dimiliki oleh seorang user harus menginstal aplikasi yang sudah disediakan di playstore dengan mengakses di playstore dengan kata kunci “**Tracer Alumni Unibba**”. Fitur menu pada diagram model alumni ini memiliki fungsi diantaranya menu registrasi berfungsi pada saat alumni memilih menu registrasi, reaksi sistem akan menampilkan halaman registrasi, jika pada saat proses registrasi benar maka akan menampilkan halaman menunggu validasi, jika registrasi data sudah ada maka akan menampilkan halaman notifikasi data sudah terdaftar dan jika registrasi data salah maka akan menampilkan halaman notifikasi data yang di inputkan tidak benar.

Menu Login berfungsi sebagai tampilan awal dengan memilih untuk *login* untuk masuk ke aplikasi atau *forgot password* jika lupa *password* masuk aplikasi.

Menu Dashboard berfungsi sebagai tampilan awal dengan halaman pengumuman dan pencarian alumni. pada saat alumni berhasil melakukan proses login pada halaman login, maka akan di arahkan pada menu tersebut.

Menu Chat berfungsi pada saat alumni memilih maka reaksi sistem akan menampilkan struktur dokumen pada halaman chat, yang dimana alumni dapat melakukan komunikasi sesama pengguna aplikasi.

Menu Informasi Angkatan berfungsi pada saat alumni memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi angkatan, yang dimana alumni dapat memilih angkatan dan melihat daftar alumni yang sesuai dengan angkatan yang dipilih.

Menu Informasi Tempat Bekerja Alumni berfungsi pada saat alumni memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi tempat bekerja alumni, yang dimana admin dapat memilih tempat bekerja dan melihat daftar alumni yang sesuai dengan tempat bekerja alumni yang dipilih.

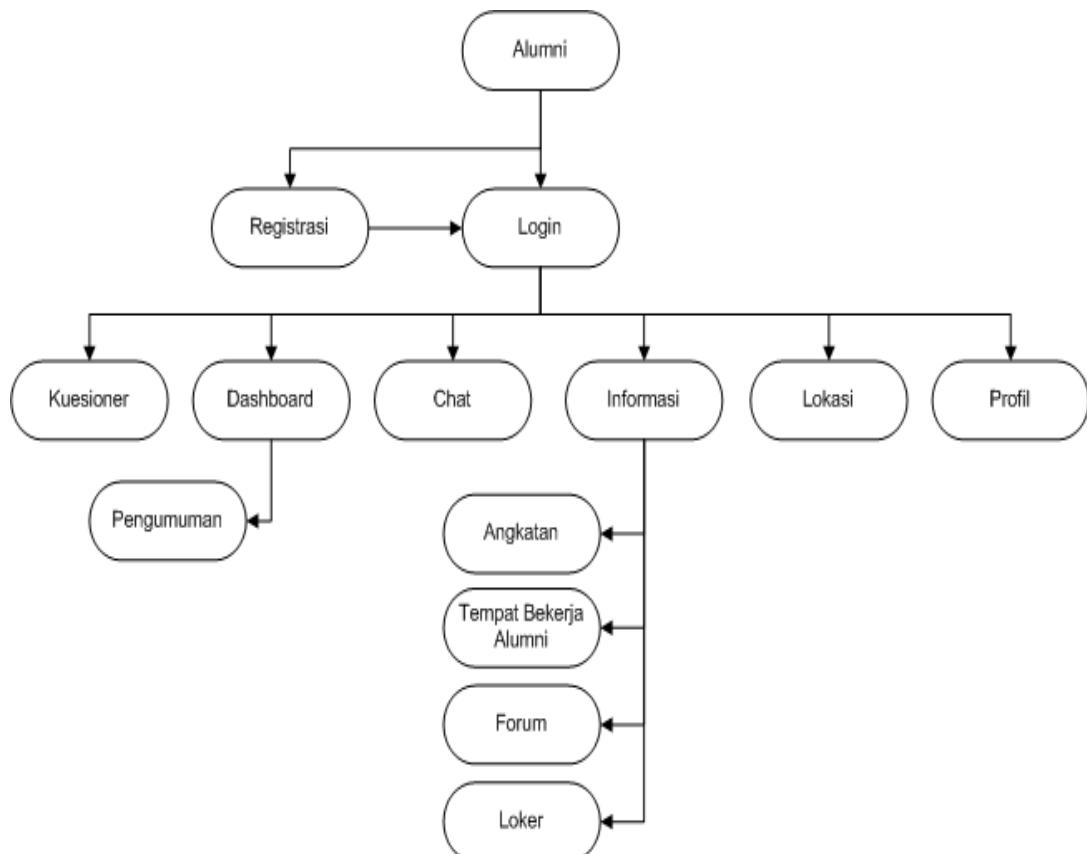
Menu Informasi Forum berfungsi pada saat alumni memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi forum, yang dimana alumni dapat melakukan posting dan dapat menghapus postingan sendiri pada menu tersebut.

Menu Informasi Loker berfungsi pada saat alumni memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi loker, yang dimana alumni dapat melakukan posting dan dapat menghapus postingan sendiri pada menu tersebut.

Menu Informasi Pengumuman berfungsi pada saat alumni memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman informasi pengumuman, yang dimana alumni dapat melakukan posting dan dapat menghapus postingan sendiri pada menu tersebut.

Menu Lokasi berfungsi pada saat alumni memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman lokasi, yang dimana alumni dapat melakukan pencarian lokasi keberadaan alumni dengan tampilan map beserta koordinat yang terdapat pada menu tersebut.

Menu Profil berfungsi pada saat alumni memilih, maka reaksi sistem akan menampilkan struktur dokumen pada halaman profil, yang dimana alumni dapat melakukan perubahan informasi data pribadi, set lokasi, password dan logout keluar aplikasi yang terdapat pada menu tersebut.



Gambar 4.61 *Diagram model alumni*

2) Perancangan Antarmuka Aplikasi (User Interface)

Perancangan Antarmuka Aplikasi adalah perancangan User Interface yang merupakan bentuk tampilan grafis yang berhubungan langsung dengan pengguna (user). Antarmuka pengguna berfungsi untuk menghubungkan antara pengguna dengan sistem.

1. Perancangan Halaman Gagal Koneksi Internet

Pada perancangan Antarmuka halaman Gagal Koneksi Internet yang dapat dilihat pada Gambar 4.62 *User Interface* Halaman Login, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Coba lagi, untuk melakukan sinkron data kembali



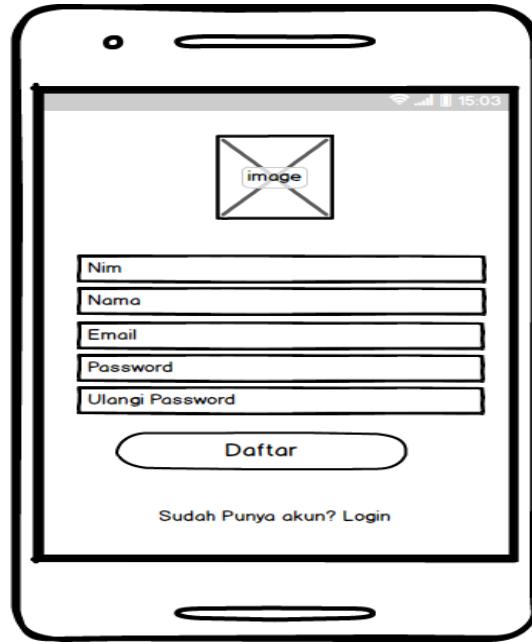
Gambar 4.62 *User Interface* Halaman Gagal Koneksi Internet

2. Perancangan Halaman Registrasi

Pada perancangan Antarmuka halaman Registrasi yang dapat dilihat pada Gambar 4.63 *User Interface* Halaman Registrasi, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Daftar, untuk registrasi akun baru
2. Login, untuk masuk ke sistem aplikasi



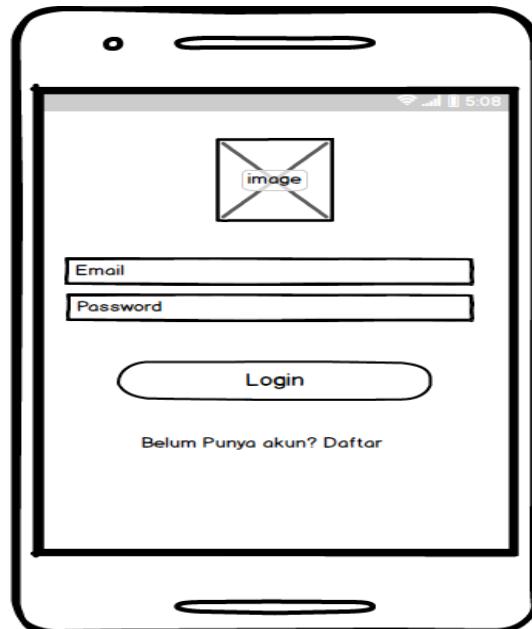
Gambar 4.63 *User Interface* Halaman Registrasi

3. Perancangan Halaman Login

Pada perancangan Antarmuka halaman Login yang dapat dilihat pada Gambar 4.64 *User Interface* Halaman Login, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Login, untuk masuk ke sistem aplikasi
2. Daftar, untuk registrasi akun baru



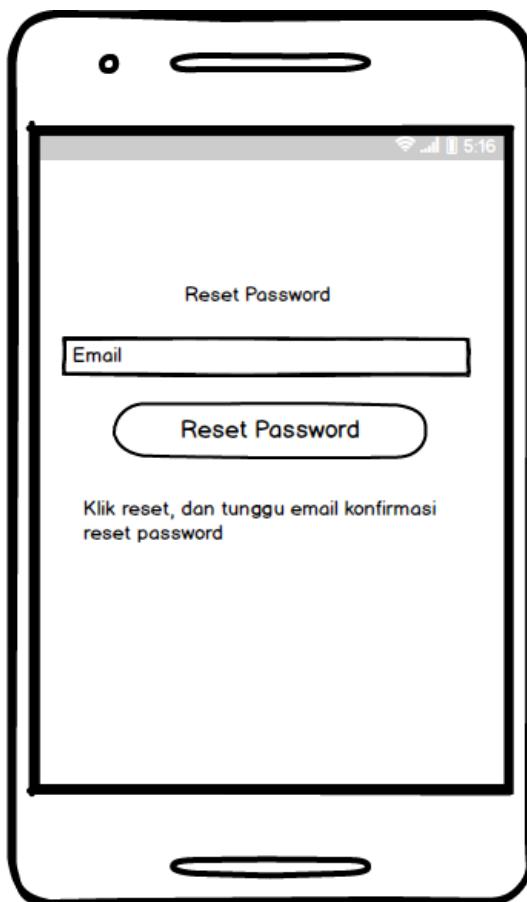
Gambar 4.64 *User Interface* Halaman Login

4. Perancangan Halaman *Reset Password*

Pada perancangan Antarmuka halaman *reset password* yang dapat dilihat pada Gambar 4.65 *User Interface* Halaman *Reset Password*, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Login, untuk masuk ke sistem aplikasi
2. Daftar, untuk registrasi akun baru



Gambar 4.65 *User Interface* Halaman *Reset Password*

5. Perancangan Halaman Registrasi Menunggu Validasi

Pada perancangan Antarmuka halaman Registrasi Menunggu Validasi yang dapat dilihat pada Gambar 4.66 *User Interface* Halaman Halaman Registrasi Menunggu Validasi, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Tutup, untuk keluar dari aplikasi



Gambar 4.66 *User Interface* Halaman Registrasi Menunggu Validasi

6. Perancangan Halaman Registrasi Gagal Validasi

Pada perancangan Antarmuka halaman Registrasi Gagal Validasi yang dapat dilihat pada Gambar 4.67 User Interface Halaman Registrasi Gagal Validasi, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Daftar kembali, untuk registrasi akun baru
2. Tutup, untuk keluar dari aplikasi



Gambar 4.67 User Interface Halaman Registrasi Gagal Validasi

7. Perancangan Halaman Registrasi Berhasil Validasi

Pada perancangan Antarmuka halaman Registrasi Berhasil Validasi yang dapat dilihat pada Gambar 4.68 Halaman Registrasi Berhasil Validasi, dari aplikasi *Tracer* Alumni di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Mulai, untuk memulai mengisi kuesioner



Gambar 4.68 Halaman Registrasi Berhasil Validasi

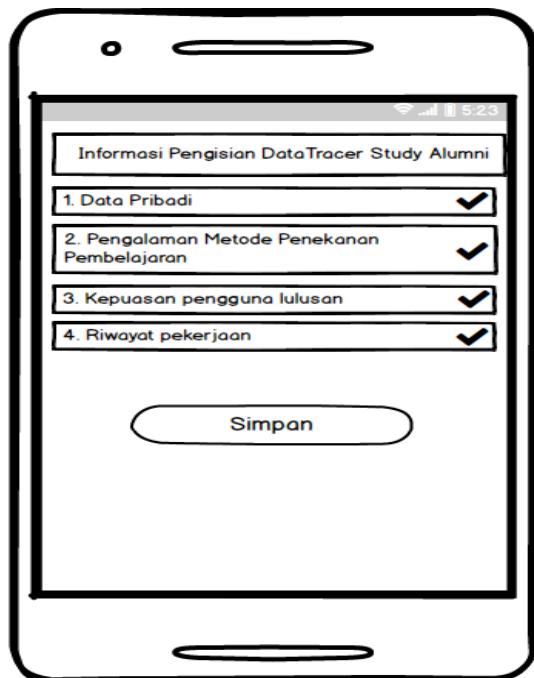
8. Perancangan Halaman Kuesioner

Pada perancangan Antarmuka halaman Kuesioner yang dapat dilihat pada Gambar 4.34 Halaman Kuesioner Data Pribadi, Gambar 4.35 Halaman Kuesioner Pembelajaran, Gambar 4.69 Halaman Kuesioner Pekerjaan, dari aplikasi *Tracer* Alumni di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Data Pribadi, untuk menampilkan Data pribadi dapat dilihat Gambar 4.70 Halaman Kuesioner Data Pribadi
2. Pengalaman Metode Penekanan Pembelajaran, untuk menampilkan Pembelajaran dapat dilihat Gambar 4.71 Halaman Kuesioner Pengalaman Metode Penekanan Pembelajaran

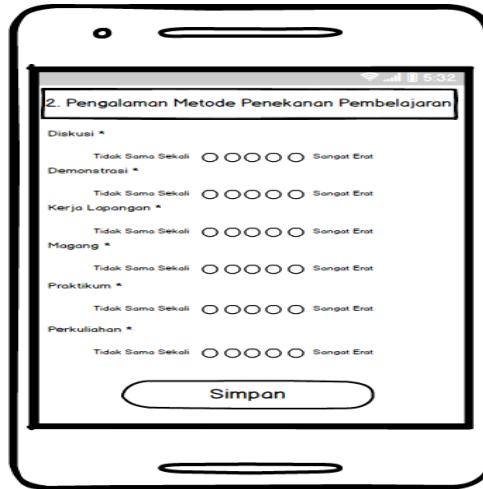
3. Kepuasan Pengguna Lulusan, untuk menampilkan Kepuasan Pengguna Lulusan dapat dilihat Gambar 4.72 Halaman Kepuasan Pengguna Lulusan
4. Riwayat Pekerjaan, untuk menampilkan Riwayat Pekerjaan dapat dilihat Gambar 4.73 Halaman Kuesioner Riwayat Pekerjaan
5. Simpan, untuk menyimpan data dari hasil inputan



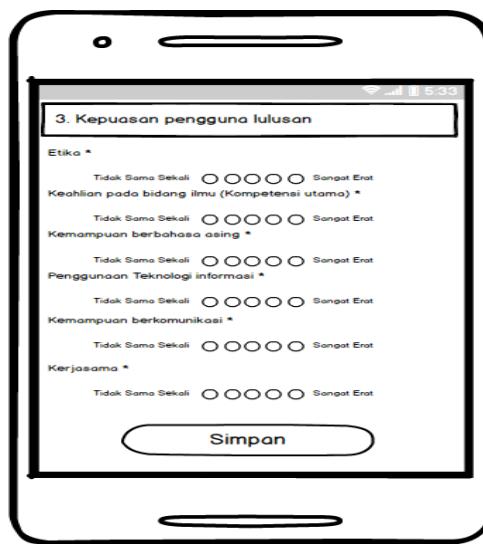
Gambar 4.69 Halaman Kuesioner Menu

A line drawing of a smartphone showing its front view. The screen displays a survey application titled "1. Data Pribadi". It contains several input fields with placeholder text and validation notes: "Fit. Identitas" (Nomor Pokok Mahasiswa *), "Nama Lengkap *" (disediakan dengan nama lengkap), "Nomor Telepon/HP *" (disediakan dengan nomor handphone), "Email address *", "Program Studi *" (disediakan dengan program studi Anda), and "Angkatan *" (disediakan dengan angkatan tahun masuk). At the bottom of the screen is a blue rounded rectangular button labeled "Simpan".

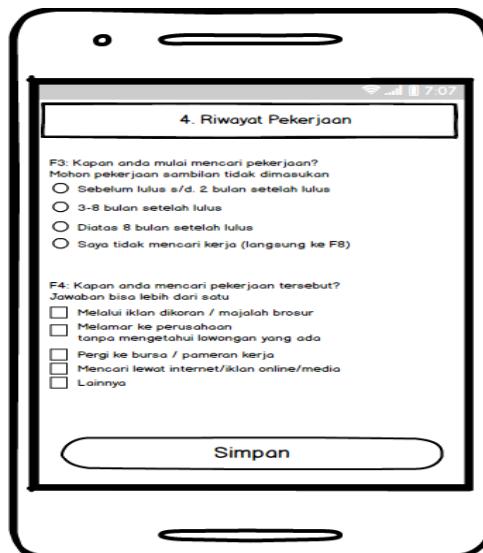
Gambar 4.70 Halaman Kuesioner Data Pribadi



Gambar 4.71 Halaman Kuesioner Pengalaman Metode Pembelajaran



Gambar 4.72 Halaman Kuesioner Kepuasan Pengguna Lulusan



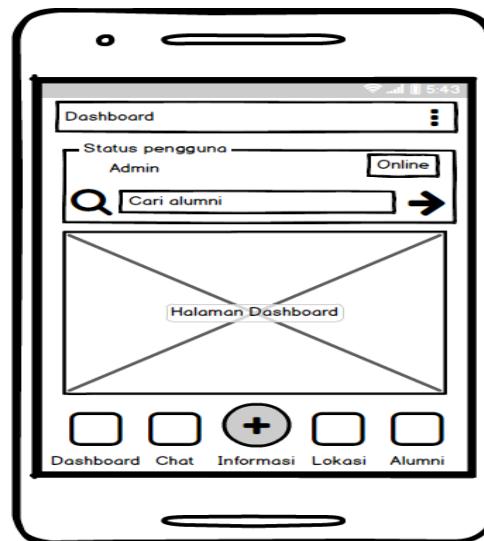
Gambar 4.73 Halaman Kuesioner Riwayat Pekerjaan

9. Perancangan Halaman Dashboard

Pada perancangan Antarmuka halaman Dashboard yang dapat dilihat pada Gambar 4.38 Halaman Dashboard, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Dashboard, untuk memilih halaman Dashboard
2. Chat, untuk memilih halaman Chat
3. Informasi, untuk memilih halaman Informasi
4. Lokasi, untuk memilih halaman Lokasi
5. Profil, untuk memilih halaman Profil



Gambar 4.74 Halaman Dashboard

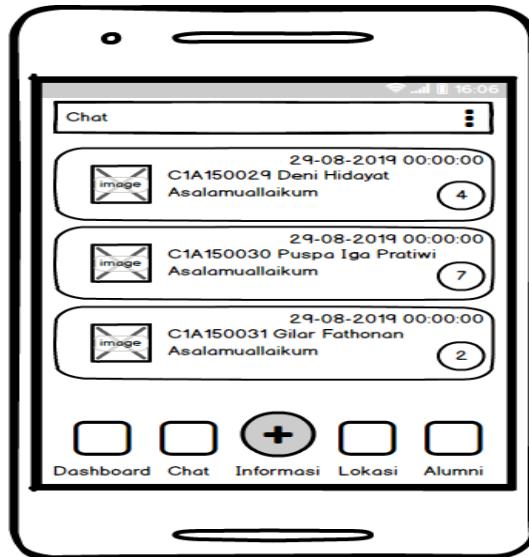
10. Perancangan Halaman Chat

Pada perancangan Antarmuka halaman Chat yang dapat dilihat pada Gambar 4.39 Halaman Chat, Gambar 4.40 Halaman Chat Kirim, Gambar 4.41 Halaman Chat Lihat Gambar, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

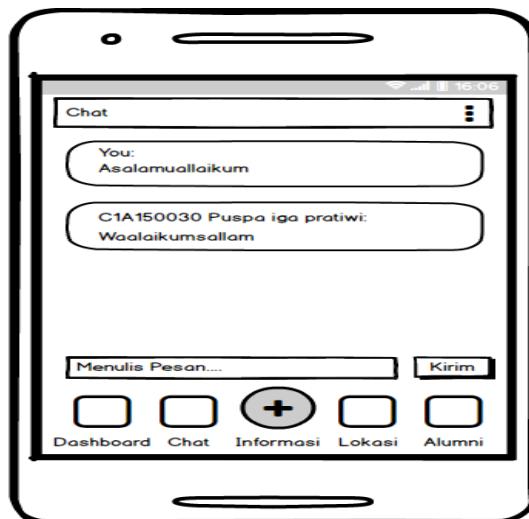
Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Dashboard, untuk memilih halaman Dashboard
2. Chat, untuk memilih halaman Chat
3. Informasi, untuk memilih halaman Informasi
4. Lokasi, untuk memilih halaman Lokasi

5. Profil, untuk memilih halaman Profil
6. Kirim, untuk mengirim pesan
7. *Share*, untuk berbagi gambar
8. *Save to Gallery*, untuk simpan gambar ke gallery hp



Gambar 4.75 Halaman Chat



Gambar 4.76 Halaman Chat Kirim

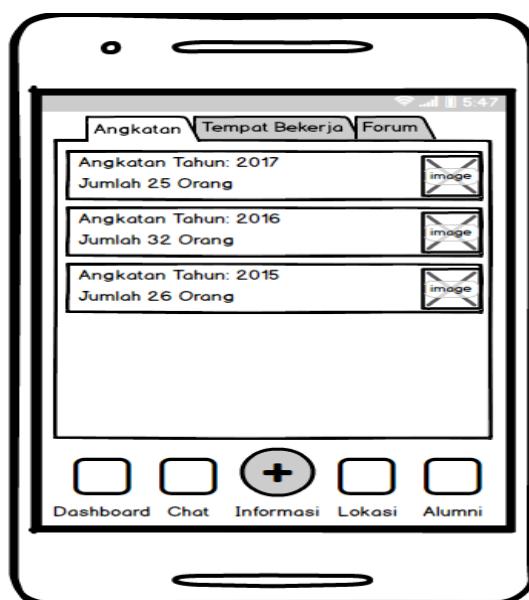
11. Perancangan Halaman Informasi

Pada perancangan Antarmuka halaman Informasi yang dapat dilihat pada Gambar 4.42 Halaman Informasi, dari aplikasi *Tracer* Alumni di Fakultas Teknologi Informasi Universitas Bale Bandung.

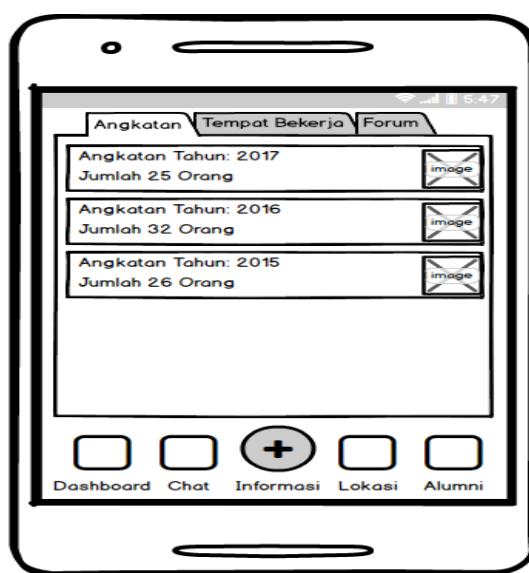
Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Tab Angkatan untuk menampilkan halaman angkatan

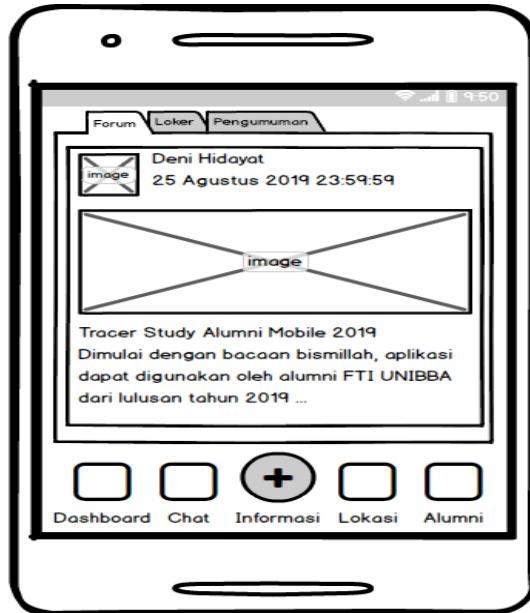
2. Tab Tempat Bekerja Alumni untuk menampilkan halaman tempat bekerja alumni
3. Tab Forum untuk menampilkan halaman Forum
4. Tab Loker untuk menampilkan halaman Loker
5. Tab Pengumuman untuk menampilkan halaman Pengumuman
6. Tab FLP (Forum loker dan pengumuman) halaman mengacu kepada Gambar 4.79 Halaman Informasi FLP
7. Form inputan FLP (Forum loker dan pengumuman) halaman mengacu kepada Gambar 4.80 Halaman Informasi Form FLP



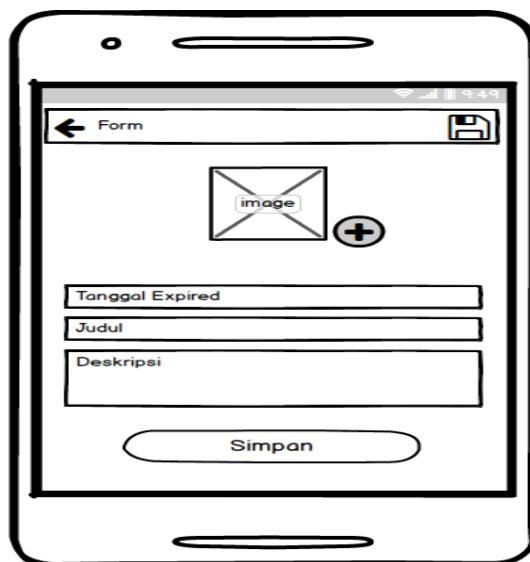
Gambar 4.77 Halaman Informasi Angkatan



Gambar 4.78 Halaman Informasi Tempat Bekerja Alumni



Gambar 4.79 Halaman Informasi FLP



Gambar 4.80 Halaman Informasi Form FLP

12. Perancangan Halaman Lokasi

Pada perancangan Antarmuka halaman Lokasi yang dapat dilihat pada Gambar 4.81 Halaman Lokasi, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Cari untuk mencari berdasarkan program studi dan tahun angkatan
2. Marker atau penanda di map ketika di klik akan menampilkan informasi profil alumni sesuai dengan yang dipilih



Gambar 4.81 Halaman Lokasi

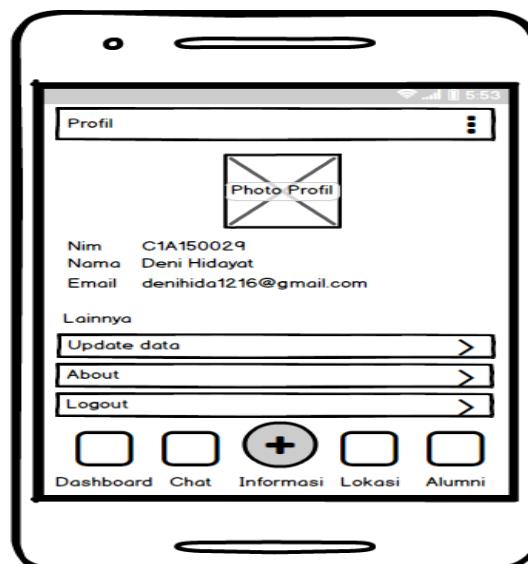
13. Perancangan Halaman Profil

Pada perancangan Antarmuka halaman Profil yang dapat dilihat pada Gambar 4.82 Halaman Profil, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

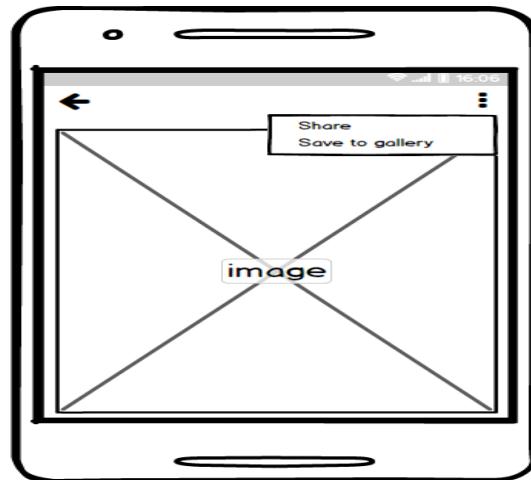
Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Photo, untuk merubah photo profil
2. Update data, memperbaharui data profil
3. About, informasi tentang aplikasi
4. Logout, keluar dari aplikasi
5. Photo profil ketika diklik menampilkan halaman Gambar 4.83

Halaman Profil Lihat Gambar



Gambar 4.82 Halaman Profil



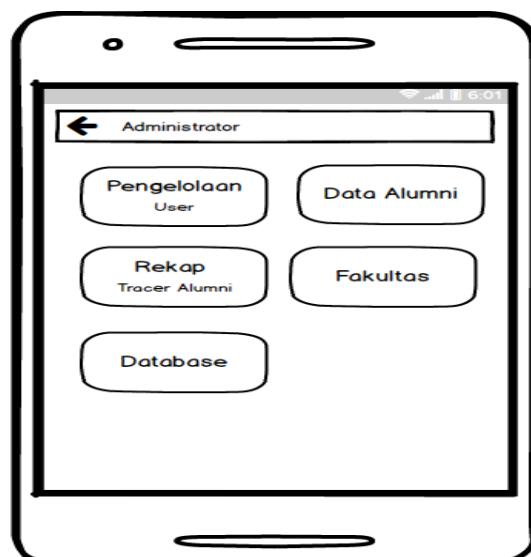
Gambar 4.83 Halaman Profil Lihat Gambar

14. Perancangan Halaman Administrator

Pada perancangan Antarmuka halaman Admin yang dapat dilihat pada Gambar 4.84 Halaman Administrator, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Pengelolaan User, untuk menampilkan menu Pengelolaan User
2. Data Alumni, untuk menampilkan menu Data Alumni
3. Rekap Tracer Alumni, untuk menampilkan menu Rekap Tracer Alumni
4. Fakultas, untuk menampilkan menu Fakultas
5. Database, untuk menampilkan menu Database



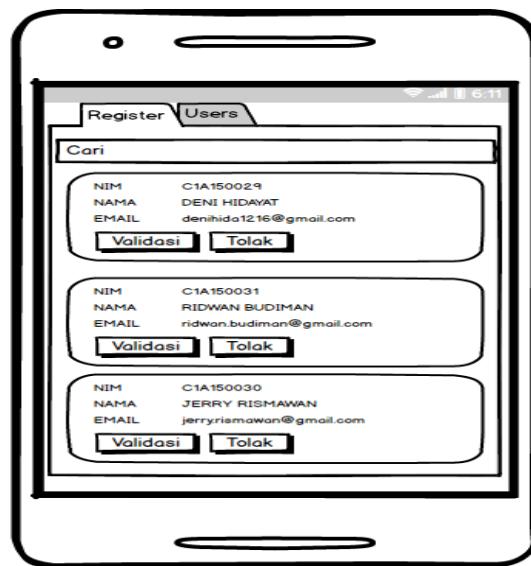
Gambar 4.84 Halaman Administrator

15. Perancangan Halaman Pengelolaan User

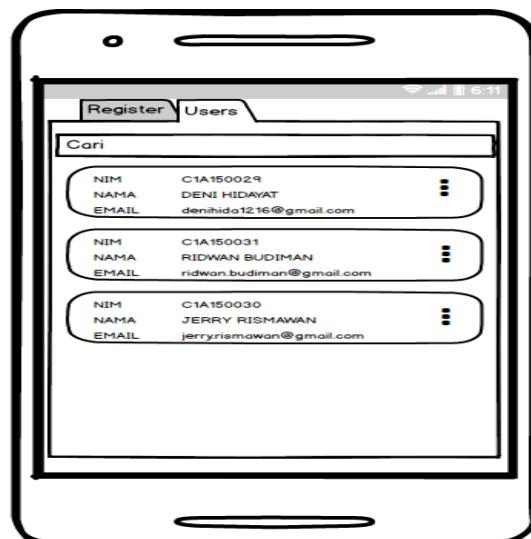
Pada perancangan Antarmuka halaman Pengelolaan User yang dapat dilihat pada Gambar 4.85 Halaman Pengelolaan User Register, Gambar 4.86 Halaman Pengelolaan User Users dari aplikasi *Tracer* Alumni di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button tabs* pada halaman ini adalah:

1. Tab Register, untuk menampilkan list user registrasi mengacu pada Gambar 4.85 Halaman Pengelolaan User Register
2. Tab Users, untuk menampilkan list users Users mengacu pada Gambar 4.86 Halaman Pengelolaan User Users



Gambar 4.85 Halaman Pengelolaan *User Register*



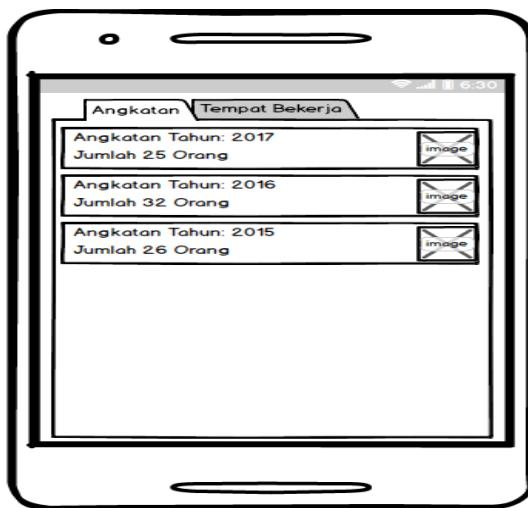
Gambar 4.86 Halaman Pengelolaan *User Users*

16. Perancangan Halaman Data Alumni

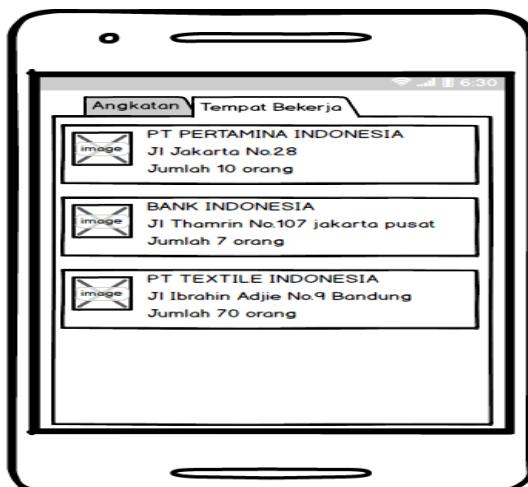
Pada perancangan Antarmuka halaman Pelaporan Data Alumni yang dapat dilihat pada Gambar 4.33 Halaman Pelaporan Data Alumni, dari aplikasi *Tracer* Alumni di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* tabs pada halaman ini adalah:

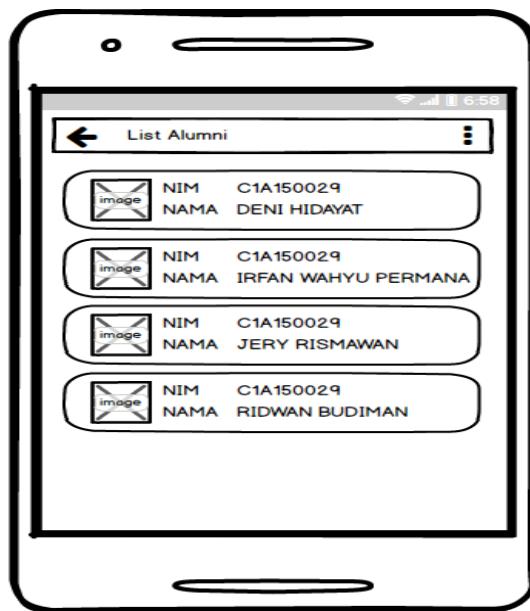
1. Tab Angkatan, untuk menampilkan list user registrasi mengacu pada Gambar 4.87 Halaman Data Alumni Angkatan
2. Tab Tempat Bekerja Alumni, untuk menampilkan list user registrasi mengacu pada Gambar 4.88 Halaman Data Alumni Tempat Bekerja
3. Halaman List Alumni, untuk menampilkan list alumni ketika memilih Angkatan atau tempat bekerja alumni. halaman mengacu pada Gambar 4.89 Halaman Data Alumni Tempat Bekerja



Gambar 4.87 Halaman Data Alumni Angkatan



Gambar 4.88 Halaman Data Alumni Tempat Bekerja



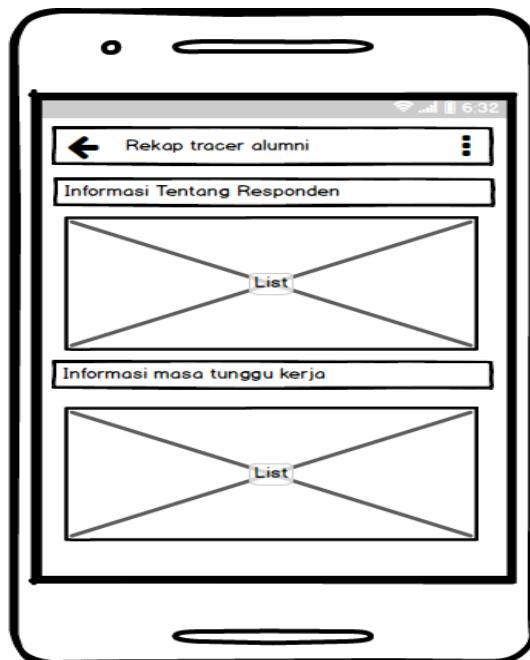
Gambar 4.89 Halaman Data Alumni List Alumni

17. Perancangan Halaman Rekap Tracer Alumni

Pada perancangan Antarmuka halaman Rekap Tracer Alumni yang dapat dilihat pada Gambar 4.90 Halaman Pelaporan Rekap Tracer Alumni, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Filter untuk melakukan filter dari inputan tahun
2. Titik tiga sebagai menu export file ke csv



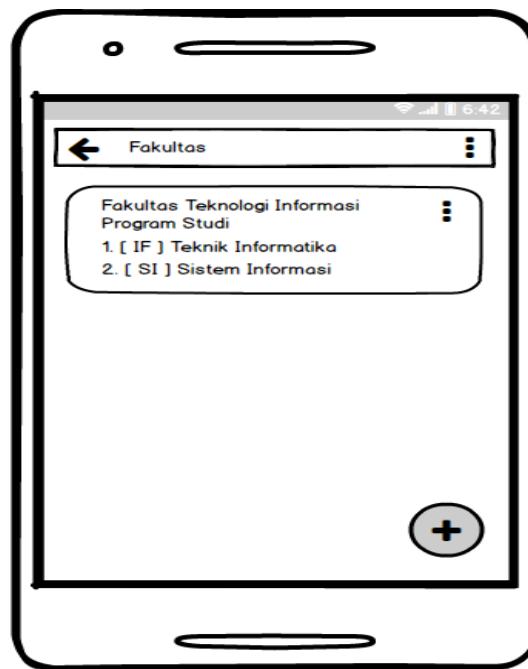
Gambar 4.90 Halaman Pelaporan Rekap Tracer Alumni

18. Perancangan Halaman Fakultas

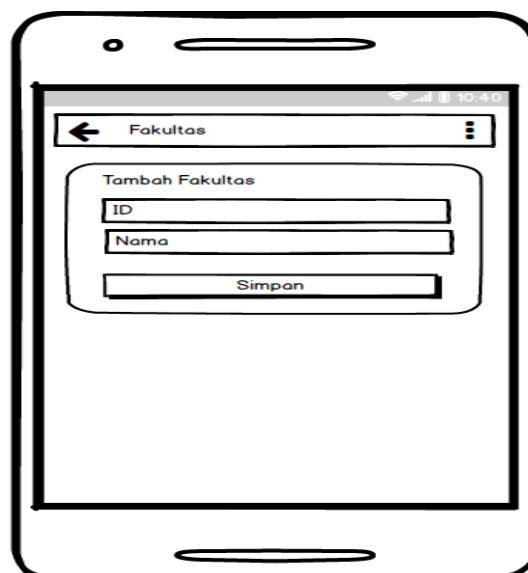
Pada perancangan Antarmuka halaman Pelaporan Fakultas yang dapat dilihat pada Gambar 4.91 Halaman Fakultas, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

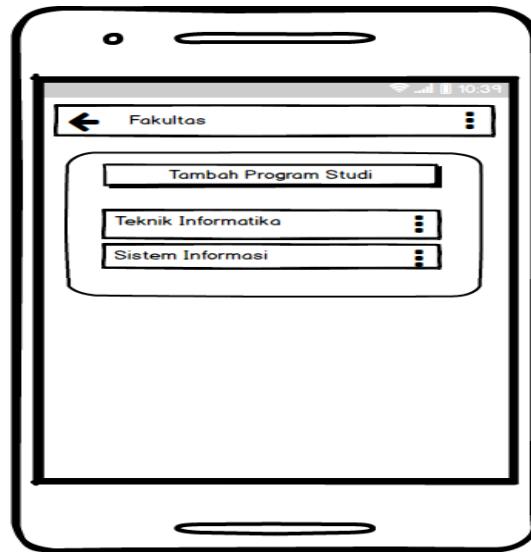
1. Panah Kembali, untuk kembali ke halaman sebelumnya
2. Cetak, untuk menyimpan rekap tracer alumni kedalam format pdf



Gambar 4.91 Halaman Fakultas



Gambar 4.92 Halaman Form Fakultas



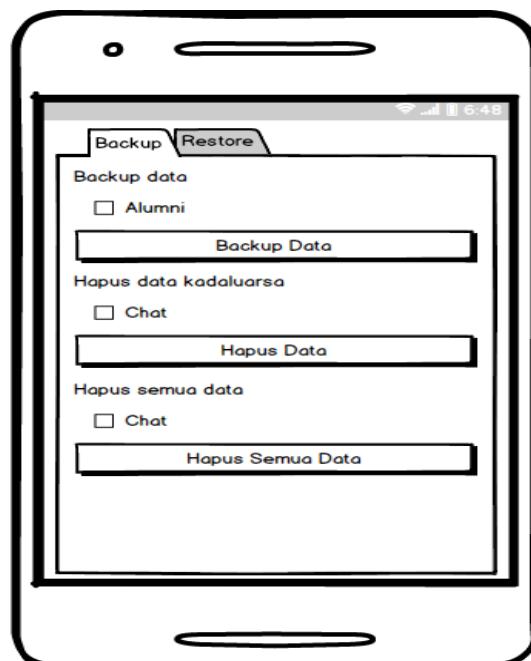
Gambar 4.93 Halaman Form Fakultas Program Studi

19. Perancangan Halaman Database

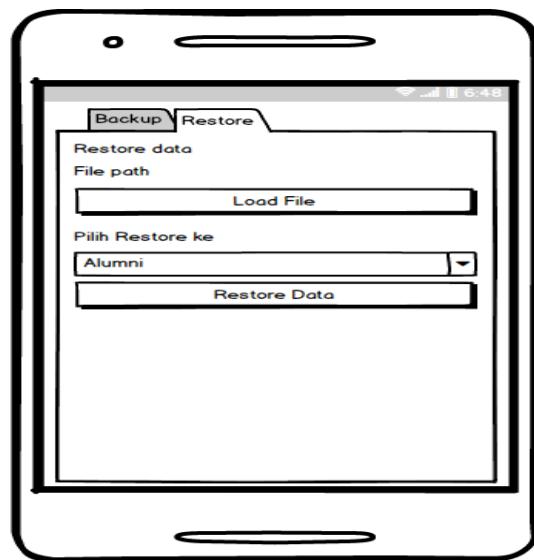
Pada perancangan Antarmuka halaman Pelaporan Database yang dapat dilihat pada Gambar 4.94 Halaman Database Backup, Gambar 4.95 Halaman Database Restore, dari aplikasi *Tracer Alumni* di Fakultas Teknologi Informasi Universitas Bale Bandung.

Keterangan fungsi setiap *button* pada halaman ini adalah:

1. Panah Kembali, untuk kembali ke halaman sebelumnya
2. Cetak, untuk menyimpan rekap tracer alumni kedalam format pdf



Gambar 4.94 Halaman *Database Backup*



Gambar 4.95 Halaman *Database Restore*

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Implementasi dilakukan di Fakultas Teknologi Informasi Universitas Bale Bandung bertempat di Jl.R.A.A wiranata kusumah No.7 Baleendah Bandung. terdiri dari lingkungan implementasi dan implementasi sistem adalah sebagai berikut:

5.1.1 Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*) yang digunakan pada saat pembuatan program yang dibutuhkan untuk mendukung jalannya aplikasi *tracer* alumni berbasis android adalah sebagai berikut:

1. Lingkungan Perangkat Keras (*Hardware*)

Spesifikasi hardware yang digunakan pada saat implementasi program ini adalah:

Tabel 5.1 Spesifikasi *Hardware*

| NO | Nama | Spesifikasi |
|----|--------------------------------------|---|
| 1 | Laptop HP | <ul style="list-style-type: none">1. Processor Intel® Core™ i5 @ 1.60GHz2. Ram 16GB3. Hardisk 1TB4. Monitor 17 inch |
| 2 | <i>Smartphone</i> Xiaomi Redmi 6A | <ul style="list-style-type: none">1. Jaringan (2G 3G 4G)2. Processor quadcore max 2.00Ghz3. Ram 2GB4. Internal storage 16GB5. Layar 5.45 inch |
| 3 | <i>Smartphone</i> Lenovo A2010 | <ul style="list-style-type: none">1. Jaringan (2G 3G 4G)2. Processor quadcore 1.00Ghz |

| | | |
|--|--|--|
| | | 3. <i>Ram</i> 1GB 4. <i>Internal storage</i> 8GB 5. Layar 4.5 inch |
|--|--|--|

2. Lingkungan Perangkat Lunak (Software)

Spesifikasi Software yang digunakan pada saat implementasi program ini adalah:

Tabel 5.2 Spesifikasi *Software*

| NO | Nama | Spesifikasi |
|----|--------------------------------------|---|
| 1 | Laptop HP | 1. Sistem Operasi Windows 10 dan Linux mint versi 19 2. Program aplikasi menggunakan Android Studio 3.1 3. Emulator menggunakan Genymotion 4. Database menggunakan Firebase Cloud Online 5. Browser menggunakan Mozilla Firefox |
| 2 | <i>Smartphone</i> Xiaomi Redmi 6A | Sistem operasi android 8.1 (Oreo) |
| 3 | <i>Smartphone</i> Lenovo A2010 | Sistem operasi android 5.1 (Lollipop) |

5.1.2 Implementasi Sistem

Implementasi sistem pada penelitian ini digunakan untuk keperluan aplikasi yang dibangun dapat di akses secara online maupun di download pada smartphone android meliputi Impementasi Kebutuhan *Firebase Database*, Impementasi Basis Data, Implementasi Antar Muka dan Implementasi Kebutuhan *Playstore* sebagai berikut:

1) Implementasi Kebutuhan *Firebase Database*

Pada Implementasi kebutuhan *Firebase Database* menjadi dasar untuk penyimpanan data pada aplikasi tracer alumni berbasis android. Spesifikasi

minimum untuk Syarat menjalankan *firebase database*, sistem operasi android API level 16 (*Jelly Bean*) dan menggunakan Gradle 4.1. Menambahkan *firebase* konfigurasi *service* kedalam project android beserta tampilan *database* pada *web browser* hanya admin yang bisa mengakses *firebase database* ini.

The screenshot shows the Firebase Project Overview page for a project named "Tracerstudy". Under the "Develop" section, the "Database" option is selected. On the right, under "Android apps", there is a card for "Tracer Study Alumni" with package name "com.unibba.traceralumni". It displays the following configuration details:

- Download the latest config file**: A button to download "google-services.json".
- App ID**: 1:435624143232:android:efca2020350a9dcb
- App nickname**: Tracer Study Alumni
- Package name**: com.unibba.traceralumni
- SHA certificate fingerprints**: 65:b7:bb:a7:de:e2:ad:16:33:3c:48:eb:41:f7:2c:48:cb:7c:01:16 (Type: SHA-1)
- Add fingerprint**: A link to add more fingerprints.
- Remove this app**: A button to remove the app configuration.

Gambar 5.1 *Firebase Konfiguasi Service*

The screenshot shows the Firebase Database interface for the same project. Under the "Database" tab, the "Data" tab is selected. The database structure is displayed as follows:

```

alumni
  +-- C1A150001
    +-- tracerstudy-skripsi2019
    |    +-- alumni
    |    |    +-- C1A150001
    |    |    +-- C1A150002
    |    |    +-- C1A150003
    |    |    +-- C1A150004
    |    |    +-- C1A150005
    |    |    +-- C1A150006
    |    |    +-- C1A150007
    |    |    +-- C1A150008
    |    |    +-- C1A150009
    |    +-- chat
    |    +-- fakultas
    |    +-- flp
    |    +-- tracerstudy
    |    +-- users
    |    +-- users_register
  +-- C1A150002
  +-- C1A150003
  +-- C1A150004
  +-- C1A150005
  +-- C1A150006
  +-- C1A150007
  +-- C1A150008
  +-- C1A150009

```

Gambar 5.2 *Firebase Database*

2) Implementasi Basis Data

Pada aplikasi *tracer* alumni FTI UNIBBA, menggunakan satu buah basis data yang bernama *tracerstudy_skripsi_2019* yang terdiri dari 7 buah tabel yaitu:

1. Tabel alumni

Pada file ini menerangkan tentang *field* mengenai data alumni dengan spesifikasi field adalah sebagai berikut:

Nama File : alumni

Media penyimpanan : *Database Cloud*

Implementasi dari Tabel alumni dalam format pohon nosql adalah sebagai berikut:

Tabel 5.3 Implementasi Tabel alumni

| |
|--------------------|
| alumni:collection |
| --id:document |
| --TS:int |
| --alamat:string |
| --alamatp:string |
| --angkatan:int |
| --email:string |
| --fakultas:string |
| --image_url:string |
| --latitude:string |
| --longitude:string |
| --nama:string |
| --namap:string |
| --nim:string |
| --nohp:string |
| --notlpp:string |
| --prodi:string |

2. Tabel chat

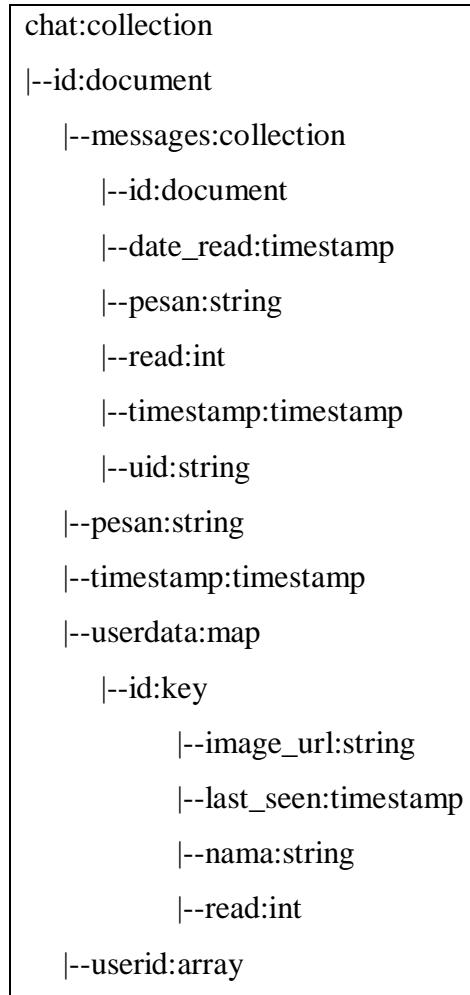
Pada file ini menerangkan tentang *field* mengenai data chatting dengan spesifikasi field adalah sebagai berikut:

Nama File : chat

Media penyimpanan : *Database Cloud*

Implementasi dari Tabel chat dalam format pohon nosql adalah sebagai berikut:

Tabel 5.4 Implementasi Tabel chat



3. Tabel fakultas

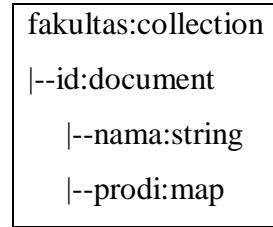
Pada file ini menerangkan tentang *field* mengenai data fakultas dan didalam fakultas memiliki data *collection* program studi dengan spesifikasi *field* adalah sebagai berikut:

Nama File : fakultas

Media penyimpanan : *Database Cloud*

Implementasi dari Tabel fakultas dalam format pohon nosql adalah sebagai berikut:

Tabel 5.5 Implementasi Tabel fakultas



4. Tabel flp

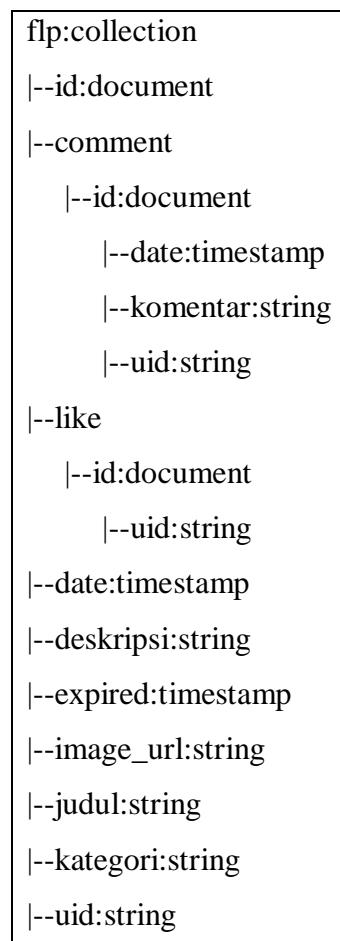
Pada *file* ini menerangkan tentang field mengenai data informasi forum, lowongan pekerjaan (loker) dan pengumuman dengan spesifikasi *fieldnya* adalah sebagai berikut:

Nama File : flp

Media penyimpanan : *Database Cloud*

Implementasi dari Tabel flp dalam format pohon nosql adalah sebagai berikut:

Tabel 5.6 Implementasi Tabel flp



| |
|-------------------|
| --write:timestamp |
|-------------------|

5. Tabel tracerstudy

Pada file ini menerangkan tentang field mengenai data kuesioner *tracer study* alumni dengan spesifikasi fieldnya adalah sebagai berikut:

Nama File : tracerstudy

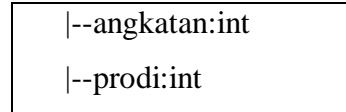
Media penyimpanan : *Database Cloud*

Implementasi dari Tabel tracerstudy dalam format pohon nosql adalah sebagai berikut:

Tabel 5.7 Implementasi Tabel tracerstudy

| |
|------------------------|
| tracerstudy:collection |
| --id:document |
| --F11_1:int |
| --F11_2:int |
| --F11_3:int |
| --F11_4:int |
| --F11_5:int |
| --F11_5_other:string |
| --F11_6:string |
| --F11_7:string |
| --F11_8:string |
| --F11_9:string |
| --F17_1:int |
| --F17_2:int |
| --F2_1:int |
| --F2_10:int |
| --F2_11:int |
| --F2_12:int |
| --F2_13:int |
| --F2_14:int |
| --F2_2:int |
| --F2_3:int |

| |
|-------------------|
| --F2_4:int |
| --F2_5:int |
| --F2_6:int |
| --F2_7:int |
| --F2_8:int |
| --F2_9:int |
| --F3:int |
| --F4_1:int |
| --F4_10:int |
| --F4_11:int |
| --F4_12:int |
| --F4_13:int |
| --F4_14:int |
| --F4_15:int |
| --F4_15_other:int |
| --F4_2:int |
| --F4_3:int |
| --F4_4:int |
| --F4_5:int |
| --F4_6:int |
| --F4_7:int |
| --F4_8:int |
| --F4_9:int |
| --F5:int |
| --F6:string |
| --F7:string |
| --F7_a:string |
| --F8:int |
| --TS_1:int |
| --TS_2:int |
| --TS_3:int |
| --TS_4:int |



6. Tabel users

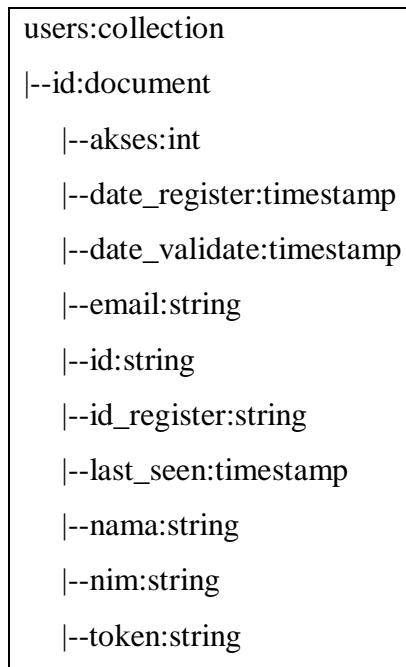
Pada file ini menerangkan tentang field mengenai data user login dengan spesifikasi fieldnya adalah sebagai berikut:

Nama File : users

Media penyimpanan : *Database Cloud*

Implementasi dari Tabel users dalam format pohon nosql adalah sebagai berikut:

Tabel 5.8 Implementasi Tabel users



7. Tabel users_registrasi

Pada file ini menerangkan tentang field mengenai data registrasi user dengan spesifikasi fieldnya adalah sebagai berikut:

Nama File : users_registrasi

Media penyimpanan : *Database Cloud*

Implementasi dari Tabel user_register dalam format pohon nosql adalah sebagai berikut:

Tabel 5.9 Implementasi Tabel users_register

```

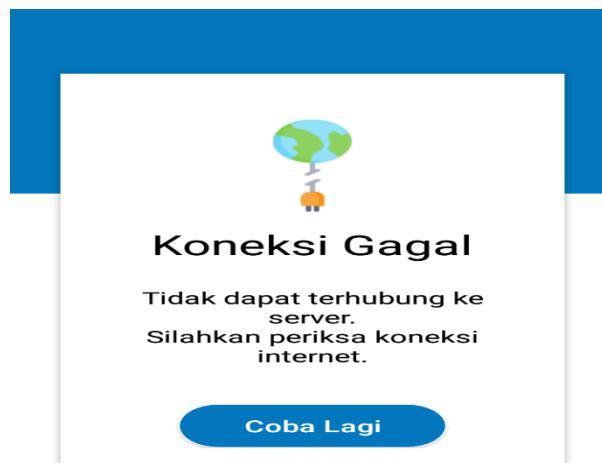
users_register:collection
|--id:document
|--email:string
|--nama:string
|--nim:string
|--password:string
|--tanggal:timestamp
|--validasi:int

```

3) Implementasi Antar Muka

1. Halaman Gagal Koneksi Internet

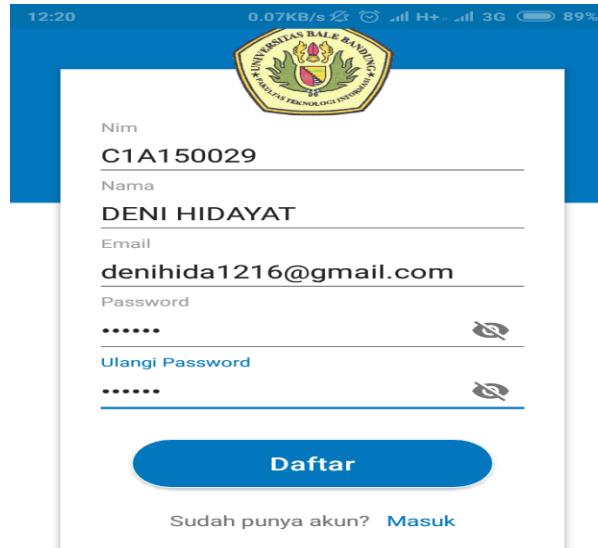
Halaman Gagal Koneksi Internet merupakan halaman ketika aplikasi tidak terkoneksi ke internet / tidak ada jaringan maka akan menampilkan halaman ini.



Gambar 5.3 Halaman Gagal Koneksi Internet

2. Halaman Registrasi

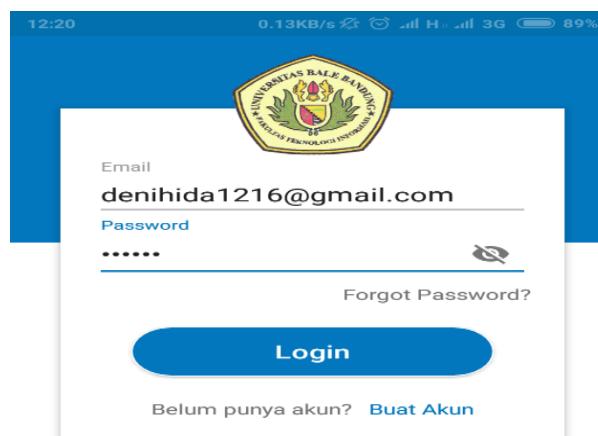
Halaman Registrasi merupakan halaman digunakan untuk mendaftar *tracer study* alumni jika alumni belum mempunyai akun untuk masuk ke aplikasi.



Gambar 5.4 Halaman Registrasi Aplikasi

3. Halaman *Login*

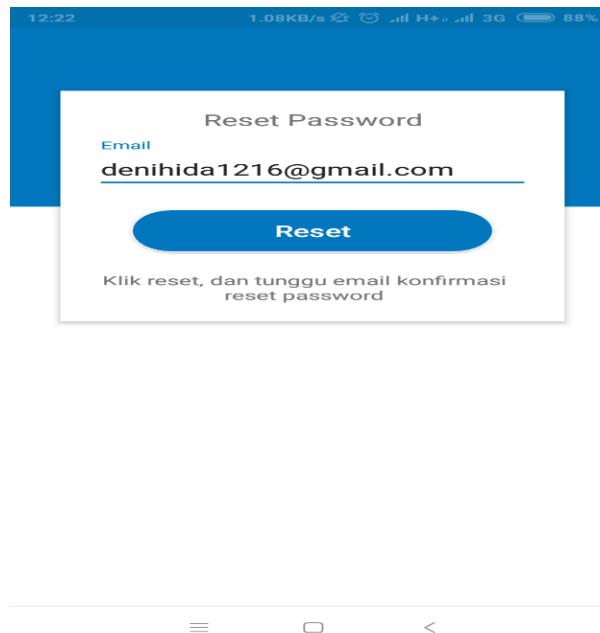
Halaman *Login* merupakan halaman awal sebelum masuk ke sistem harus mengisi email dan *password* yang sudah terdaftar.



Gambar 5.5 Halaman *Login* Aplikasi

4. Halaman Forgot Password

Halaman Forgot Password merupakan halaman digunakan untuk reset password dengan *email* yang sudah terdaftar di sistem *tracer study* alumni.



Gambar 5.6 Halaman *Forgot Password* Aplikasi

5. Halaman Menunggu Validasi

Halaman Menunggu Validasi merupakan halaman yang digunakan jika user sudah melakukan registrasi dan menunggu divalidasi oleh admin.



Gambar 5.7 Halaman Menunggu Validasi

6. Halaman Gagal Validasi

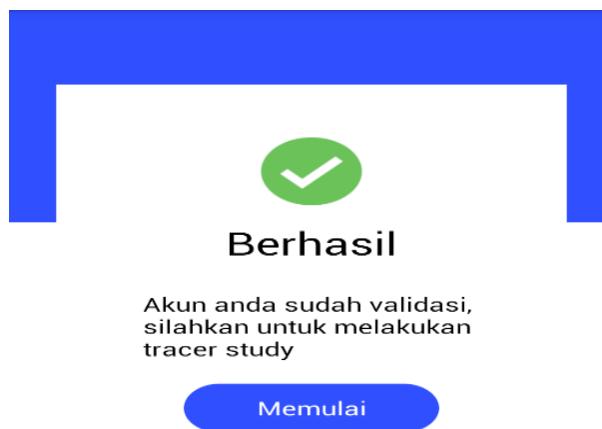
Halaman Gagal Validasi merupakan halaman yang digunakan jika user sudah melakukan registrasi tapi ditolak oleh admin.



Gambar 5.8 Halaman Gagal Validasi

7. Halaman Berhasil Validasi

Halaman Berhasil Validasi merupakan halaman yang digunakan jika user sudah melakukan registrasi dan diverifikasi oleh admin, sehingga bisa memulai pengisian kuesioner *tracer* alumni.



Gambar 5.9 Halaman Berhasil Validasi Aplikasi

8. Halaman Kuesioner

Halaman Kuesioner merupakan halaman yang digunakan untuk mengisi dan menjawab pertanyaan-pertanyaan yang sudah disediakan di sistem.

INFORMASI PENGISIAN DATA TRACER STUDY ALUMNI

- 1 Data Pribadi** ✓
- 2 Pengalaman Metode Penekanan Pembelajaran** ✓
- 3 Kepuasan Pengguna Lulusan** ✓
- 4 Riwayat Pekerjaan** ✓

Catatan:
*Pada Tahap ke 4 bagi yang sudah bekerja

Simpan

Gambar 5.10 Halaman Kuesioner Menu Aplikasi

Data Pribadi

Nim
C1A150029

Nama
DENI HIDAYAT

Email
denihida1216@gmail.com

No Telepon / HP
085721661555

Program Studi
Teknik Informatika ▾
Angkatan
2015

Alamat
Sukamanah No.23 Majalaya Bandung

Simpan

Gambar 5.11 Halaman Kuesioner Data Pribadi Aplikasi

Metode Pembelajaran

Menurut anda seberapa besar penekanan pada metode pembelajaran di bawah ini dilaksanakan di program studi anda?

* Wajib diisi

Magang *

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|
| Tidak Sama sekali | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Besar |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|

Praktikum *

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|
| Tidak Sama sekali | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Besar |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|

Perkuliahahan *

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|
| Tidak Sama sekali | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Besar |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|

Partisipasi dalam proyek riset *

| | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|
| Tidak Sama sekali | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Besar |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|--------------|

Simpan

≡ □ <

Gambar 5.12 Halaman Kuesioner Pengalaman Pembelajaran Aplikasi

Kepuasan Pengguna Lulusan

Etika *

| | | | | | |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|
| Kurang | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Baik |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|

Keahlian pada bidang ilmu (kompetensi utama) *

| | | | | | |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|
| Kurang | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Baik |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|

Kemampuan berbahasa asing *

| | | | | | |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|
| Kurang | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Baik |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|

Penggunaan Teknologi Informasi *

| | | | | | |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|
| Kurang | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Baik |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|

Kemampuan berkomunikasi *

| | | | | | |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|
| Kurang | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Baik |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|

Kerjasama *

| | | | | | |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|
| Kurang | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Sangat Baik |
|--------|-----------------------|-----------------------|-----------------------|----------------------------------|-------------|

≡ □ <

Gambar 5.13 Halaman Kepuasan Pengguna Lulusan Aplikasi

Pekerjaan

* Wajib diisi

F3: Kapan anda mulai mencari pekerjaan? Mohon pekerjaan sambilan tidak dimasukkan *

Sebelum lulus S/D. 2 bulan setelah Lulus
 3 - 8 bulan setelah lulus
 Diatas 8 bulan setelah lulus
 Saya tidak mencari kerja (langsung ke F8)

F8: Apakah anda bekerja saat ini (termasuk kerja sambilan dan wirausaha)? *

Ya (Jika ya, lanjutkan ke F11)
 Tidak

F11: Apa jenis perusahaan/instansi/institusi tempat anda bekerja sekarang?

Instansi pemerintah (termasuk BUMN)
 Organisasi non-profit/Lembaga Swadaya Masyarakat
 Perusahaan swasta
 Wirausaha/perusahaan sendiri
 Other

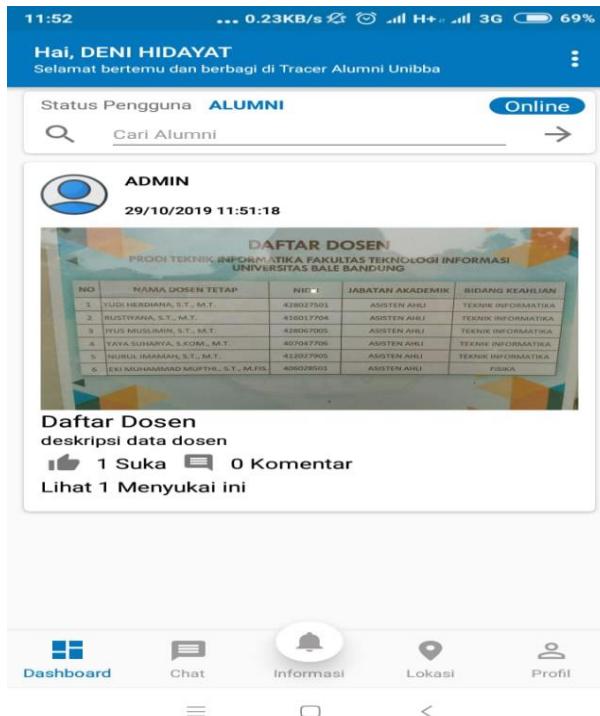
Gambar 5.14 Halaman Riwayat Pekerjaan Aplikasi

9. Halaman *Dashboard*

Halaman *Dashboard* merupakan halaman utama saat masuk kedalam sistem menampilkan jumlah responden, informasi hasil *tracer* alumni dalam bentuk grafik dan pengumuman terbaru.



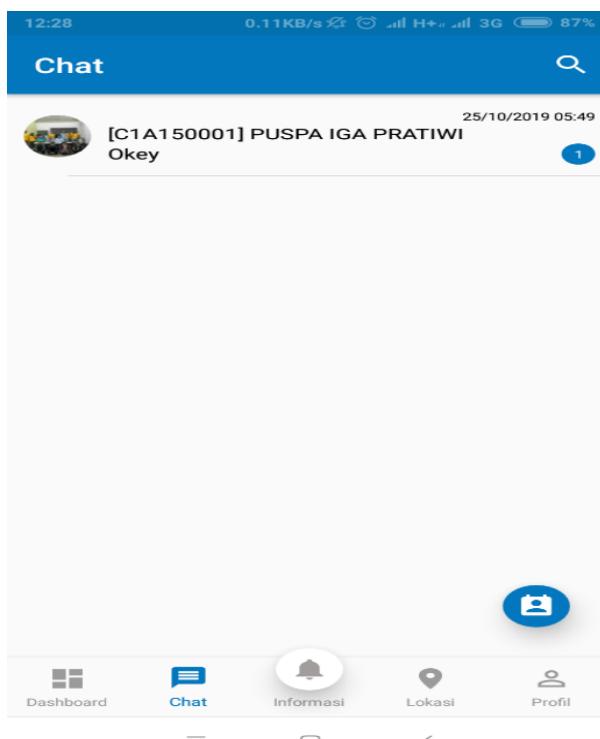
Gambar 5.15 Halaman Dashboard Admin Aplikasi



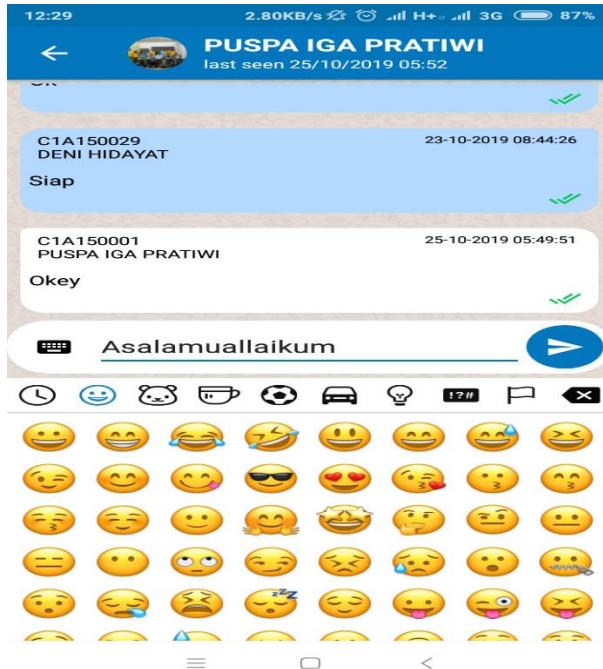
Gambar 5.16 Halaman Dashboard User Aplikasi

10. Halaman Chat

Halaman Chat merupakan halaman komunikasi chatting sesama pengguna aplikasi tracer alumni.



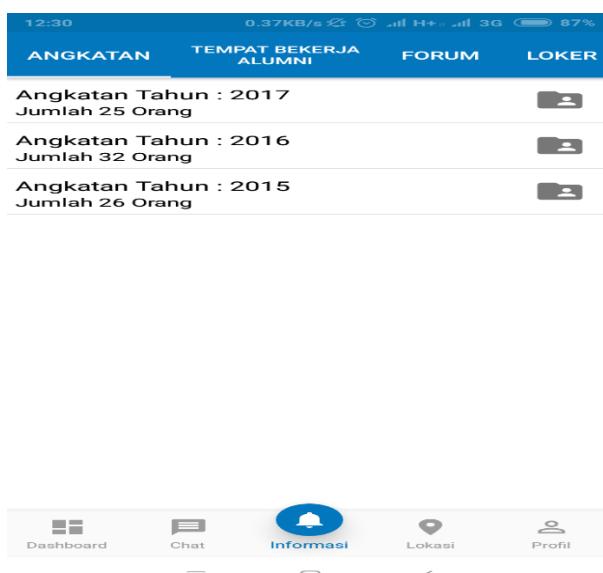
Gambar 5.17 Halaman Chat List Aplikasi



Gambar 5.18 Halaman Chat Kirim Aplikasi

11. Halaman Informasi

Halaman Informasi merupakan *sub menu* dari halaman berbagi informasi seputar forum alumni, halaman berbagi informasi seputar lowongan pekerjaan untuk alumni yang belum bekerja dan yang sudah bekerja bisa berbagi informasi lowongan bekerja, sangat membantu bagi alumni yang belum bekerja, halaman daftar nama alumni, pencarian nama berdasarkan nim, nama, tahun, angkatan dan program studi.



Gambar 5.19 Halaman Informasi Angkatan Aplikasi



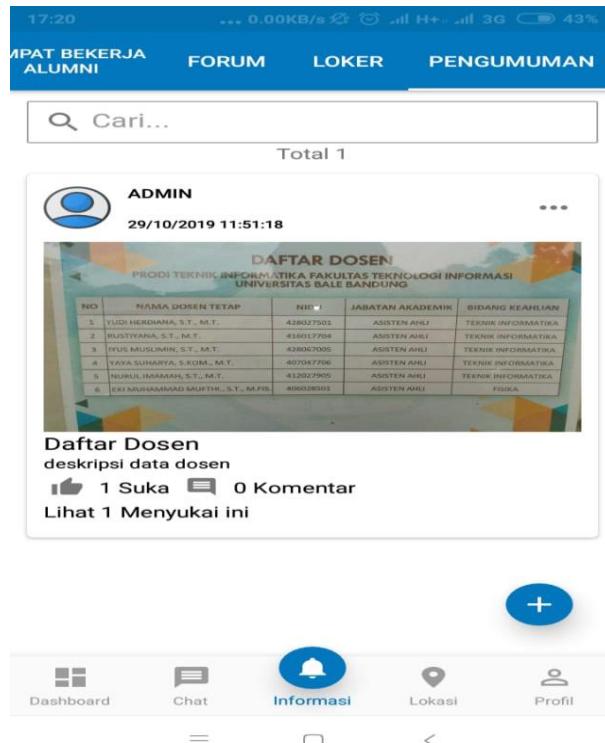
Gambar 5.20 Halaman Informasi Tempat Bekerja Alumni Aplikasi



Gambar 5.21 Halaman Informasi Forum Aplikasi



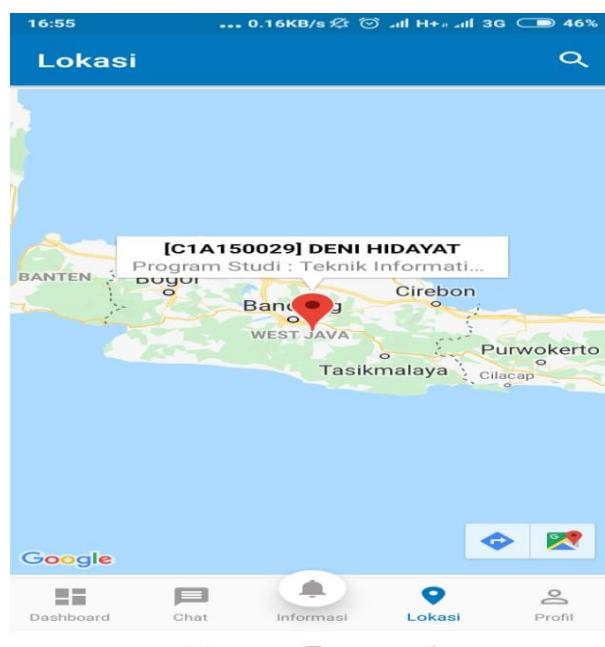
Gambar 5.22 Halaman Informasi Loker Aplikasi



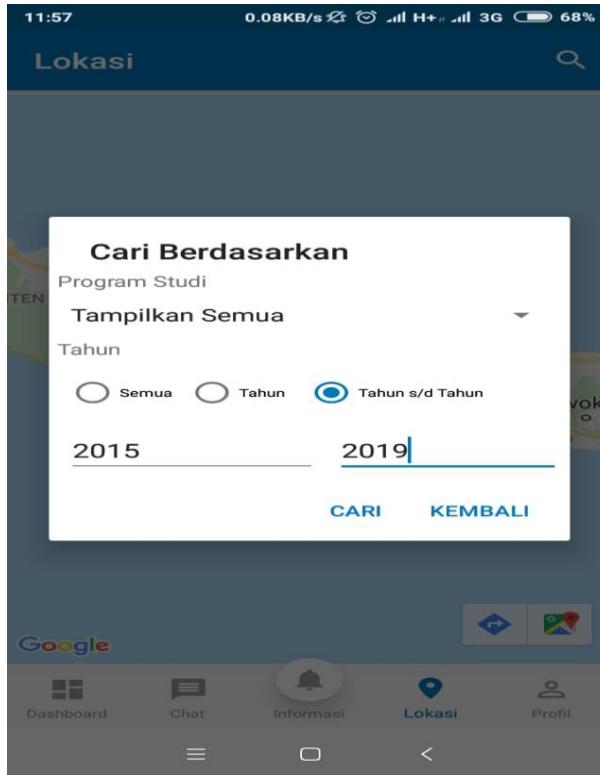
Gambar 5.23 Halaman Informasi Pengumuman Aplikasi

12. Halaman Lokasi

Halaman Lokasi merupakan halaman posisi lokasi keberadaan alumni yang tersedia pada aplikasi dengan menampilkan posisi map beserta koordinat lokasi alumni.



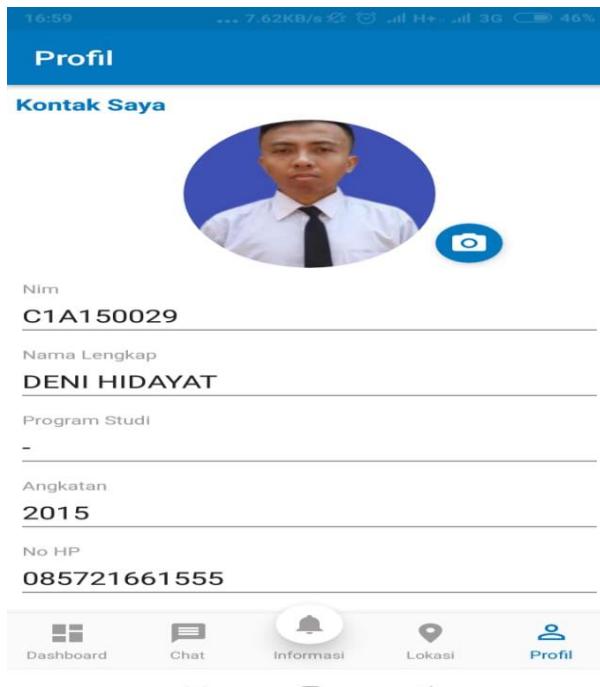
Gambar 5.24 Halaman Lokasi Aplikasi



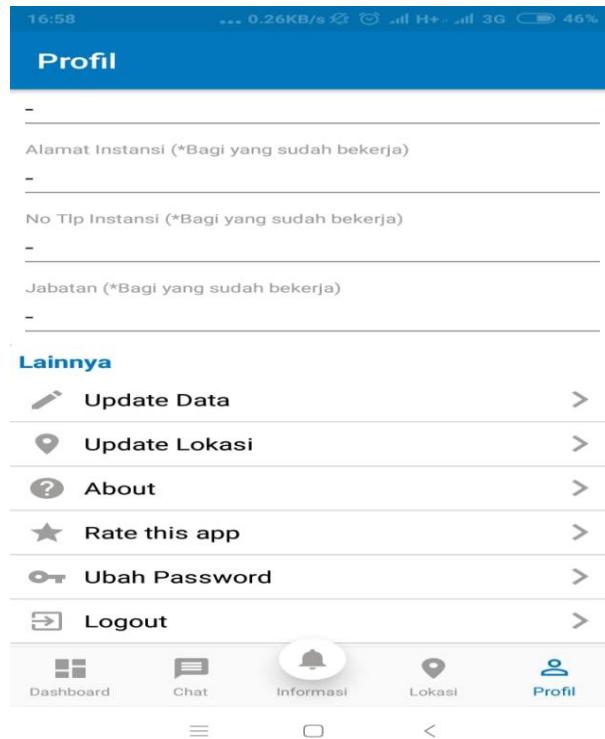
Gambar 5.25 Halaman Lokasi Pencarian Aplikasi

13. Halaman Profil

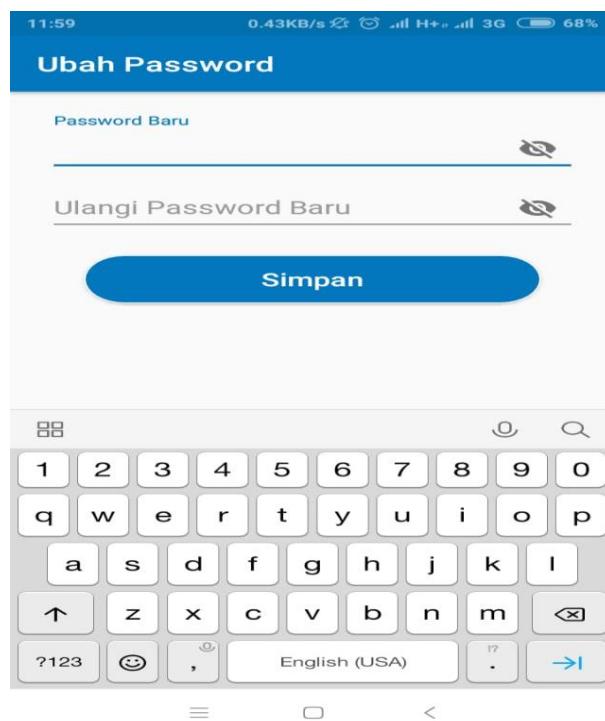
Halaman Profil merupakan halaman berisi informasi profil user beserta photo Profil dan user bisa dengan mudah melakukan *update* data.



Gambar 5.26 Halaman Profil Aplikasi



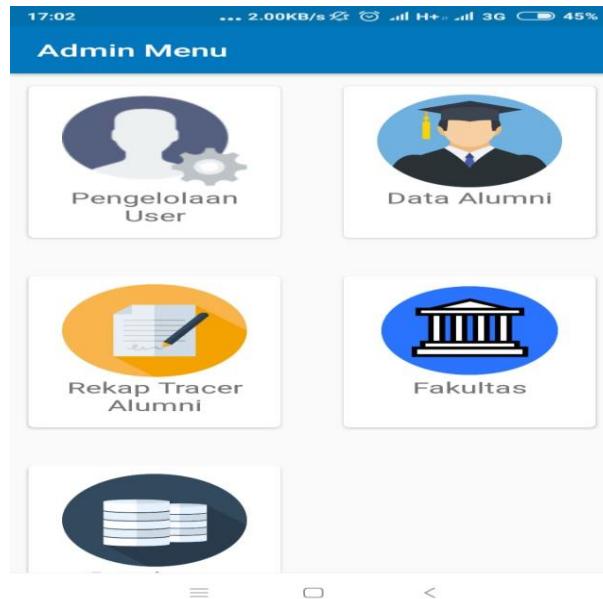
Gambar 5.27 Halaman Profil Item Menu Aplikasi



Gambar 5.28 Halaman Ubah Password Aplikasi

14. Halaman *Administrator*

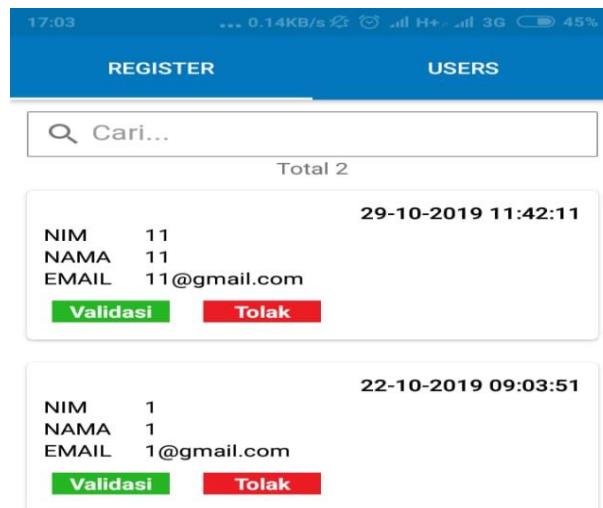
Halaman Administrator merupakan halaman berisi menu pengelolaan *user*, data alumni, rekap *tracer* alumni, fakultas dan *database*.



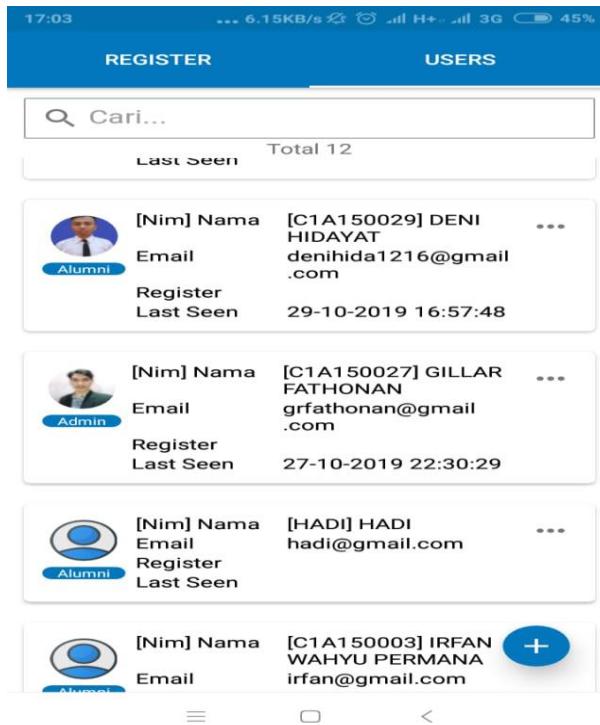
Gambar 5.29 Halaman *Administrator* Aplikasi

15. Halaman Pengelolaan *User*

Halaman Pengelolaan *User* merupakan dengan sub menu halaman pengelolaan user registrasi dan login, admin dapat menambah, ubah, delete dan validasi user.



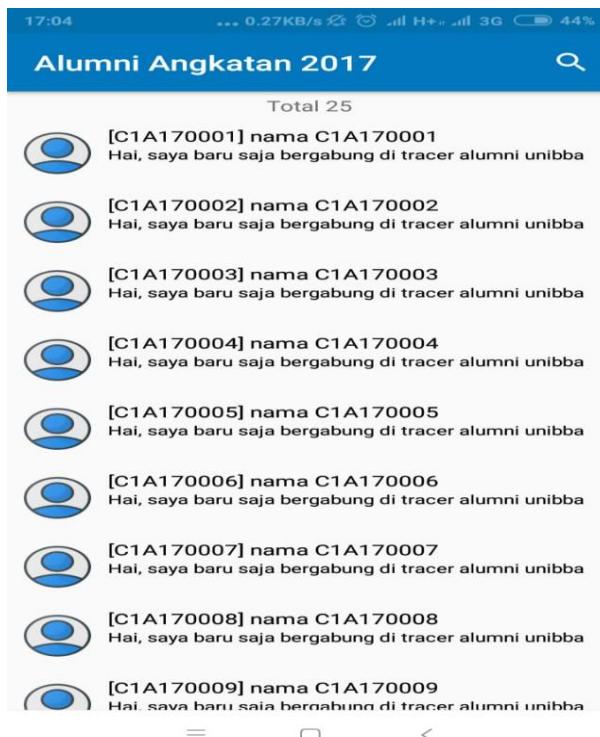
Gambar 5.30 Halaman Pengelolaan *User List* Registrasi Aplikasi



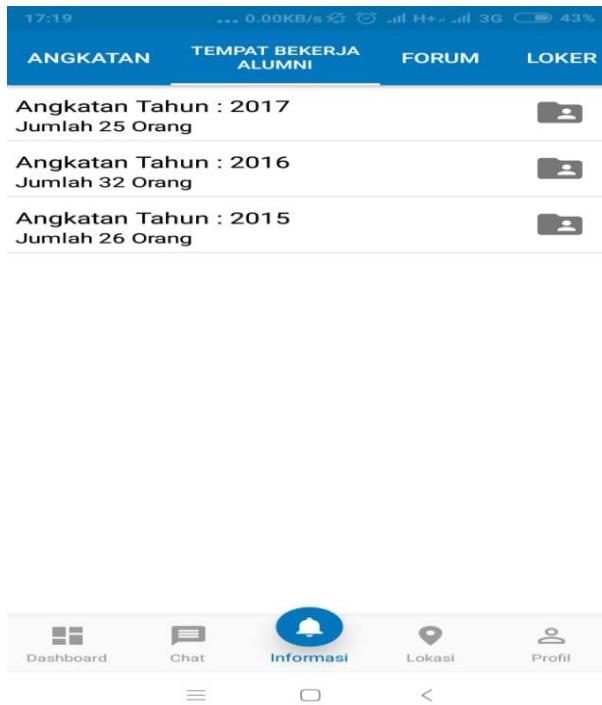
Gambar 5.31 Halaman Pengelolaan *User List Users* Aplikasi

16. Halaman Data Alumni

Halaman Pelaporan Data Alumni merupakan halaman data Alumni, admin dapat melakukan pencarian dan cetak ke format pdf.



Gambar 5.32 Halaman Data Alumni Angkatan Aplikasi



Gambar 5.33 Halaman Data Alumni Tempat Bekerja Aplikasi

17. Halaman Rekap *Tracer Alumni*

Halaman Pelaporan Rekap *Tracer Alumni* merupakan halaman Rekap *Tracer Alumni*, admin dapat melakukan pencarian dan cetak ke format pdf.

| Keterangan tentang lulusan | Jumlah | Percentase |
|---|--------|------------|
| Jumlah Lulusan Tahun 2017 (Seluruh Prodi S1) | 25 | 30.12% |
| Jumlah Responden Tahun 2017 (Jumlah lulusan yang terkontak/yang menjawab survei) | 25 | 30.12% |
| Jumlah Lulusan Tahun 2016 (Seluruh Prodi S1) | 32 | 38.55% |
| Jumlah Responden Tahun 2016 (Jumlah lulusan yang terkontak/yang menjawab survei) | 32 | 38.55% |
| Jumlah Lulusan Tahun 2015 (S1 [C1A150001] C1A150001 Program studi [C1A150001] C1A150001) | 26 | 31.33% |
| Jumlah Responden Tahun 2015 (Jumlah lulusan yang terkontak/yang menjawab survei) | 26 | 31.33% |

Gambar 5.34 Halaman Rekap *Tracer Alumni* Aplikasi

The screenshot shows a mobile application interface titled 'Rekap Data Tracer'. At the top, there is a header with the date '17:03', signal strength, battery level at 45%, and a 'Masukan Tahun' field containing '2019'. To the right of this field are three export options: 'Export Excel', 'Export Csv', and 'Export Pdf'. Below the header, the title 'Pedoman Kementerian tentang Pelaksanaan Tracer Study di Perguruan Tinggi' is displayed. A section titled 'Informasi Tentang Responden:' contains a table with the following data:

| Keterangan tentang lulusan | Jumlah | Persentase |
|--|--------|------------|
| Jumlah Lulusan Tahun 2017 (Seluruh Prodi S1) | 25 | 30.12% |
| Jumlah Responden Tahun 2017 (Jumlah lulusan yang terkontak/yang menjawab survei) | 25 | 30.12% |
| Jumlah Lulusan Tahun 2016 (Seluruh Prodi S1) | 32 | 38.55% |
| Jumlah Responden Tahun 2016 (Jumlah lulusan yang terkontak/yang menjawab survei) | 32 | 38.55% |
| Jumlah Lulusan Tahun 2015 (S1 [C1A150001] C1A150001 Program studi [C1A150001] C1A150001) | 26 | 31.33% |
| Jumlah Responden Tahun 2015 | 26 | 31.33% |

Gambar 5.35 Halaman Rekap *Tracer Alumni Export* Aplikasi

18. Halaman Fakultas

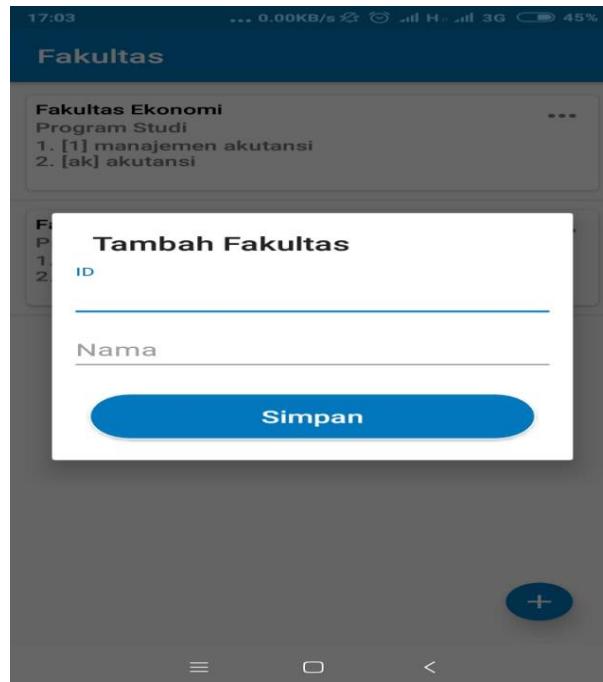
Halaman Fakultas merupakan halaman untuk mengelola fakultas beserta program studi.

The screenshot shows a mobile application interface titled 'Fakultas'. At the top, there is a header with the date '12:34', signal strength, battery level at 87%, and a title 'Fakultas'. Below the header, there are two cards representing faculties:

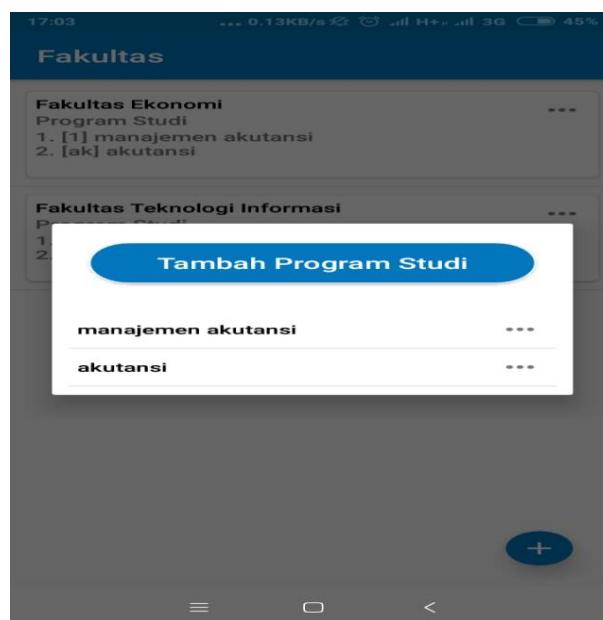
- Fakultas Ekonomi**
 - Program Studi
 - 1. [1] manajemen akutansi
 - 2. [ak] akutansi
- Fakultas Teknologi Informasi**
 - Program Studi
 - 1. [SI] Sistem Informasi
 - 2. [IF] Teknik Informatika

A blue circular button with a '+' sign is located at the bottom center of the screen.

Gambar 5.36 Halaman Fakultas Aplikasi



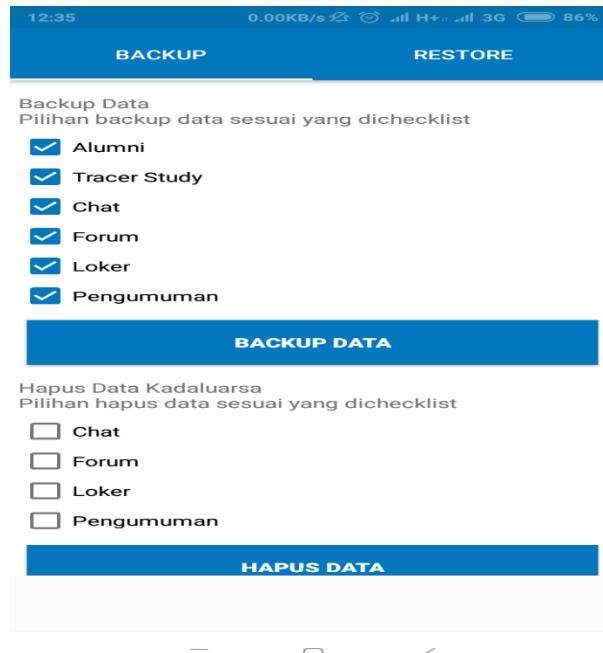
Gambar 5.37 Halaman Fakultas Aksi Tambah Aplikasi



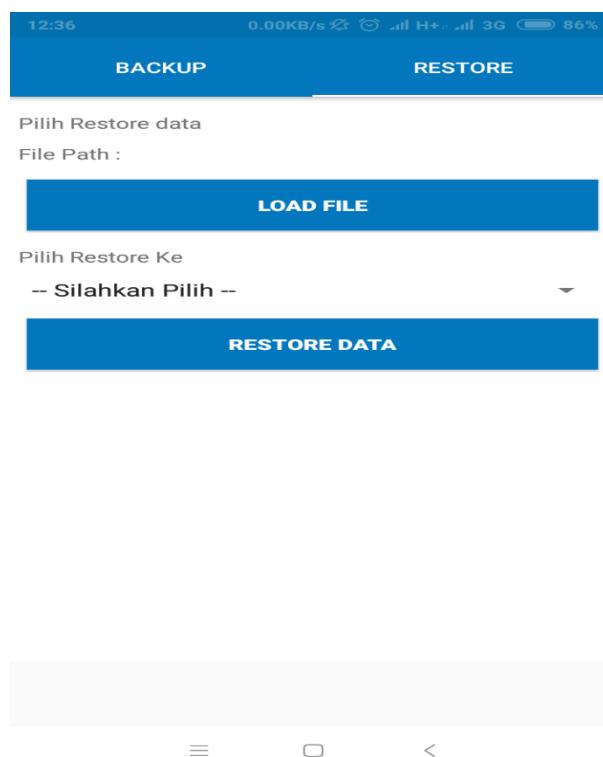
Gambar 5.38 Halaman Fakultas Aksi Tambah Program Studi Aplikasi

19. Halaman *Database*

Halaman *Database* merupakan untuk *backup* dan *restore* data *format file* yang digunakan adalah .json sehingga admin dapat melakukan aksi *backup* data *export* file .json sdcard *smartphone*, *delete* data tabel sesuai dengan yang dipilih dan melakukan *restore* data *import* file .json kirim data ke *database*.



Gambar 5.39 *Database Backup Aplikasi*



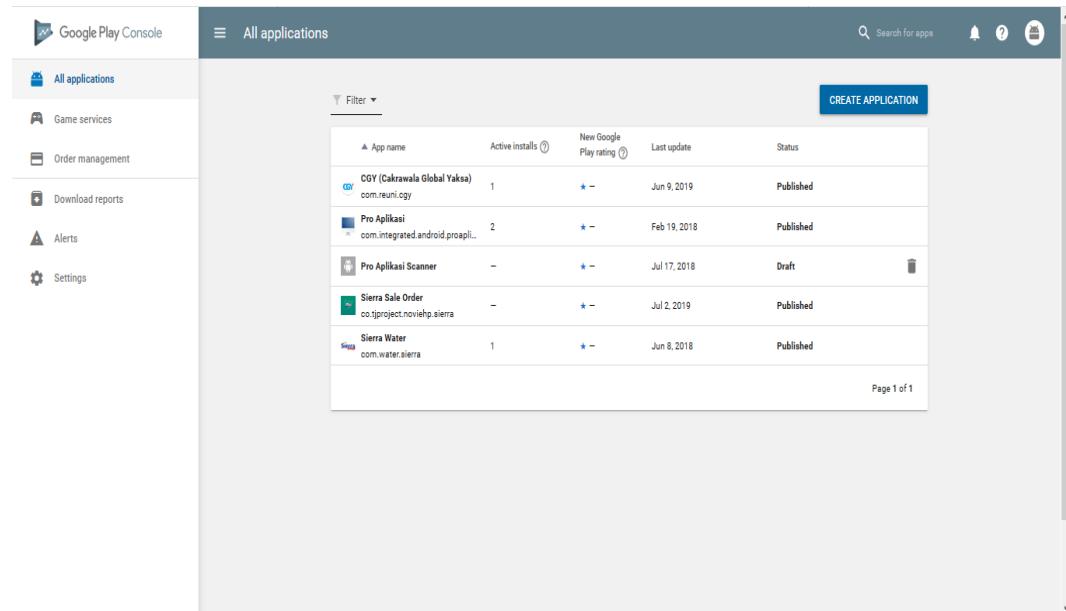
Gambar 5.40 *Database Restore Aplikasi*

4) Implementasi Kebutuhan *Playstore*

Pada Implementasi kebutuhan *playstore* setelah proses *package project* yang menghasilkan sebuah file *APK*, langkah selanjutnya adalah mengunggah /

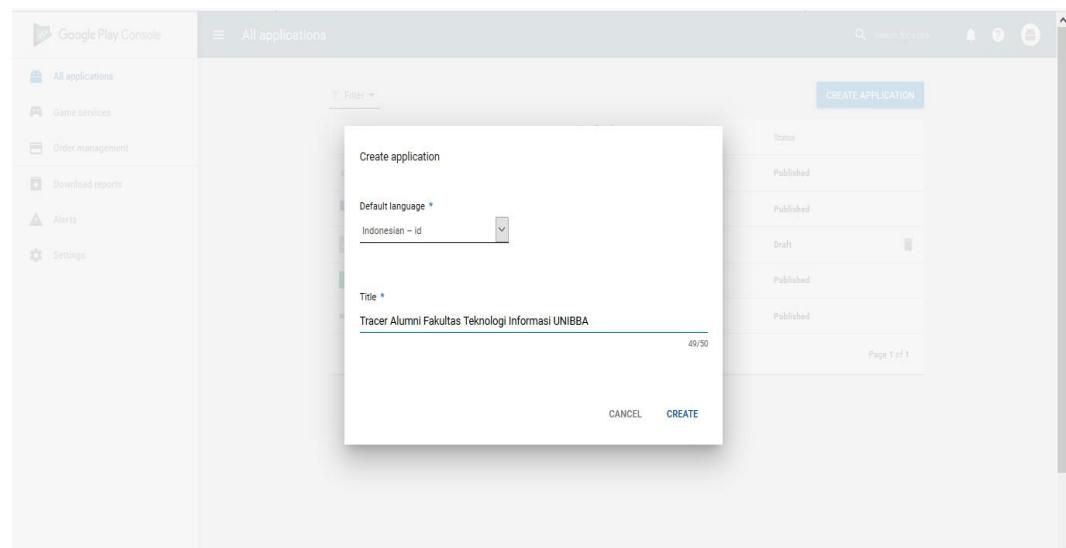
upload file *APK* tersebut ke server *Playstore*. Pada bab ini tidak akan dibahas tentang bagaimana membuat akun Android *developer*. penulis asumsikan sudah memiliki akun Android *Developer*. Proses ini menggunakan sebuah browser, langkahnya adalah sebagai berikut:

1. Buka browser, kemudian ketikan alamat web <https://play.google.com/apps/publish>



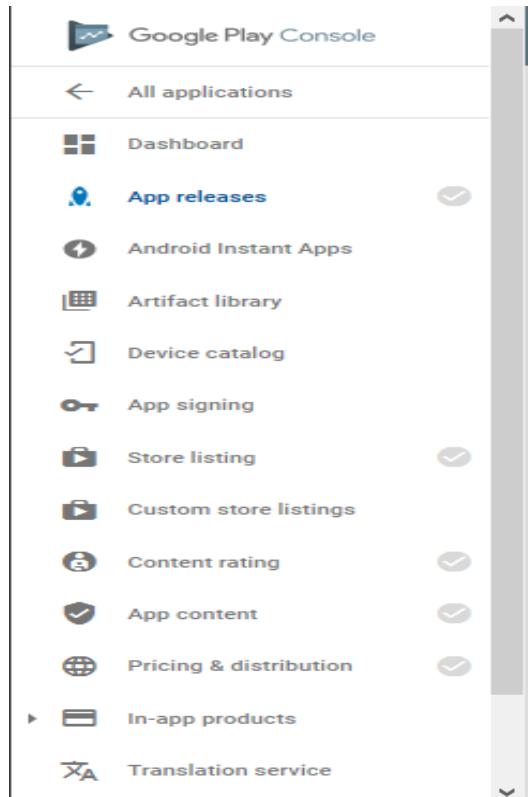
Gambar 5.41 Halaman *Google Play Console Dashboard*

2. Pada bagian halaman dashboard google *developer* klik tombol *create application*, isi *default language* dan *title* kemudian klik tombol *create*



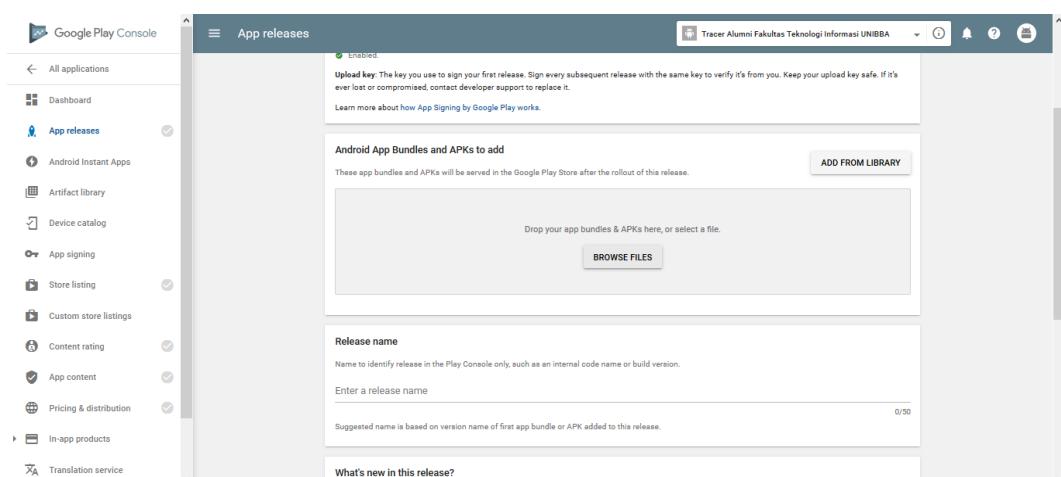
Gambar 5.42 Halaman *Google Play Console Create Application*

3. Pada Bagian ini data yang harus diisi untuk tahapan upload apk ke google playstore yaitu *app releases, store listing, content rating, app content* dan *Pricing & distribution*.



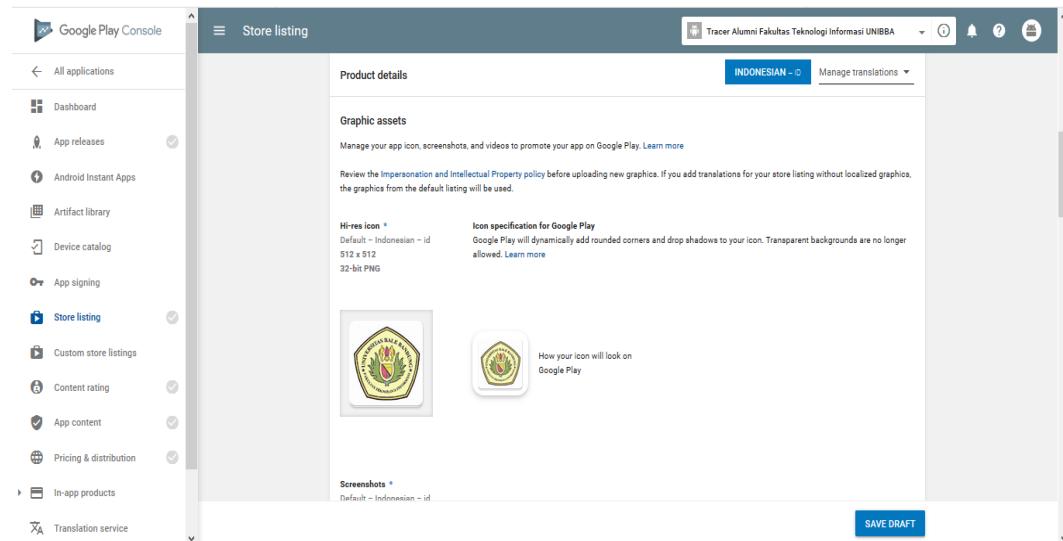
Gambar 5.43 Halaman *Google Play Console Panel*

4. Pada bagian halaman *App Releases* ini untuk mengelola rilis *developer apk* yang berisi inputan dengan memilih *production create releases, upload file android apk*, rilis nama aplikasi, deskripsi rilis aplikasi dan klik tombol *review*.



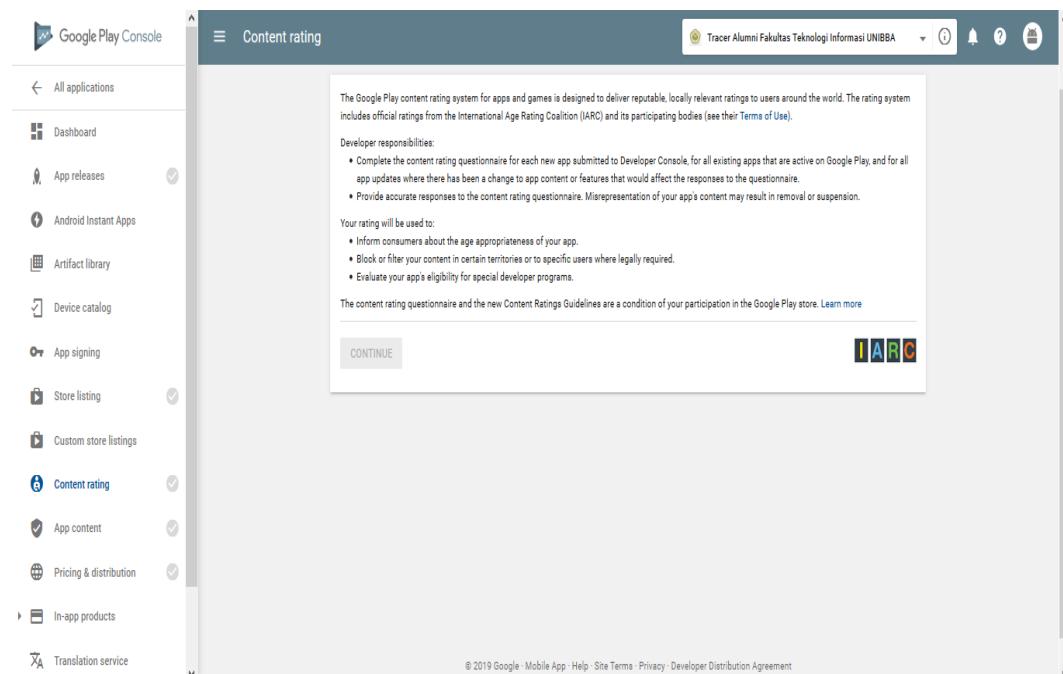
Gambar 5.44 Halaman *Google Play Console App Releases*

5. Pada bagian halaman *Store Listing* ini untuk keterangan daftar toko aplikasi yang berisi inputan Nama Title Aplikasi, Deskripsi Singkat Aplikasi, Deskripsi Lengkap Aplikasi, Asset Grafis dengan gambar ukuran 512x512 32 bit png, Fitur Grafik dengan gambar ukuran 1024x500, Tipe Aplikasi, Kategori Aplikasi, Website, Email, Telepon, *Privacy Police* dan klik tombol *save*.



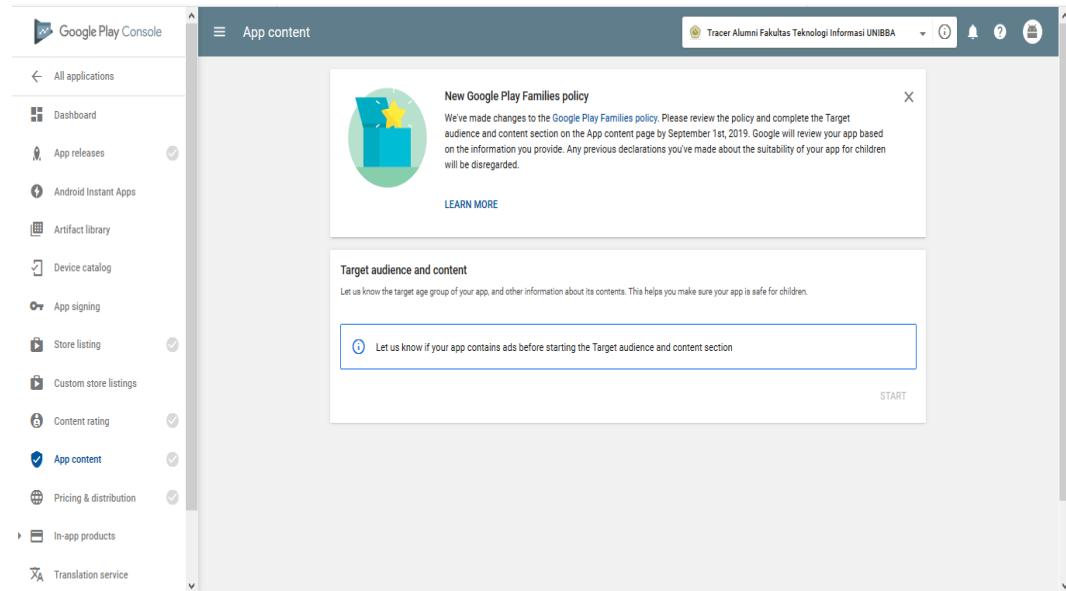
Gambar 5.45 Halaman Google Play Console *Store Listing*

6. Pada bagian halaman *Content Rating* ini terdapat isi rating konten aplikasi.



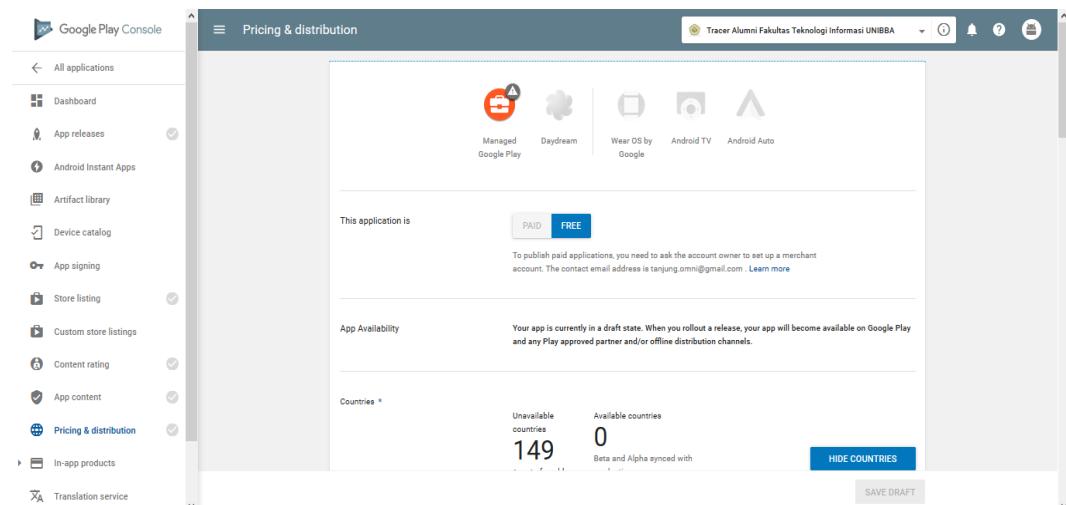
Gambar 5.46 Halaman Google Play Console *Content Rating*

- Pada bagian halaman ini konten aplikasi yang diunggah masuk harus memilih dalam kategori usia dari 13 tahun sampai 15 tahun, 16 tahun sampai 17 tahun dan 18 tahun keatas.



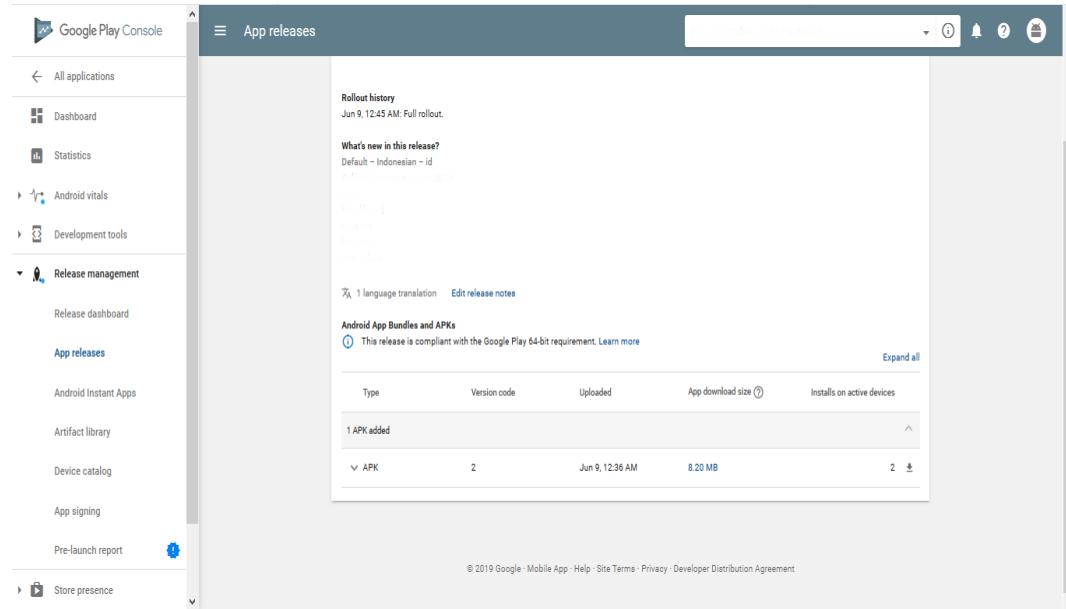
Gambar 5.47 Halaman *Google Play Console App Content*

- Pada bagian halaman ini terdapat penentuan harga aplikasi dan pengaturan negara yang dapat melihat atau menginstal aplikasi *tracer* alumni.



Gambar 5.48 Halaman *Google Play Console Pricing & Distribution*

- Pada Tahap terakhir Publikasikan Aplikasi ke *Google Playstore* Setelah status aplikasi menjadi siap dipublikasikan, maka langkah selanjutnya adalah mempublikasikan aplikasi tersebut ke google play dengan cara Pindah ke tab Kelola Rilis dan Klik tombol Lanjutkan.

Gambar 5.49 Halaman *Google Play Publish*

5.2 Pengujian

Pengujian yang akan dilakukan adalah pengujian dengan metode *black box* untuk setiap proses yang ada didalam aplikasi. Mulai dari saat pengguna membuka hingga menutup aplikasi.

5.2.1 Rencana Pengujian

Rencana pengujian pada aplikasi dibagi dua terdiri dari rencana pengujian fungsional sistem dan rencana pengujian interface sistem adalah sebagai berikut:

Tabel 5.10 Rencana Pengujian Fungsionalitas Sistem

| Rencana Pengujian Fungsionalitas Sistem | | | |
|--|-----------------------|--|------------------|
| No | Item Uji | Detail Uji | Jenis Uji |
| 1 | Registrasi | Validasi registrasi | <i>Black Box</i> |
| | | Verifikasi cek data ke <i>database</i> | <i>Black Box</i> |
| | | Menyimpan data ke <i>database</i> | <i>Black Box</i> |
| 2 | <i>Login</i> | Validasi <i>login</i> | <i>Black Box</i> |
| | | Verifikasi cek data ke <i>database</i> | <i>Black Box</i> |
| 3 | <i>Reset Password</i> | Validasi <i>email</i> | <i>Black Box</i> |
| | | Verifikasi <i>email</i> | <i>Black Box</i> |

| | | | |
|----|---------------------------------|---|------------------|
| | | Mengirim link <i>reset password</i> melalui email | <i>Black Box</i> |
| 4 | Kuesioner | Menampilkan data | <i>Black Box</i> |
| | | Validasi menyimpan, mengedit dan menghapus data | <i>Black Box</i> |
| | | Menyimpan data ke <i>database</i> | <i>Black Box</i> |
| 5 | Dashboard | Menampilkan data ke dalam grafik | <i>Black Box</i> |
| | | Pencarian alumni | <i>Black Box</i> |
| 6 | Chat | Menampilkan data | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |
| | | Mengirim Pesan | <i>Black Box</i> |
| | | Menerima Pesan | <i>Black Box</i> |
| | | Hapus pesan | <i>Black Box</i> |
| | | Notifikasi pesan baru | <i>Black Box</i> |
| 7 | Informasi Angkatan | Menampilkan data | <i>Black Box</i> |
| | | Memilih dan menampilkan profil alumni | <i>Black Box</i> |
| 8 | Informasi Tempat Bekerja Alumni | Menampilkan data | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |
| | | Memilih dan menampilkan profil alumni | <i>Black Box</i> |
| 9 | Informasi Forum | Menampilkan data | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |
| | | Validasi Menyimpan, Mengedit dan menghapus data | <i>Black Box</i> |
| | | Menyimpan data ke <i>databse</i> | <i>Black Box</i> |
| 10 | Informasi Loker | Menampilkan data | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |
| | | Validasi menyimpan, mengedit dan menghapus data | <i>Black Box</i> |
| | | Menyimpan data ke <i>databse</i> | <i>Black Box</i> |
| 11 | Informasi Pengumuman | Menampilkan data | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |
| | | Validasi menyimpan, mengedit dan menghapus data | <i>Black Box</i> |
| | | Menyimpan data ke <i>databse</i> | <i>Black Box</i> |
| 12 | Lokasi | Menampilkan data | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |

| | | | |
|----|--|--|------------------|
| 13 | Profil | Menampilkan data Verifikasi Menyimpan dan mengedit data Set lokasi Ubah <i>password</i> Menyimpan data ke database | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | Set lokasi | <i>Black Box</i> |
| | | Ubah <i>password</i> | <i>Black Box</i> |
| | | Menyimpan data ke database | <i>Black Box</i> |
| 14 | <i>Administrator Pengelolaan user</i> | Menampilkan data Pencarian Validasi menyimpan dan mengedit data Menyimpan data ke <i>databse</i> | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |
| | | Validasi menyimpan dan mengedit data | <i>Black Box</i> |
| | | Menyimpan data ke <i>databse</i> | <i>Black Box</i> |
| 15 | <i>Administrator Data alumni</i> | Menampilkan data Pencarian <i>export</i> ke pdf, json atau csv | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | Pencarian | <i>Black Box</i> |
| | | <i>export</i> ke pdf, json atau csv | <i>Black Box</i> |
| 16 | <i>Administrator Rekap tracer alumni</i> | Menampilkan data <i>Filter</i> tahun awal <i>export</i> ke pdf, json atau csv | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | <i>Filter</i> tahun awal | <i>Black Box</i> |
| | | <i>export</i> ke pdf, json atau csv | <i>Black Box</i> |
| 17 | <i>Administrator Fakultas</i> | Menampilkan data Validasi menyimpan, mengedit dan menghapus data Menyimpan data ke <i>databse</i> | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | Validasi menyimpan, mengedit dan menghapus data | <i>Black Box</i> |
| | | Menyimpan data ke <i>databse</i> | <i>Black Box</i> |
| 18 | <i>Administrator Database</i> | <i>Export Json</i> <i>Import Json</i> | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | <i>Import Json</i> | <i>Black Box</i> |
| 19 | <i>Logout</i> | Menghapus sesi masuk dan kembali ke form login | <i>Black Box</i> |

Tabel 5.11 Rencana Pengujian *Interface Sistem*

| Rencana Pengujian Interface Sistem | | | |
|---|-----------------|--|------------------|
| No | Item Uji | Detail Uji | Jenis Uji |
| 1 | Registrasi | Mengisi semua <i>field</i> Menekan button daftar Menekan button pilih <i>login</i> | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | Menekan button daftar | <i>Black Box</i> |
| | | Menekan button pilih <i>login</i> | <i>Black Box</i> |
| 2 | <i>Login</i> | Mengisi field <i>email</i> dan <i>password</i> Menekan button <i>login</i> | <i>Black Box</i> |
| | | | <i>Black Box</i> |
| | | Menekan button <i>login</i> | <i>Black Box</i> |

| | | | |
|----|---------------------------------|---|------------------|
| | | Menekan <i>button</i> pilih daftar | <i>Black Box</i> |
| 3 | <i>Reset Password</i> | Mengisi <i>field email</i> | <i>Black Box</i> |
| | | Menekan <i>button reset</i> | <i>Black Box</i> |
| 4 | Kuesioner | Mengisi dan memilih kuesioner | <i>Black Box</i> |
| | | Menekan <i>button simpan</i> | <i>Black Box</i> |
| 5 | Menu Utama | Menekan <i>button dashboard</i> | <i>Black Box</i> |
| | | Menekan <i>button chat</i> | <i>Black Box</i> |
| | | Menekan <i>button informasi</i> | <i>Black Box</i> |
| | | Menekan <i>button lokasi</i> | <i>Black Box</i> |
| | | Menekan <i>button profil</i> | <i>Black Box</i> |
| 6 | <i>Dashboard</i> | Menekan <i>button cari alumni</i> | <i>Black Box</i> |
| | | Melihat hasil <i>tracer studyalumni</i> | <i>Black Box</i> |
| | | Menekan <i>button administrator</i> | <i>Black Box</i> |
| 7 | <i>Chatting</i> | Menekan list <i>chatting</i> | <i>Black Box</i> |
| | | Menekan <i>button cari kontak</i> | <i>Black Box</i> |
| | | Menekan <i>button kirim pesan</i> | <i>Black Box</i> |
| | | Menekan <i>button hapus kontak</i> | <i>Black Box</i> |
| 8 | Informasi | Menekan <i>menu angkatan</i> | <i>Black Box</i> |
| | | Menekan <i>menu tempat bekerja alumni</i> | <i>Black Box</i> |
| | | Menekan <i>menu forum</i> | <i>Black Box</i> |
| | | Menekan <i>menu loker</i> | <i>Black Box</i> |
| | | Menekan <i>menu pengumuman</i> | <i>Black Box</i> |
| 9 | Informasi Angkatan | Menekan list angkatan | <i>Black Box</i> |
| 10 | Informasi Tempat Bekerja Alumni | Menekan list tempat bekerja alumni | <i>Black Box</i> |
| 11 | Informasi Forum | Mengisi <i>edittext</i> cari | <i>Black Box</i> |
| | | Menekan <i>button tambah</i> | <i>Black Box</i> |
| | | Mengisi semua <i>field</i> | <i>Black Box</i> |
| | | Memilih tanggal <i>expired</i> | <i>Black Box</i> |
| | | Menekan <i>button edit</i> | <i>Black Box</i> |
| | | Menekan <i>button hapus</i> | <i>Black Box</i> |
| 12 | Informasi Loker | Mengisi <i>edittext</i> cari | <i>Black Box</i> |
| | | Menekan <i>button tambah</i> | <i>Black Box</i> |
| | | Mengisi semua <i>field</i> | <i>Black Box</i> |
| | | Memilih tanggal <i>expired</i> | <i>Black Box</i> |
| | | Menekan <i>button edit</i> | <i>Black Box</i> |
| | | Menekan <i>button hapus</i> | <i>Black Box</i> |

| | | | |
|----|--|--|------------------|
| 13 | Informasi Pengumuman | Mengisi <i>edittext</i> cari Menekan <i>button</i> tambah Mengisi semua <i>field</i> Memilih tanggal <i>expired</i> Menekan <i>button edit</i> Menekan <i>button hapus</i> | <i>Black Box</i> |
| | | | |
| 14 | Lokasi | Menekan menu lokasi Menekan <i>button</i> pencarian Menekan penanda map | <i>Black Box</i> |
| | | | |
| 15 | Profil | Menekan <i>menu profil</i> Menekan <i>button set lokasi</i> Menekan <i>button about</i> Menekan <i>button rate this app</i> Menekan <i>button ubah password</i> Menekan <i>button logout</i> | <i>Black Box</i> |
| | | | |
| 16 | Administrator | Menekan menu <i>button pengelolaan user</i> Menekan menu <i>button</i> data alumni Menekan menu <i>button</i> rekap tracer alumni Menekan menu <i>button fakultas</i> Menekan menu <i>button database</i> | <i>Black Box</i> |
| | | | |
| 17 | <i>Administrator</i> Pengelolaan User | Menekan <i>tabs register</i> Mengisi <i>edittext</i> pencarian Menekan <i>button validasi</i> Menekan <i>button tolak</i> Menekan <i>tabs users</i> Mengisi <i>edittext</i> pencarian Menekan <i>button tambah</i> Menekan <i>button ubah email</i> Menekan <i>button ubah akses</i> | <i>Black Box</i> |
| | | | |
| 18 | <i>Administrator</i> Data Alumni | Menekan <i>tabs angkatan</i> | <i>Black Box</i> |

| | | | |
|----|--|--|------------------|
| | | Menekan list angkatan | <i>Black Box</i> |
| | | Menekan <i>tabs</i> tempat bekerja alumni | <i>Black Box</i> |
| | | Menekan list tempat bekerja alumni | <i>Black Box</i> |
| | | Menekan <i>button export</i> | <i>Black Box</i> |
| 19 | <i>Administrator Rekap Tracer Alumni</i> | Mengisi <i>edittext</i> tahun | <i>Black Box</i> |
| | | Menekan <i>button filter</i> | <i>Black Box</i> |
| | | Menekan <i>button export</i> | <i>Black Box</i> |
| 20 | <i>Administrator Fakultas</i> | Menekan menu <i>button</i> fakultas | <i>Black Box</i> |
| | | Menekan <i>button</i> tambah, edit dan hapus fakultas | <i>Black Box</i> |
| | | Menekan <i>button</i> tambah, edit dan hapus program studi | <i>Black Box</i> |
| 21 | <i>Administrator Database</i> | Menekan <i>tabs backup</i> | <i>Black Box</i> |
| | | Memilih semua checklist | <i>Black Box</i> |
| | | Menekan <i>button</i> backup data | <i>Black Box</i> |
| | | Menekan <i>button</i> hapus data | <i>Black Box</i> |
| | | Menekan <i>button</i> hapus semua data | <i>Black Box</i> |
| | | Menekan <i>tabs restore</i> | <i>Black Box</i> |
| | | Menekan <i>button load file</i> | <i>Black Box</i> |
| | | Memilih tujuan tabel <i>restore</i> | <i>Black Box</i> |
| | | Menekan <i>button restore</i> data | <i>Black Box</i> |

5.2.2 Kasus dan Hasil Pengujian

Kasus dan hasil pengujian pada aplikasi dilakukan untuk memenuhi syarat apakah aplikasi tersebut layak untuk di gunakan serta melakukan pengetesan dengan kasus sesuai dengan fungsional aplikasi apakah sudah berjalan dengan baik, tampilan layout berjalan tanpa ada masalah pada saat aplikasi tersebut digunakan. Pengujian dibagi dua terdiri dari rencana pengujian fungsional sistem dan rencana pengujian interface sistem dengan kasus item yang diuji, skenario uji, hasil yang diharapkan dan hasil pengujian.

Tabel 5.12 Hasil Pengujian Fungsionalitas Sistem

| Pengujian Fungsionalitas Sistem | | | | |
|--|-----------------|--|---|------------------------|
| No | Item Uji | Skenario Uji | Hasil yang diharapkan | Hasil Pengujian |
| 1 | Registrasi | Validasi registrasi | Ketika data tidak sesuai muncul notifikasi peringatan | + Berhasil |
| | | Verifikasi cek data ke <i>database</i> | Pencocokan data jika data sudah terdaftar muncul notifikasi peringatan | + Berhasil |
| | | Menyimpan data ke <i>database</i> | Hasil inputan registrasi yang sudah diverifikasi tersebut disimpan ke database | + Berhasil |
| 2 | Login | Validasi <i>login</i> | Ketika data tidak sesuai muncul notifikasi peringatan | + Berhasil |
| | | Verifikasi cek data ke <i>database</i> | Pencocokan data jika data tidak ada muncul notifikasi peringatan | + Berhasil |
| | | Validasi <i>email</i> | Ketika data tidak sesuai muncul notifikasi peringatan | + Berhasil |
| 3 | Reset Password | Verifikasi <i>email</i> | Pencocokan data jika data tidak ada muncul notifikasi peringatan | + Berhasil |
| | | Mengirim link <i>reset password</i> melalui <i>email</i> | Dapat mengirim link reset password akun melalui email dengan aktivasi password baru | + Berhasil |
| | | Menampilkan data | Menampilkan data kuesioner tracer alumni | + Berhasil |
| 4 | Kuesioner | Validasi menyimpan, mengedit dan menghapus data | Ketika melakukan pengisian yang tidak sesuai memunculkan notifikasi peringatan | + Berhasil |
| | | Menyimpan data ke <i>database</i> | Data disimpan keserver <i>database</i> | + Berhasil |
| | | Menampilkan data ke dalam grafik | Menampilkan data alumni dan hasil kuesioner tracer alumni dalam bentuk grafik | + Berhasil |
| 5 | Dashboard | | | |

| | | | | |
|----|---------------------------------|---|--|------------|
| | | Pencarian alumni | Melakukan pencarian alumni dan menampilkan data yang dicari | + Berhasil |
| 6 | Chat | Menampilkan data | Menampilkan data chat user pengguna | + Berhasil |
| | | Pencarian | Melakukan pencarian histori chat dan kontak | + Berhasil |
| | | Mengirim Pesan | Mengirim pesan kepada user yang dipilih | + Berhasil |
| | | Menerima Pesan | Menerima pesan dari user pengirim | + Berhasil |
| | | Hapus pesan | Menghapus pesan sesuai dengan pengirim dan penerima | + Berhasil |
| | | Notifikasi pesan baru | Notifikasi pesan baru muncul | + Berhasil |
| 7 | Informasi Angkatan | Menampilkan data | Menampilkan data angkatan alumni | + Berhasil |
| | | Memilih dan menampilkan profil alumni | Ketika mencari dan memilih kontak alumni dapat menampilkan profil sesuai data yang dipilih | + Berhasil |
| 8 | Informasi Tempat Bekerja Alumni | Menampilkan data | Menampilkan data tempat bekerja alumni | + Berhasil |
| | | Pencarian | Melakukan pencarian data tempat bekerja alumni | + Berhasil |
| | | Memilih dan menampilkan profil alumni | Ketika mencari dan memilih kontak alumni dapat menampilkan profil sesuai data yang dipilih | + Berhasil |
| 9 | Informasi Forum | Menampilkan data | Menampilkan data forum alumni | + Berhasil |
| | | Pencarian | Melakukan pencarian data forum alumni | + Berhasil |
| | | Validasi Menyimpan, Mengedit dan Menghapus data | Ketika melakukan pengisian yang tidak sesuai memunculkan notifikasi peringatan | + Berhasil |
| | | Menyimpan data kedatabse | Data disimpan keserver database | + Berhasil |
| 10 | Informasi Loker | Menampilkan data | Menampilkan data loker alumni | + Berhasil |

| | | | | |
|----|---------------------------------------|---|--|------------|
| | | Pencarian | Melakukan pencarian data lowongan pekerjaan | + Berhasil |
| | | Validasi Menyimpan, Mengedit dan Menghapus data | Ketika melakukan pengisian yang tidak sesuai memunculkan notifikasi peringatan | + Berhasil |
| | | Menyimpan data ke <i>databse</i> | Data disimpan ke <i>server database</i> | + Berhasil |
| 11 | Informasi Pengumuman | Menampilkan data | Menampilkan data pengumuman alumni | + Berhasil |
| | | Pencarian | Melakukan pencarian berdasarkan kategori program studi dan tahun angkatan | + Berhasil |
| | | Validasi Menyimpan, Mengedit dan Menghapus data | Ketika melakukan pengisian yang tidak sesuai memunculkan notifikasi peringatan | + Berhasil |
| | | Menyimpan data ke <i>databse</i> | Data disimpan ke <i>server database</i> | + Berhasil |
| 12 | Lokasi | Menampilkan data | Menampilkan data lokasi alumni | + Berhasil |
| | | Pencarian | Menampilkan posisi lokasi alumni jika alumni sudah menset lokasi | + Berhasil |
| 13 | Profil | Menampilkan data | Menampilkan data profil alumni | + Berhasil |
| | | Verifikasi Menyimpan dan mengedit data | Jika data tidak sesuai akan memunculkan notifikasi pesan peringatan | + Berhasil |
| | | Set lokasi | Set lokasi alumni untuk ditampilkan pada menu <i>maps</i> lokasi | + Berhasil |
| | | Ubah <i>password</i> | Merubah <i>password user</i> | + Berhasil |
| | | Menyimpan data ke <i>databse</i> | Data disimpan keserver database | + Berhasil |
| 14 | Administrator Pengelolaan <i>user</i> | Menampilkan data | Menampilkan data pengelolaan user | + Berhasil |
| | | Pencarian | Melakukan pencarian user | + Berhasil |
| | | Validasi menyimpan dan mengedit data | Ketika melakukan pengisian yang tidak sesuai memunculkan | + Berhasil |

| | | | | |
|----|--|---|---|------------|
| | | | notifikasi peringatan | |
| | | Menyimpan data ke <i>databse</i> | Data disimpan ke <i>server database</i> | + Berhasil |
| 15 | <i>Administrator</i> Data alumni | Menampilkan data | Menampilkan data alumni yang sudah terdaftar | + Berhasil |
| | | Pencarian | Melakukan pencarian data alumni | + Berhasil |
| | | <i>export</i> ke pdf, json atau csv | <i>Export</i> ke file pdf, json dan csv ke penyimpanan lokal <i>smartphone</i> | + Berhasil |
| 16 | <i>Administrator</i> Rekap <i>Tracer</i> alumni | menampilkan data | Menampilkan data hasil <i>tracer</i> alumni | + Berhasil |
| | | Filter tahun awal | Menampilkan data sesuai tahun awal sesuai yang di <i>filter</i> | + Berhasil |
| | | <i>export</i> ke pdf, json atau csv | <i>Export</i> ke file pdf, json dan csv ke penyimpanan lokal <i>smartphone</i> | + Berhasil |
| 17 | <i>Administrator</i> Fakultas | Menampilkan data | Menampilkan data sesuai menu yang dipilih | + Berhasil |
| | | Validasi menyimpan, mengedit dan menghapus data | Ketika melakukan pengisian yang tidak sesuai memunculkan notifikasi peringatan | + Berhasil |
| | | Menyimpan data kedatabase | Data disimpan keserver database | + Berhasil |
| 18 | <i>Administrator</i> <i>Database</i> | <i>Export Json</i> | <i>Export</i> ke file .json ke penyimpanan lokal <i>smartphone</i> | + Berhasil |
| | | <i>Import Json</i> | <i>Import</i> file .json ke aplikasi kemudian jika data sesuai akan dikirim ke <i>database server</i> | + Berhasil |
| 19 | Logout | Menghapus sesi masuk dan kembali ke form login | Ketika <i>logout</i> sesi di aplikasi terhapus dan kembali ke menu login atau <i>register</i> | + Berhasil |

Tabel 5.13 Hasil Pengujian Interface Sistem

Pengujian Interface Aplikasi

| No | Item Uji | Skenario Uji | Hasil yang diharapkan | Hasil Pengujian |
|----|-----------------------|--|--|-----------------|
| 1 | Registrasi | Mengisi semua <i>field</i> | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |
| | | Menekan <i>button</i> daftar | Ketika menekan tombol daftar menampilkan langsung form daftar | + Berhasil |
| | | Menekan <i>button</i> pilih <i>login</i> | Ketika menekan tombol pilih <i>login</i> maka menampilkan langsung form <i>login</i> | + Berhasil |
| 2 | <i>Login</i> | Mengisi <i>field email</i> dan <i>password</i> | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |
| | | Menekan <i>button login</i> | Ketika menekan button <i>login</i> jika data tidak sesuai menampilkan notifikasi peringatan | + Berhasil |
| | | Menekan <i>button</i> pilih daftar | Ketika menekan tombol pilih daftar maka menampilkan langsung form daftar | + Berhasil |
| 3 | <i>Reset Password</i> | Mengisi <i>field email</i> | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |
| | | Menekan <i>button reset</i> | Ketika menekan <i>button reset</i> jika gagal atau berhasil akan menampilkan notifikasi | + Berhasil |
| 4 | Kuesioner | Mengisi dan memilih kuesioner | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |
| | | Menekan <i>button simpan</i> | Ketika menekan <i>button simpan</i> jika data tidak sesuai menampilkan notifikasi peringatan | + Berhasil |
| 5 | Menu Utama | Menekan <i>button dashboard</i> | Menampilkan form <i>dashboard</i> | + Berhasil |
| | | Menekan <i>button chat</i> | Menampilkan form <i>chat</i> | + Berhasil |
| | | Menekan <i>button</i> informasi | Menampilkan form informasi dengan sub menu forum, loker dan list alumni | + Berhasil |
| | | Menekan <i>button</i> lokasi | Menampilkan form lokasi | + Berhasil |

| | | | | |
|----|---------------------------------|---|---|------------|
| | | Menekan <i>button</i> profil | Menampilkan <i>form</i> profil | + Berhasil |
| 6 | <i>Dashboard</i> | Menekan <i>button</i> cari alumni | Ketika menekan <i>button</i> cari alumni akan menampilkan halaman list alumni | + Berhasil |
| | | Melihat hasil <i>tracer studyalumni</i> | Menampilkan data dalam bentuk grafik | + Berhasil |
| | | Menekan <i>button administrator</i> | Menampilkan form administrator menu lanjutan | + Berhasil |
| 7 | <i>Chatting</i> | Menekan list chatting | Menampilkan isi chatting | + Berhasil |
| | | Menekan <i>button</i> cari kontak | Menampilkan list kontak | + Berhasil |
| | | Menekan <i>button</i> kirim pesan | Menampilkan isi dan tampilan kirim pesan | + Berhasil |
| | | Menekan <i>button</i> hapus kontak | Menampilkan dialog hapus atau tidak | + Berhasil |
| 8 | Informasi | Menekan menu angkatan | Menampilkan list angkatan | + Berhasil |
| | | Menekan menu tempat bekerja alumni | Menampilkan list tempat bekerja alumni | + Berhasil |
| | | Menekan menu forum | Menampilkan list forum | + Berhasil |
| | | Menekan menu loker | Menampilkan list loker | + Berhasil |
| | | Menekan menu pengumuman | Menampilkan list pengumuman | + Berhasil |
| 9 | Informasi Angkatan | Menekan list angkatan | Ketika menekan list angkatan akan menampilkan list alumni | + Berhasil |
| 10 | Informasi Tempat Bekerja Alumni | Menekan list tempat bekerja alumni | Ketika menekan list tempat bekerja alumni akan menampilkan list alumni | + Berhasil |
| 11 | Informasi Forum | Mengisi <i>edittext</i> cari | Ketika <i>edittext</i> cari diisi akan menampilkan list sesuai dengan yang dicari | + Berhasil |
| | | Menekan <i>button</i> tambah | Ketika menekan <i>button</i> tambah maka akan menampilkan halaman isian forum | + Berhasil |

| | | | | |
|----|----------------------|------------------------------|---|------------|
| | | Mengisi semua field | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |
| | | Memilih tanggal expired | Ketika menekan field expired muncul panel dialog memilih tanggal | + Berhasil |
| | | Menekan button edit | Ketika menekan button edit maka akan menampilkan halaman isian forum | + Berhasil |
| | | Menekan button hapus | Ketika menekan button hapus maka akan dihapus sesuai data yang dipilih | + Berhasil |
| 12 | Informasi Loker | Mengisi edittext cari | Ketika edittext cari diisi akan menampilkan list sesuai dengan yang dicari | + Berhasil |
| | | Menekan button tambah | Ketika menekan button tambah maka akan menampilkan halaman isian loker | + Berhasil |
| | | Mengisi semua field | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |
| | | Memilih tanggal expired | Ketika menekan field expired muncul panel dialog memilih tanggal | + Berhasil |
| | | Menekan button edit | Ketika menekan button tambah maka akan menampilkan halaman isian loker | + Berhasil |
| | | Menekan button hapus | Ketika menekan button hapus maka akan dihapus sesuai data yang dipilih | + Berhasil |
| 13 | Informasi Pengumuman | Mengisi <i>edittext</i> cari | Ketika edittext cari diisi akan menampilkan list sesuai dengan yang dicari | + Berhasil |
| | | Menekan button tambah | Ketika menekan button tambah maka akan menampilkan halaman isian pengumuman | + Berhasil |
| | | Mengisi semua field | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |

| | | | | |
|----|---------------|---|--|------------|
| | | Memilih tanggal <i>expired</i> | Ketika menekan field <i>expired</i> muncul panel dialog memilih tanggal | + Berhasil |
| | | Menekan <i>button edit</i> | Ketika menekan button tambah maka akan menampilkan halaman isian pengumuman | + Berhasil |
| | | Menekan <i>button hapus</i> | Ketika menekan button hapus maka akan dihapus sesuai data yang dipilih | + Berhasil |
| 14 | Lokasi | Menekan menu lokasi | Ketika dipilih menampilkan form lokasi | + Berhasil |
| | | Menekan <i>button pencarian</i> | Ketika dipilih menampilkan <i>form pencarian lokasi</i> | + Berhasil |
| | | Menekan penanda map | Ketika dipilih menampilkan informasi profil alumni | + Berhasil |
| 15 | Profil | Menekan menu profil | Ketika dipilih menampilkan form profil | + Berhasil |
| | | Menekan <i>button set lokasi</i> | Ketika menekan menampilkan lokasi map latitude dan longitude | + Berhasil |
| | | Menekan <i>button about</i> | Ketika dipilih akan menampilkan dialog info aplikasi dan ketika button help dipilih maka akan direct ke situs Fti Unibba | + Berhasil |
| | | Menekan <i>button rate this app</i> | Ketika dipilih akan <i>direct</i> ke situs <i>Playstore</i> aplikasi | + Berhasil |
| | | Menekan <i>button ubah password</i> | Ketika dipilih maka akan menampilkan halaman ubah password | + Berhasil |
| | | Menekan <i>button logout</i> | Ketika dipilih menampilkan form login | + Berhasil |
| 16 | Administrator | Menekan menu <i>button pengelolaan user</i> | Ketika dipilih menampilkan halaman pengelolaan user | + Berhasil |
| | | Menekan menu <i>button data</i> | Ketika dipilih menampilkan halaman | + Berhasil |

| | | | | |
|----|--|---|---|------------|
| | | alumni | data alumni | |
| | | Menekan menu <i>button</i> rekap <i>tracer</i> alumni | Ketika dipilih menampilkan halaman rekap <i>tracer</i> alumni | + Berhasil |
| | | Menekan menu <i>button</i> fakultas | Ketika dipilih menampilkan halaman fakultas | + Berhasil |
| | | Menekan menu <i>button database</i> | Ketika dipilih menampilkan halaman <i>database</i> | + Berhasil |
| 17 | <i>Administrator</i> Pengelolaan <i>User</i> | Menekan <i>tabs register</i> | Ketika dipilih menampilkan halaman <i>register</i> | + Berhasil |
| | | Mengisi <i>edittext</i> pencarian | Ketika <i>edittext</i> cari diisi akan menampilkan list sesuai dengan yang dicari | + Berhasil |
| | | Menekan <i>button</i> validasi | Ketika menekan <i>button</i> validasi maka akan divalidasi sesuai data yang dipilih | + Berhasil |
| | | Menekan <i>button</i> tolak | Ketika menekan <i>button</i> tolak maka akan ditolak sesuai data yang dipilih | + Berhasil |
| | | Menekan <i>tabs users</i> | Ketika dipilih menampilkan halaman <i>users</i> | + Berhasil |
| | | Mengisi <i>edittext</i> pencarian | Ketika mengisi pencarian list pengelolaan <i>users</i> akan muncul sesuai data yang dicari | + Berhasil |
| | | Menekan <i>button</i> tambah | Ketika menekan button tambah maka akan muncul dialog panel tambah <i>user</i> | + Berhasil |
| | | Menekan <i>button</i> ubah <i>email</i> | Ketika menekan <i>button</i> ubah <i>email</i> maka akan muncul dialog panel tambah <i>user email</i> | + Berhasil |
| | | Menekan <i>button</i> ubah akses | Ketika menekan button ubah akses maka akan muncul dialog panel ubah user akses | + Berhasil |
| 18 | <i>Administrator</i> Data Alumni | Menekan <i>tabs angkatan</i> | Ketika dipilih menampilkan halaman angkatan | + Berhasil |

| | | | | |
|----|--|--|--|------------|
| | | Menekan list angkatan | Ketika dipilih menampilkan halaman list alumni | + Berhasil |
| | | Menekan <i>tabs</i> tempat bekerja alumni | Ketika dipilih menampilkan halaman tempat bekerja alumni | + Berhasil |
| | | Menekan list tempat bekerja alumni | Ketika dipilih menampilkan halaman list alumni | + Berhasil |
| | | Menekan <i>button export</i> | Ketika dipilih akan menampilkan notifikasi berhasil atau gagal | + Berhasil |
| 19 | <i>Administrator Rekap Tracer Alumni</i> | Mengisi <i>edittext</i> tahun | Inputan benar dan jika salah menampilkan notifikasi peringatan | + Berhasil |
| | | Menekan <i>button filter</i> | Ketika dipilih akan menampilkan data rekap tracer alumni | + Berhasil |
| | | Menekan <i>button export</i> | Ketika dipilih akan menampilkan notifikasi berhasil atau gagal | + Berhasil |
| 20 | <i>Administrator Fakultas</i> | Menekan menu <i>button</i> fakultas | Ketika dipilih akan menampilkan halaman fakultas | + Berhasil |
| | | Menekan <i>button tambah, edit dan hapus fakultas</i> | Ketika dipilih akan menampilkan halaman fakultas tambah, edit dan hapus sesuai yang dipilih | + Berhasil |
| | | Menekan <i>button tambah, edit dan hapus program studi</i> | Ketika dipilih akan menampilkan halaman program studi tambah, edit dan hapus sesuai yang dipilih | + Berhasil |
| 21 | <i>Administrator Database</i> | Menekan <i>tabs backup</i> | Ketika dipilih menampilkan halaman backup | + Berhasil |
| | | Memilih semua <i>checklist</i> | Memilih <i>checklist</i> dan <i>unchecklist</i> sesuai yang dipilih | + Berhasil |
| | | Menekan <i>button backup data</i> | Ketika <i>button backup data</i> dipilih jika gagal atau berhasil akan menampilkan notifikasi | + Berhasil |
| | | Menekan <i>button hapus data</i> | Ketika <i>button hapus data</i> dipilih jika gagal atau berhasil akan | + Berhasil |

| | | | | |
|--|--|--|--|------------|
| | | | menampilkan notifikasi | |
| | | Menekan <i>button hapus</i> semua data | Ketika <i>button hapus</i> semua data dipilih jika gagal atau berhasil akan menampilkan notifikasi | + Berhasil |
| | | Menekan <i>tabs restore</i> | Ketika <i>tabs restore</i> dipilih menampilkan halaman restore | + Berhasil |
| | | Menekan <i>button load file</i> | Menekan <i>button load file</i> dan menampilkan informasi lokasi path <i>file</i> berada | + Berhasil |
| | | Memilih tujuan tabel <i>restore</i> | Menampilkan list tujuan tabel yang akan dipilih | + Berhasil |
| | | Menekan <i>button restore data</i> | Ketika <i>button restore data</i> dipilih jika gagal atau berhasil akan menampilkan notifikasi | + Berhasil |

Keterangan: + Berhasil, = Tidak berhasil

5.2.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian melakukan kegiatan *tracer study* alumni studi kasus di Fakultas Teknologi Informasi Universitas Bale Bandung dengan kasus *Black box* dapat ditarik kesimpulan bahwa perangkat lunak dapat mengetahui fungsi – fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *database eksternal*, kesalahan kinerja, inisialisasi, kesalahan terminasi dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan. membantu permasalahan pada kegiatan pengumpulan data *tracer study* alumni dan memper-erat silaturahmi antar alumni dan fakultas. Tidak menutup kemungkinan dapat terjadi kesalahan suatu saat pada saat aplikasi digunakan sehingga membutuhkan proses perawatan atau maintenance untuk lebih mengetahui kekurangan dari aplikasi sehingga dapat dikembangkan lagi lebih lanjut.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian dan analisa serta perancangan aplikasi yang telah dihasilkan, penulis menyelesaikan laporan dan aplikasi *Tracer Alumni Berbasis Android Studi Kasus Di Fakultas Teknologi Informasi* maka dapat ditarik kesimpulan bahwa:

1. Penelitian ini menghasilkan suatu sistem *tracer study* alumni berbasis android yang dapat diakses melalui perangkat smartphone android.
2. Aplikasi *mobile tracer study* lumni dapat diterapkan dalam membantu proses pendataan alumni FTI UNIBBA dengan metode penyimpanan data yang dinamis dan dapat diperbarui dengan mudah oleh para alumni langsung menggunakan aplikasi mobile berbasis android. Hasil pengujian yang dilakukan pada berbagai varian android, aplikasi dapat berjalan secara optimal kecuali versi-versi yang merupakan diluar Batasan minimum android versi 4.2 *jelly bean*.
3. Sistem dapat menghasilkan laporan berupa grafik dan tabel berdasarkan kriteria pencarian, sehingga memudahkan pihak FTI UNIBBA dan bagian akademik dalam mengetahui hasil *tracer study* alumni.

6.2 Saran

Aplikasi *tracer study* ini masih memiliki banyak kekurangan, oleh sebab itu penulis berharap aplikasi ini dapat dikembangkan lebih lanjut demi tercapainya hasil sesuai dengan yang diharapkan. Untuk itu penulis memberikan beberapa saran sebagai berikut:

1. Untuk pengembangan selanjutnya diharapkan aplikasi ini dapat dikembangkan menggunakan *framework react*, *Cordova* atau *framework* lainnya menjadi *universal app* berbasis *web* dan *mobile* sehingga bisa dijalankan *multi platform* serta lebih memudahkan dalam proses *tracer study* di FTI UNIBBA.

2. Perlu ditambahkan sebuah prosedur kuisioner yang bersifat dinamis dengan memanfaatkan basis data sehingga kuisioner dapat menyesuaikan keperluan dari manajemen FTI UNIBBA.
3. Perlu ditambahkan sebuah prosedur *backup* data secara berkala dengan lokasi khusus untuk penyimpanan backup data.

DAFTAR PUSTAKA

- Android Studio.* (2019). Retrieved from Android Studio:
<https://developer.android.com/studio/intro?hl=ID>
- Balsamiq. (2019). *Balsamiq Wireframes.* Retrieved from Balsamiq:
<https://balsamiq.com/products/mockups/>
- D. E., & S. W. (n.d.). *Tentang Tracer Study.* Retrieved from ITB Tracer Study:
<https://tracer.itb.ac.id/id/tentang/tentang-tracer-study>
- dikti, b. r. (n.d.). Retrieved from
<https://belmawa.ristekdikti.go.id/2016/08/02/pelaksanaan-tracer-study-tingkat-perguruan-tinggi>
- Hakim, L. e. (2017). Perancangan Sistem Tracer Alumni Pada STMIK Musi Rawas Berbasis Web Mobile. *Perancangan Sistem Tracer Alumni Pada STMIK Musi Rawas Berbasis Web Mobile.*
- Millington, C., & College, B. C. (2006). The Use of Tracer Studies for Enhancing Relevance and Marketability in Online and Distance Education. *Tracer Studies*, 5. Retrieved from
https://www.academia.edu/27236474/The_Use_of_Tracer_Studies_for_Enhancing_Relevance_and_Marketability_in_Online_and_Distance_Education
- Muzakki, M. S. (2016, June 2). Pengembangan Aplikasi Tracer Study Berbasis Android. *Golden Age Jurnal Ilmiah Tumbuh Kembang Anak Usia Dini*, Vol. 1 No. 2 Juni 2016, 7. Retrieved from
https://www.academia.edu/32262225/Pengembangan_Aplikasi_Tracer_Study_Berbasis_Android
- Posdot, T. (2017, September 13). *Postdot Technology.* Retrieved from Postman is the most complete API Development Environment:
<https://www.getpostman.com/postman>
- Rosa, A. S., & Shalahuddin, M. (2011). Modul Pembelajaran: Rekayasa Perangkat Lunak, Modula, Bandung. *Modul Pembelajaran: Rekayasa Perangkat Lunak, Modula, Bandung.* Retrieved from Modul Pembelajaran: Rekayasa Perangkat Lunak, Modula, Bandung.

- Schomburg, K. H. (2003, July 1). *Handbook of Tracer Study*. Germany: Schomburg, Harald: "Standard Instrument for Graduate and Employer Surveys". Retrieved from <http://www.unhas.ac.id/hasbi/lkpp/hasbi-kbk-softskill-unistaff-scl/hasbi-unistaff-document/modul%20unistaff%20surabaya%202006/qt/tayangan/handbook%20of%20tracer%20study.pdf>
- uptodown. (2019). Retrieved from uptodown:
<https://genymotion.id.uptodown.com/windows>
- Widianto, D. E. (2012). Pemodelan Sistem dengan UML. *Pemodelan Sistem dengan UML*.
- Wikipedia. (2019, 08 19). *Wikipedia*. Retrieved from Wikipedia:
<https://id.wikipedia.org>
- Yadi, S. Y., & Susihono, W. (2013). Evaluasi Tracer Study Untuk Pembelajaran dengan Pendekatan Ergonomi Makro. *Evaluasi Tracer Study Untuk Pembelajaran dengan Pendekatan Ergonomi Makro*.

LAMPIRAN

Lampiran 1 Form Wawancara

1. Wawancara Pertama

Narasumber : Ahmad Faozan Muntaha S.Kom

Hari / Tanggal: Jumat / 07 Mei 2019

Waktu : 16:00

Tempat : Ruangan Tata Usaha

| No | Pertanyaan | Jawaban | |
|----|--|---------|-------|
| | | Ya | Tidak |
| 1 | Apakah Pengolahan Data Alumni Masih Manual? | ✓ | |
| 2 | Apakah pengumpulan data hasil Kuesioner <i>Tracer</i> Alumni sudah rapih dan sesuai standar, baik <i>Hardcopy</i> maupun <i>Softcopy</i> ? | | ✓ |
| 3 | Apakah Semua Lulusan Merespon Kegiatan <i>Tracer</i> Alumni? | | ✓ |
| 4 | Apakah alumni mengisi Kuesioner <i>Tracer</i> Alumni melalui aplikasi yang sudah disediakan? | | ✓ |
| 5 | Apakah perlu adanya aplikasi untuk membantu proses pencarian hasil Kuesioner <i>Tracer</i> Alumni? | ✓ | |
| 6 | Apakah aplikasi berbasis android dapat membantu mempermudah dalam pengumpulan data alumni? | ✓ | |
| 7 | Apakah alumni diberi fasilitas <i>online</i> untuk berkomunikasi dengan antar alumni maupun fakultas? | ✓ | |

2. Wawancara Kedua

Narasumber : Yusuf Muhamar, S.Kom

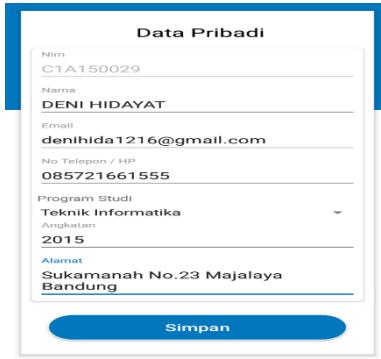
Hari / Tanggal: Jumat / 07 Mei 2019

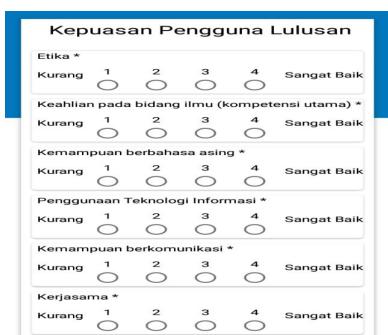
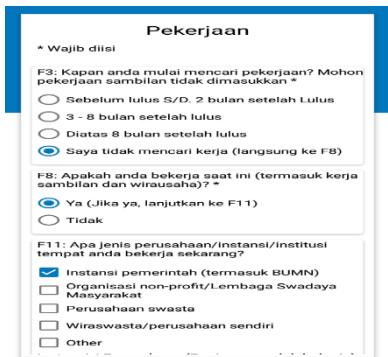
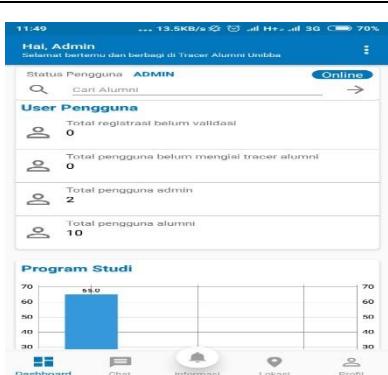
Waktu : 16:10

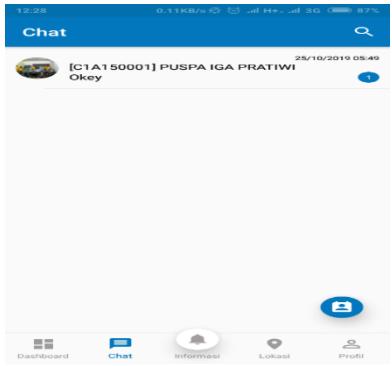
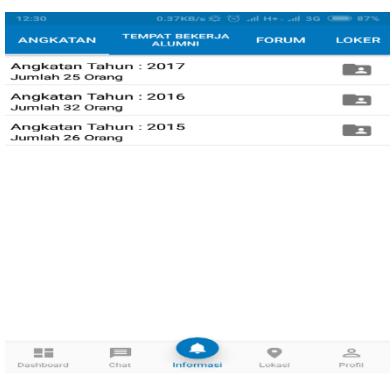
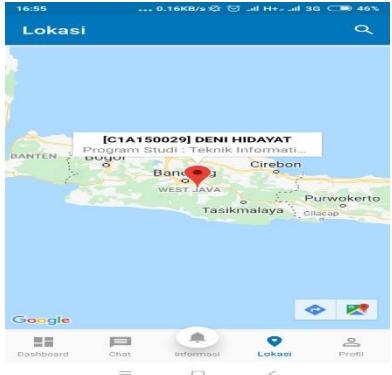
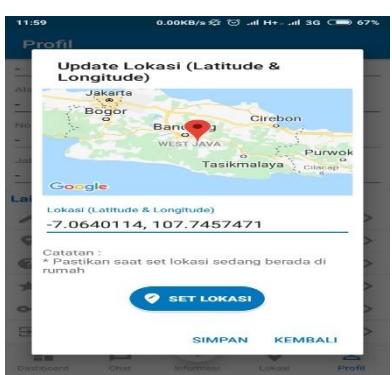
Tempat : Ruangan Tata Usaha

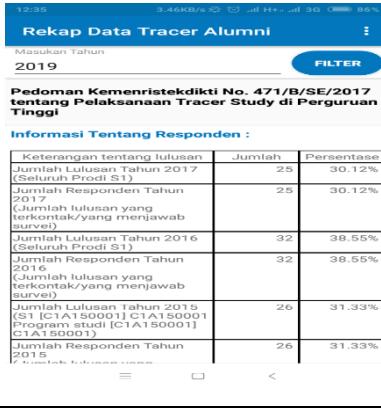
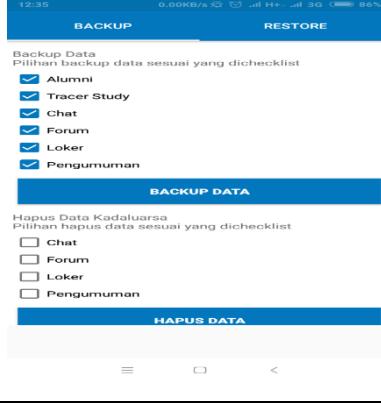
| No | Pertanyaan | Jawaban | |
|----|--|---------|-------|
| | | Ya | Tidak |
| 1 | Apakah Pengolahan Data Alumni Masih Manual? | ✓ | |
| 2 | Apakah pengumpulan data hasil Kuesioner <i>Tracer</i> Alumni sudah rapih dan sesuai standar, baik <i>Hardcopy</i> maupun <i>Softcopy</i> ? | | ✓ |
| 3 | Apakah Semua Lulusan Merespon Kegiatan <i>Tracer</i> Alumni? | | ✓ |
| 4 | Apakah alumni mengisi Kuesioner <i>Tracer</i> Alumni melalui aplikasi yang sudah disediakan? | ✓ | |
| 5 | Apakah perlu adanya aplikasi untuk membantu proses pencarian hasil Kuesioner <i>Tracer</i> Alumni? | ✓ | |
| 6 | Apakah aplikasi berbasis android dapat membantu mempermudah dalam pengumpulan data alumni? | ✓ | |
| 7 | Apakah alumni diberi fasilitas <i>online</i> untuk berkomunikasi dengan antar alumni maupun fakultas? | | ✓ |

Lampiran 2 Testing Program

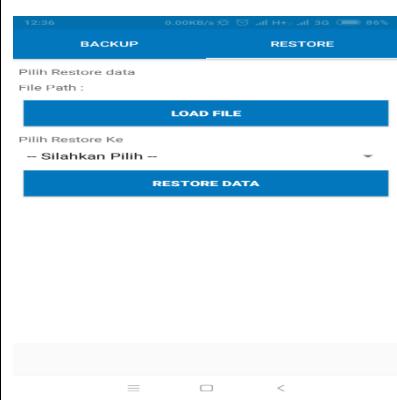
| No | Capture Testing Hasil Pengujian | Keterangan |
|----|---|---------------------------------|
| 1 |  | Tampilan Koneksi Gagal |
| 2 |  | Tampilan Kuesioner |
| 3 |  | Tampilan Kuesioner Data Pribadi |

| | | |
|---|---|--|
| 4 |  | Tampilan Kuesioner |
| 5 |  | Tampilan Kuesioner Kepuasan Pengguna Lulusan |
| 6 |  | Tampilan Kuesioner Riwayat Pekerjaan |
| 7 |  | Tampilan Dashboard |

| | | |
|----|--|---------------------|
| 8 |  <p>12:28 0.11KB/s 3G all H+ all 3G 100% 87%</p> <p>Chat</p> <p>[C1A150001] PUSPA IGA PRATIWI Okey</p> <p>25/10/2019 05:49</p> <p>Dashboard Chat Informasi Lokasi Profil</p> | Tampilan Chat |
| 9 |  <p>12:30 0.16KB/s 3G all H+ all 3G 100% 87%</p> <p>ANGKATAN TEMPAT BEKERJA ALUMNI FORUM LOKER</p> <p>Angkatan Tahun : 2017 Jumlah 25 Orang</p> <p>Angkatan Tahun : 2016 Jumlah 32 Orang</p> <p>Angkatan Tahun : 2015 Jumlah 26 Orang</p> <p>Dashboard Chat Informasi Lokasi Profil</p> | Tampilan Informasi |
| 10 |  <p>16:55 0.16KB/s 3G all H+ all 3G 100% 46%</p> <p>Lokasi</p> <p>[C1A150029] DENI HIDAYAT Program Studi : Teknik Informatika</p> <p>BANTEN Bogor Bandung Cirebon Tasikmalaya Purwokerto Cilegon</p> <p>Google</p> <p>Dashboard Chat Informasi Lokasi Profil</p> | Tampilan Lokasi |
| 11 |  <p>11:59 0.00KB/s 3G all H+ all 3G 100% 67%</p> <p>Profil</p> <p>Update Lokasi (Latitude & Longitude)</p> <p>Jakarta Bogor Bandung Cirebon Tasikmalaya Purwokerto Cilegon</p> <p>Lokasi (Latitude & Longitude) -7.0640114, 107.7457471</p> <p>Catatan : • Pastikan saat set lokasi sedang berada di rumah</p> <p>SET LOKASI</p> <p>SIMPAN KEMBALI</p> <p>Dashboard Chat Informasi Lokasi Profil</p> | Tampilan Set Lokasi |

| 12 |  | Tampilan Profil | | | | | | | | | | | | | | | | | | | | | |
|---|--|----------------------------|--------|------------|--|----|--------|---|----|--------|---|----|--------|---|----|--------|--|----|--------|-----------------------------|----|--------|------------------------------|
| 13 |  | Tampilan Administrator | | | | | | | | | | | | | | | | | | | | | |
| 14 |  <table border="1"> <thead> <tr> <th>Keterangan tentang lulusan</th> <th>Jumlah</th> <th>Persentase</th> </tr> </thead> <tbody> <tr> <td>Jumlah Lulusan Tahun 2017 (S1 [C1A150001] C1A150001)</td> <td>25</td> <td>30.12%</td> </tr> <tr> <td>Jumlah Responden Tahun 2017 (Untuk lulusan yang terkontak/yang menjawab survei)</td> <td>25</td> <td>30.12%</td> </tr> <tr> <td>Jumlah Lulusan Tahun 2016 (S1 [Prodi S1])</td> <td>32</td> <td>38.55%</td> </tr> <tr> <td>Jumlah Responden Tahun 2016 (Untuk lulusan yang terkontak/yang menjawab survei)</td> <td>32</td> <td>38.55%</td> </tr> <tr> <td>Jumlah Lulusan Tahun 2015 (S1 [C1A150001] C1A150001)</td> <td>26</td> <td>31.33%</td> </tr> <tr> <td>Jumlah Responden Tahun 2015</td> <td>26</td> <td>31.33%</td> </tr> </tbody> </table> | Keterangan tentang lulusan | Jumlah | Persentase | Jumlah Lulusan Tahun 2017 (S1 [C1A150001] C1A150001) | 25 | 30.12% | Jumlah Responden Tahun 2017 (Untuk lulusan yang terkontak/yang menjawab survei) | 25 | 30.12% | Jumlah Lulusan Tahun 2016 (S1 [Prodi S1]) | 32 | 38.55% | Jumlah Responden Tahun 2016 (Untuk lulusan yang terkontak/yang menjawab survei) | 32 | 38.55% | Jumlah Lulusan Tahun 2015 (S1 [C1A150001] C1A150001) | 26 | 31.33% | Jumlah Responden Tahun 2015 | 26 | 31.33% | Tampilan Rekap Tracer Alumni |
| Keterangan tentang lulusan | Jumlah | Persentase | | | | | | | | | | | | | | | | | | | | | |
| Jumlah Lulusan Tahun 2017 (S1 [C1A150001] C1A150001) | 25 | 30.12% | | | | | | | | | | | | | | | | | | | | | |
| Jumlah Responden Tahun 2017 (Untuk lulusan yang terkontak/yang menjawab survei) | 25 | 30.12% | | | | | | | | | | | | | | | | | | | | | |
| Jumlah Lulusan Tahun 2016 (S1 [Prodi S1]) | 32 | 38.55% | | | | | | | | | | | | | | | | | | | | | |
| Jumlah Responden Tahun 2016 (Untuk lulusan yang terkontak/yang menjawab survei) | 32 | 38.55% | | | | | | | | | | | | | | | | | | | | | |
| Jumlah Lulusan Tahun 2015 (S1 [C1A150001] C1A150001) | 26 | 31.33% | | | | | | | | | | | | | | | | | | | | | |
| Jumlah Responden Tahun 2015 | 26 | 31.33% | | | | | | | | | | | | | | | | | | | | | |
| 15 |  | Administrator Backup | | | | | | | | | | | | | | | | | | | | | |

16

*Administrator Restore*

Lampiran 3 Listing Program

1. Build.gradle project

```
// Top-level build file where you can add configuration options common to
// all sub-projects/modules.

buildscript {
    ext.kotlin_version = '1.3.11'
    repositories {
        google()
        mavenCentral()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.4.2'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'com.google.gms:google-services:4.3.2'
        // NOTE: Do not place your application dependencies here; they
belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
        jcenter()
        maven { url "https://maven.google.com" }
        maven { url 'https://jitpack.io' }
        maven { url "https://dl.bintray.com/babedev/maven" }
    }
}
```

```
task clean(type: Delete) {
    delete rootProject.buildDir
}
```

2. Build gradle app

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
apply plugin: 'kotlin-kapt'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId 'com.unibba.traceralumni2019'
        minSdkVersion 16
        targetSdkVersion 28
        multiDexEnabled true
        versionCode 5
        versionName '1.5'
        testInstrumentationRunner
        'android.support.test.runner.AndroidJUnitRunner'
    }
    buildTypes {
        release {
            android {
                dexOptions {
                    javaMaxHeapSize "4G"
                }
                lintOptions {
                    checkReleaseBuilds false
                    // Or, if you prefer, you can continue to check for errors in
```

```

release builds,
    // but continue the build even when errors are found:
    abortOnError false
}
}

minifyEnabled false
    proguardFiles      getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
}

}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation "org.jetbrains.kotlin:kotlin-reflect:$kotlin_version"
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.gridlayout:gridlayout:1.0.0'
    implementation 'com.android.support:support-v4:28.0.0'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.android.support:cardview-v7:28.0.0'
    implementation 'com.android.support:gridlayout-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:2.0.0-
alpha3'
    implementation 'com.android.support:support-vector-drawable:28.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-
core:3.0.2'
    implementation 'com.google.firebaseio:firebase-core:17.2.0'
}

```

```
implementation 'com.google.firebaseio:firebase-analytics:17.2.0'  
implementation 'com.google.firebaseio:firebase-auth:19.1.0'  
implementation 'com.firebaseioui:firebase-ui-auth:4.2.0'  
implementation 'com.google.firebaseio:firebase-firebase:21.1.1'  
implementation 'com.google.firebaseio:firebase-messaging:20.0.0'  
implementation 'com.google.firebaseio:firebase-storage:19.1.0'  
implementation 'com.google.android.gms:play-services-flags:17.0.0'  
implementation 'com.google.android.gms:play-services-maps:17.0.0'  
implementation 'com.google.android.gms:play-services-auth:17.0.0'  
implementation 'com.google.android.gms:play-services-location:17.0.0'  
implementation 'com.journeyapps:zxing-android-embedded:3.5.0'  
implementation 'com.github.babedev.dexter.dsl:dsl:1.0.1'  
implementation 'com.google.code.gson:gson:2.8.5'  
implementation 'com.squareup.retrofit2:retrofit:2.5.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.5.0'  
implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'  
implementation 'com.android.support:multidex:1.0.3'  
implementation 'com.scottyab:secure-preferences-lib:0.1.4'  
implementation 'com.theartofdev.edmodo:android-image-cropper:2.3.1'  
implementation 'com.github.bumptech.glide:glide:4.10.0'  
annotationProcessor 'com.github.bumptech.glide:compiler:4.10.0'  
implementation 'com.github.chrisbanes:PhotoView:2.0.0'  
implementation  
'com.toptiche.searchablespinner:searchablespinnerlibrary:1.3.1'  
implementation 'com.github.aliwaris0572:GifDialog:0.0.4'  
implementation 'com.itextpdf:itextg:5.5.10'  
implementation 'com.vanniktech:emoji-ios:0.6.0'  
implementation 'com.github.d-max:spots-dialog:1.1@aar'  
implementation 'de.siegmar:fastcsv:1.0.3'  
implementation 'org.apache.poi:poi:3.17'  
}  
apply plugin: 'com.google.gms.google-services'
```

3. Gradle properties

```

# Project-wide Gradle settings.

# IDE (e.g. Android Studio) users:

# Gradle settings configured through the IDE *will override*
# any settings specified in this file.

# For more details on how to configure your build environment visit
# http://www.gradle.org/docs/current/userguide/build_environment.html

# Specifies the JVM arguments used for the daemon process.

# The setting is particularly useful for tweaking memory settings.

org.gradle.jvmargs=-Xmx1536m

# When configured, Gradle will run in incubating parallel mode.

# This option should only be used with decoupled projects. More details,
visit

#
http://www.gradle.org/docs/current/userguide/multi_project_builds.html#sec:deco
upled_projects

# org.gradle.parallel=true

# Kotlin code style for this project: "official" or "obsolete":
kotlin.code.style=official

android.enableJetifier=true

android.useAndroidX=true

```

4. AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.unibba.traceralumni2019">

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission

```

```
        android:name="android.permission.READ_EXTERNAL_STORAGE"/>
        <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
        <uses-permission android:name="android.permission.FLASHLIGHT"/>
        <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
        <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION"/>
        <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION"/>
        <uses-permission
        android:name="android.permission.SEND_DOWNLOAD_COMPLETED_INTENTS"/>

    <application
        android:largeHeap="true"
        android:allowBackup="true"
        android:hardwareAccelerated="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher"
        android:supportsRtl="true"
        android:theme="@style/AppTheme.NoActionBar"
        tools:ignore="GoogleAppIndexingWarning">
        <!-- make sure within the application tag, otherwise app will crash with
        XmlResourceParser errors -->
        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="com.codepath.fileprovider"
            android:exported="false"
            android:grantUriPermissions="true">
```

```
<meta-data
    android:name="android.support.FILE_PROVIDER_PATHS"
    android:resource="@xml/fileprovider"/>
</provider>
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key"/>
<meta-data android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version"/>
<meta-data

    android:name="com.google.firebaseio.messaging.default_notification_icon"
    android:resource="@mipmap/ic_launcher" />
<meta-data

    android:name="com.google.firebaseio.messaging.default_notification_color"
    android:resource="@color/white" />
<activity

    android:name="com.unibba.traceralumni2019.activity.MainActivity"
    android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>

        <category
            android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.DaftarActivity"
    android:theme="@style/AppTheme.NoActionBar">
```

```
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.ResetActivity"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity

        android:name="com.unibba.traceralumni2019.activity.LoginActivity"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity

        android:name="com.unibba.traceralumni2019.activity.Validasi0Activity"
        android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
    <activity

        android:name="com.unibba.traceralumni2019.activity.Validasi1Activity"
        android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
    <activity

        android:name="com.unibba.traceralumni2019.activity.Validasi2Activity"
        android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
    <activity

        android:name="com.unibba.traceralumni2019.activity.Tracerstudy0Activity"
```

```
        android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.Tracerstudy1Activity"

    android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.Tracerstudy2Activity"

    android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.Tracerstudy4Activity"

    android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.Tracerstudy3Activity"

    android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
    </activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.AdminActivity"
        android:theme="@style/AppTheme"
        android:label="Admin Menu">
```

```
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.PengelolaanUserActivity"
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.LaporanDataAlumniActivit
y"
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.LaporanRekapDataTracerA
lumniActivity"
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.FakultasActivity"
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.BackupActivity"
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.ProfilActivity"
        android:label="@string/profil"
```

```
    android:theme="@style/AppTheme">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.ProfilPasswordActivity"
        android:label="@string/ubah_password"
        android:theme="@style/AppTheme">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.ChatActivity"
        android:label=""
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.ChatKontakActivity"
        android:label=""
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.ForumActivity"
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity

    android:name="com.unibba.traceralumni2019.activity.ListAlumniActivity"
        android:theme="@style/AppTheme.NoActionBar">
</activity>
<activity
```

```

    android:name="com.unibba.traceralumni2019.activity.ConnectActivity"

    android:theme="@style/AppTheme.NoActionBar.windowFullscreen">
        </activity>
        <activity

            android:name="com.unibba.traceralumni2019.activity.ImageviewActivity"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <service

            android:name="com.unibba.traceralumni2019.util.MyFirebaseMessagingService"
            >
                <intent-filter>
                    <action
                        android:name="com.google.firebaseio.MESSAGING_EVENT"/>
                </intent-filter>
                </service>
                <!--<service-->
                <!--android:name=".util.MyFirebaseInstanceIdService"-->
                <!--<intent-filter>-->
                <!--<action
                    android:name="com.google.firebaseio.INSTANCE_ID_EVENT"/><!--&gt;
                &lt;!--&lt;/intent-filter>-->
                <!--</service>-->
            </application>

        </manifest>

```

5. Tracerstudy1Activity.kt

```

package com.unibba.traceralumni2019.activity

```

```
import android.os.Bundle
import android.text.TextUtils
import android.util.Log
import android.view.View
import android.widget.ArrayAdapter
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.firestore.EventListener
import com.google.firebase.firestore.FirebaseFirestore
import com.google.firebase.firestore.ListenerRegistration
import com.google.firebase.firestore.SetOptions
import com.unibba.traceralumni2019.R
import com.unibba.traceralumni2019.util.Run
import com.unibba.traceralumni2019.util.SessionManager
import kotlinx.android.synthetic.main.activity_tracerstudy1.*

class Tracerstudy1Activity : AppCompatActivity() {

    private val TAG = "Tracerstudy1Activity"
    private var firestoreDB: FirebaseFirestore? = null
    private var firestoreListener: ListenerRegistration? = null
    private lateinit var auth: FirebaseAuth
    private lateinit var sessionManager: SessionManager
    private var nim_s = ""
    private var nama_s = ""
    private var email_s = ""
    private var nohp_s = ""
    private var fakultas_s = ""
    private var prodi_s = ""
    private var angkatan_s = ""
    private var list_fakultas_id = ArrayList<String>()
```

```

private var list_fakultas_nama = ArrayList<String>()
private var list_prodi_id = ArrayList<String>()
private var list_prodi_nama = ArrayList<String>()

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_tracerstudy1)
    firestoreDB = FirebaseFirestore.getInstance()
    auth = FirebaseAuth.getInstance()
    sessionManager = SessionManager(this)
    if (sessionManager.isLoggedIn() == false) {
        nim_s =
        sessionManager.getValidasi().get("nim_valid").toString().toUpperCase()
        nama_s =
        sessionManager.getValidasi().get("nama_valid").toString().toUpperCase()
        email_s =
        sessionManager.getValidasi().get("email_valid").toString()
    } else {
        nim_s =
        sessionManager.getLogin().get("nim").toString().toUpperCase()
        nama_s =
        sessionManager.getLogin().get("nama").toString().toUpperCase()
        email_s = sessionManager.getLogin().get("email").toString()
    }
    selectFakultas()
    loadData()
    btnSimpan.setOnClickListener {
        if (TextUtils.isEmpty(inputNim.text.trim())) {
            inputNim.error = "Nim harus diisi!!!"
            inputNim.requestFocus()
        } else if (TextUtils.isEmpty(inputNama.text.trim())) {
            inputNama.error = "Nama harus diisi!!!"
        }
    }
}

```

```

        inputNama.requestFocus()
    } else if (TextUtils.isEmpty(inputEmail.text.trim())) {
       inputEmail.error = "Email harus diisi!!!"
       inputEmail.requestFocus()
    } else if (!isValidEmail(inputEmail.text.trim())) {
       inputEmail.error = "Email salah!!!"
       inputEmail.requestFocus()
    } else if (TextUtils.isEmpty(inputNohp.text.trim())) {
       inputNohp.error = "No Telepon / Hp salah!!!"
       inputNohp.requestFocus()
    } else if (prodi.selectedItemPosition == 0) {
        Toast.makeText(applicationContext, "Silahkan pilih program studi!!!", Toast.LENGTH_LONG).show()
    } else if (TextUtils.isEmpty(inputAngkatan.text.trim())) {
       inputAngkatan.error = "Angkatan harus diisi!!!"
       inputAngkatan.requestFocus()
    } else if (inputAngkatan.text.trim().length < 4) {
       inputAngkatan.error = "Angkatan salah!!!"
       inputAngkatan.requestFocus()
    } else {
        savedata()
    }
}

fun selectFakultas() {
    list_fakultas_id.clear()
    list_fakultas_nama.clear()
    list_fakultas_id.add("--Pilih--")
    list_fakultas_nama.add("--Pilih--")
    list_prodi_id.add("--Pilih--")
    list_prodi_nama.add("--Pilih--")
    firestoreDB!!
    .collection("fakultas")
}

```

```

    .get()
    .addOnSuccessListener { task ->
        if (task != null) {
            for (doc in task) {
                if (doc.get("prodi") != null) {
                    val prodis = doc.get("prodi") as HashMap<String, Any>
                    for (docs in prodis) {
                        list_fakultas_id.add(doc.id)
                        if (doc.get("nama") != null)
                            list_fakultas_nama.add(doc.get("nama").toString())
                        list_prodi_id.add(docs.key)
                        list_prodi_nama.add(docs.value.toString())
                    }
                }
            }
        }
    }

    // Initializing an ArrayAdapter
    val adapter = ArrayAdapter(
        this, // Context
        android.R.layout.simple_spinner_item, // Layout
        list_prodi_nama // Array
    )

    adapter.setDropDownViewResource(android.R.layout.simple_dropdown_item_1line)
    prodi.adapter = adapter
    adapter.setDropDownViewResource(android.R.layout.simple_dropdown_item_1line)
}

adapter.setDropDownViewResource(android.R.layout.simple_dropdown_it

```

```

em_1line)

    fun loadData() {
        firestoreListener = firestoreDB!!
            .collection("alumni")
            .document(nim_s)
            .addSnapshotListener(EventListener { documentSnapshots, e ->
                if (e != null) {
                    Log.e(TAG, "Listen failed!", e)
                    pbLoading.visibility = View.GONE
                    return@EventListener
                }
                if (documentSnapshots != null) {
                    if (documentSnapshots.get("nama") != null)
                        nama_s = documentSnapshots.get("nama").toString()
                    if (documentSnapshots.get("email") != null)
                        email_s = documentSnapshots.get("email").toString()
                    if (documentSnapshots.get("nohp") != null)
                        nohp_s = documentSnapshots.get("nohp").toString()
                    if (documentSnapshots.get("fakultas") != null)
                        fakultas_s = documentSnapshots.get("fakultas").toString()
                    if (documentSnapshots.get("prodi") != null)
                        prodi_s = documentSnapshots.get("prodi").toString()
                    if (documentSnapshots.get("angkatan") != null)
                        angkatan_s = documentSnapshots.get("angkatan").toString()
                }
                inputNim.setText(nim_s)
                inputNama.setText(nama_s)
               inputEmail.setText(email_s)
                inputNohp.setText(nohp_s)
                setFakultasProdi(prodi_s)
                inputAngkatan.setText(angkatan_s)
                pbLoading.visibility = View.GONE
            })
    }
}

```

```

        })
    }

    fun setFakultasProdi(id:String){
        Run.after(1000, {
            prodi.setSelection(list_prodi_id.indexOf(id))
        })
    }

    fun savedata() {
        val user = auth.currentUser
        if (user != null) {
            user.updateEmail(inputEmail.text.toString())
        }
        var data = HashMap<String, Any>()
        data.put("nim", nim_s)
        data.put("nama", inputNama.text.toString())
        data.put("email", inputEmail.text.toString())
        data.put("nohp", inputNohp.text.toString())
        data.put("fakultas", list_fakultas_id[prodi.selectedItemPosition])
        data.put("prodi", list_prodi_id[prodi.selectedItemPosition])
        data.put("angkatan", inputAngkatan.text.toString().toInt())
        data.put("alamat", inputAlamat.text.toString())
        firestoreDB!!.collection("alumni")
            .document(nim_s)
            .set(data, SetOptions.merge())
            .addOnSuccessListener {}
        var data2 = HashMap<String, Any>()
        data2.put("TS_1", 1)
        data2.put("prodi", prodi.selectedItemPosition)
        data2.put("angkatan", inputAngkatan.text.toString().toInt())
        firestoreDB!!.collection("tracerstudy")
    }
}

```

```

.document(nim_s)
.set(data2, SetOptions.merge())
.addOnSuccessListener {}
onBackPressed()
}

fun isValidEmail(target: CharSequence): Boolean {
    return !TextUtils.isEmpty(target) &&
    android.util.Patterns.EMAIL_ADDRESS.matcher(target).matches()
}
}

```

6. activity_tracerstudy1.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bggradient"
    android:orientation="vertical"
    android:scrollbarAlwaysDrawVerticalTrack="true">

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:background="#fff"
        android:elevation="4dp"
        android:orientation="vertical"
        android:padding="5dp">

```

```
<ScrollView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
<LinearLayout  
    android:id="@+id/linearLayout1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical"  
    android:paddingTop="20dp">  
  
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="5dp"  
    android:textSize="22dp"  
    android:layout_marginBottom="10dp"  
    android:textColor="@color/black"  
    android:gravity="center_horizontal"  
    android:text="Data Pribadi"  
    android:id="@+id/label_tracer"/>  
  
<androidx.cardview.widget.CardView  
    android:layout_width="match_parent"  
    android:layout_margin="5dp"  
    android:layout_height="wrap_content"  
    android:padding="15dp"  
    card_view:cardElevation="2dp"  
    card_view:cardCornerRadius="4dp"  
    android:id="@+id/card_view1">  
  
<LinearLayout  
    android:orientation="vertical"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp">

        <EditText
            android:id="@+id/inputNim"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:drawableTint="#FF4081"
            android:singleLine="true"
            android:hint="Nim"
            android:inputType="text"
            android:enabled="false"/>
    </com.google.android.material.textfield.TextInputLayout>
    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp">

        <EditText
            android:id="@+id/inputNama"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:drawableTint="#FF4081"
            android:singleLine="true"
            android:hint="Nama"
            android:inputType="textCapWords|textCapSentences|textCapCharacters"/>
    </com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">

    <EditText
        android:id="@+id/inputEmail"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:drawableTint="#FF4081"
        android:singleLine="true"
        android:hint="Email"
        android:inputType="textEmailAddress"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">

    <EditText
        android:id="@+id/inputNohp"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:drawableTint="#FF4081"
        android:singleLine="true"
        android:hint="No Telepon / HP"
        android:inputType="phone"/>
</com.google.android.material.textfield.TextInputLayout>

    <TextView
        android:text="Program Studi"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_margin="5dp"/>

<com.toptiche.searchablespinnerlibrary.SearchableSpinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/prodi"
    android:layout_weight="1"/>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">

<EditText
    android:id="@+id/inputAngkatan"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:drawableTint="#FF4081"
    android:singleLine="true"
    android:hint="Angkatan"
    android:inputType="number"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp">

<EditText
    android:id="@+id/inputAlamat"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

```
        android:drawableTint="#FF4081"
        android:singleLine="true"
        android:hint="Alamat"
        android:inputType="text|textMultiLine"/>
    </com.google.android.material.textfield.TextInputLayout>

    </LinearLayout>
</androidx.cardview.widget.CardView>

<Button
    android:id="@+id	btnSimpan"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="40dp"
    android:background="@drawable/shapelogin"
    android:onClick="True"
    android:text="Simpan"
    android:textAllCaps="false"
    android:textColor="#fff"
    android:textSize="18sp"/>
</LinearLayout>
</ScrollView>
<ProgressBar
    android:id="@+id/pbLoading"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_centerInParent="true"
    android:layout_gravity="center"
    style="@style/AppTheme"/>
```

```
</RelativeLayout>
```

```
</RelativeLayout>
```

7. ProfilFragment.kt

```
package com.unibba.traceralumni2019.fragment

import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity.RESULT_OK
import android.content.Context
import android.content.Intent
import android.graphics.Bitmap
import android.graphics.BitmapFactory
import android.location.Location
import android.net.Uri
import android.os.Bundle
import android.provider.MediaStore
import android.provider.Settings
import android.text.Editable
import android.text.InputType
import android.text.TextUtils
import android.text.TextWatcher
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.*
import androidx.annotation.Nullable
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.widget.Toolbar
```

```
import androidx.coordinatorlayout.widget.CoordinatorLayout
import androidx.core.graphics.drawable.RoundedBitmapDrawableFactory
import androidx.fragment.app.Fragment
import com.bumptech.glide.Glide
import com.bumptech.glide.request.RequestOptions
import com.github.babedev.dexter.dsl.runtimePermission
import com.google.android.gms.location.LocationServices
import com.google.android.gms.maps.CameraUpdateFactory
import com.google.android.gms.maps.GoogleMap
import com.google.android.gms.maps.MapView
import com.google.android.gms.maps.MapsInitializer
import com.google.android.gms.maps.model.CameraPosition
import com.google.android.gms.maps.model.LatLng
import com.google.android.gms.maps.model.MarkerOptions
import

com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.android.material.textfield.TextInputEditText
import com.google.android.material.textfield.TextInputLayout
import com.google.firebase.auth.FirebaseAuth
import com.google.firebaseio.firebaseio.*
import com.google.firebaseio.storage.FirebaseStorage
import com.google.firebaseio.storage.StorageReference
import com.toptiche.searchablespinnerlibrary.SearchableSpinner
import com.unibba.traceralumni2019.BuildConfig
import com.unibba.traceralumni2019.R
import com.unibba.traceralumni2019.activity.ImageViewActivity
import com.unibba.traceralumni2019.activity.ProfilPasswordActivity
import com.unibba.traceralumni2019.activity.Tracerstudy0Activity
import com.unibba.traceralumni2019.util.SessionManager
import kotlinx.android.synthetic.main.fragment_profil.*
import java.io.ByteArrayOutputStream
import java.io.IOException
```

```
class ProfilFragment : Fragment() {  
  
    private val TAG = "ProfilFragment"  
    private var firestoreDB: FirebaseFirestore? = null  
    private var storageReference: StorageReference? = null  
    private var firestoreListener: ListenerRegistration? = null  
    private lateinit var auth: FirebaseAuth  
    private lateinit var sessionManager: SessionManager  
    private lateinit var root_layout: CoordinatorLayout  
    private lateinit var rl1: RelativeLayout  
    private lateinit var rl2: RelativeLayout  
    private lateinit var rl3: RelativeLayout  
    private lateinit var rl4: RelativeLayout  
    private lateinit var rl5: RelativeLayout  
    private lateinit var rl6: RelativeLayout  
    private lateinit var toolbar: Toolbar  
    private lateinit var scrollView: ScrollView  
    private lateinit var imageview: ImageView  
    private lateinit var fab: FloatingActionButton  
    private lateinit var pbLoading: ProgressBar  
    private lateinit var textnimlayout: TextInputLayout  
    private lateinit var textnim: TextInputEditText  
    private lateinit var textnamalayout: TextInputLayout  
    private lateinit var textnama: TextInputEditText  
    private lateinit var textprodilayout: TextInputLayout  
    private lateinit var textprodi: TextInputEditText  
    private lateinit var textangkatanlayout: TextInputLayout  
    private lateinit var textangkatan: TextInputEditText  
    private lateinit var textnohplayout: TextInputLayout  
    private lateinit var textnohp: TextInputEditText  
    private lateinit var textemaillayout: TextInputLayout
```

```
private lateinit var textemail: TextInputEditText
private lateinit var textalamatlayout: TextInputLayout
private lateinit var textalamat: TextInputEditText
private lateinit var textlokasilayout: TextInputLayout
private lateinit var textlokasi: TextInputEditText
private lateinit var textnamainstansilayout: TextInputLayout
private lateinit var textnamainstansi: TextInputEditText
private lateinit var textalamatinstansilayout: TextInputLayout
private lateinit var textalamatinstansi: TextInputEditText
private lateinit var textnotlpinstansilayout: TextInputLayout
private lateinit var textnotlpinstansi: TextInputEditText
private lateinit var textjabatanlayout: TextInputLayout
private lateinit var textjabatan: TextInputEditText
private val GALLERY = 1
private val CAMERA = 2
var nim_s = ""
var akses_s = ""
var image_url = ""
var nama = "-"
var alamat = "-"
var email = "-"
var nohp = "-"
var lat = "-"
var long = "-"
var prodi_id = "-"
var angkatan = "-"
var s1 = "-"
var s2 = "-"
var s3 = "-"
var s4 = "-"
var fkl = HashMap<String, Any>()
```

```

@Nullable
override fun onCreateView(
    inflater: LayoutInflater,
    container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    val rootView: View = inflater.inflate(R.layout.fragment_profil,
    container, false)
    sessionManager = SessionManager(context!!)
    nim_s = sessionManager.getLogin().get("nim").toString()
    akses_s = sessionManager.getLogin().get("akses").toString()
    firestoreDB = FirebaseFirestore.getInstance()
    auth = FirebaseAuth.getInstance()
    storageReference = FirebaseStorage.getInstance().reference
    sessionManager.setImage("")
    rl1 = rootView.findViewById(R.id.rl1)
    rl2 = rootView.findViewById(R.id.rl2)
    rl3 = rootView.findViewById(R.id.rl3)
    rl4 = rootView.findViewById(R.id.rl4)
    rl5 = rootView.findViewById(R.id.rl5)
    rl6 = rootView.findViewById(R.id.rl6)
    root_layout = rootView.findViewById(R.id.root_layout)
    scrollView = rootView.findViewById(R.id.scrollView)
    toolbar = rootView.findViewById(R.id.toolbar)
    toolbar.title = "Profil"
    imageview = rootView.findViewById(R.id.imageView)
    fab = rootView.findViewById(R.id.fab)
    pbLoading = rootView.findViewById(R.id.pbLoading)
    textnimlayout = rootView.findViewById(R.id.textNimLayout)
    textnim = rootView.findViewById(R.id.textNim)
    textnamalayout = rootView.findViewById(R.id.textNamaLayout)
    textnama = rootView.findViewById(R.id.textNama)
}

```

```

textprodilayout = rootView.findViewById(R.id.textProdiLayout)
textprodi = rootView.findViewById(R.id.textProdi)
textangkatanlayout = =
rootView.findViewById(R.id.textAngkatanLayout)
textangkatan = rootView.findViewById(R.id.textAngkatan)
textnohplayout = rootView.findViewById(R.id.textNohpLayout)
textnohp = rootView.findViewById(R.id.textNohp)
textemaillayout = rootView.findViewById(R.id.textEmailLayout)
textemail = rootView.findViewById(R.id.textEmail)
textalamatlayout = rootView.findViewById(R.id.textAlamatLayout)
textalamat = rootView.findViewById(R.id.textAlamat)
textlokasilayout = rootView.findViewById(R.id.textLokasiLayout)
textlokasi = rootView.findViewById(R.id.textLokasi)
textnamainstansilayout = =
rootView.findViewById(R.id.textNamainstansiLayout)
textnamainstansi = rootView.findViewById(R.id.textNamainstansi)
textalamatinstansilayout = =
rootView.findViewById(R.id.textAlamatinstansiLayout)
textalamatinstansi = rootView.findViewById(R.id.textAlamatinstansi)
textnotlpinstansilayout = =
rootView.findViewById(R.id.textNotlpinstansiLayout)
textnotlpinstansi = rootView.findViewById(R.id.textNotlpinstansi)
textjabatanlayout = rootView.findViewById(R.id.textJabatanLayout)
textjabatan = rootView.findViewById(R.id.textJabatan)
(activity as AppCompatActivity).setSupportActionBar(toolbar)
setHasOptionsMenu(true)
scrollView.scrollTo(0, 0)
fab.setOnClickListener {
    showPictureDialog()
}
imageview.setOnClickListener {
    if (!TextUtils.isEmpty(image_url)) {

```

```

        val intent = Intent(context!!, ImageviewActivity::class.java)
        intent.putExtra("image_url", image_url)
        this.startActivity(intent)
    }

}

rl1.setOnClickListener {
    val intent = Intent(context!!, Tracerstudy0Activity::class.java)
    intent.putExtra("edit", true)
    context!!.startActivity(intent)
}

rl2.setOnClickListener {
    var lat = 0.0
    var long = 0.0
    if (!textLokasi.text.toString().isEmpty()) {
        val text = textLokasi.text.toString().trim().split(",")
        if (!text[0].equals("-") && !text[1].equals("-")) {
            lat = text[0].toDouble()
            long = text[1].toDouble()
        }
    }
    dialogLokasi("'" + textLokasi.text.toString().trim(), "'" + nim_s, "", 0
+ lat, 0 + long)
}

rl3.setOnClickListener {
    dialog_about()
}

rl4.setOnClickListener {
    dialog_rateus()
}

rl5.setOnClickListener {
    val intent = Intent(context!!, ProfilPasswordActivity::class.java)
}

```

```
        context!!.startActivity(intent)
    }

    rl6.setOnClickListener {
        if (firestoreListener != null)
            firestoreListener!!.remove()
        firestoreDB!!.terminate()
        sessionManager.logoutUser()
        auth.signOut()
    }

    pbLoading.visibility = View.VISIBLE
    loadData()
    return rootView
}

override fun onDestroy() {
    super.onDestroy()
    if (firestoreListener != null) {
        firestoreListener!!.remove()
        firestoreListener = null
    }
}

fun fakultas() {
    firestoreDB!!.collection("fakultas")
        .addSnapshotListener(EventListener { documentSnapshots, e ->
            if (e != null) {
                Log.e(TAG, "Listen failed!", e)
                pbLoading.visibility = View.GONE
                return@EventListener
            }
        })
}
```

```

        if (documentSnapshots != null) {
            for (doc in documentSnapshots) {
                if (doc.get("prodi") != null){
                    val pvalue = doc.get("prodi") as HashMap<String, Any>
                    for (p in pvalue) {
                        fkl.put(p.key, p.value)
                    }
                }
                System.out.println("fkl1=>" + fkl)
            }
        })
    }

    fun loadData() {
        var Fq: Query
        if (akses_s.equals("1")) {
            Fq = firestoreDB!!.collection("users")
            textprodilayout.visibility = View.GONE
            textangkatanlayout.visibility = View.GONE
            textnohlayout.visibility = View.GONE
            textalamatlayout.visibility = View.GONE
            textlokasilayout.visibility = View.GONE
            textnamainstansilayout.visibility = View.GONE
            textalamatinstansilayout.visibility = View.GONE
            textnotlpinstansilayout.visibility = View.GONE
            textjabatanlayout.visibility = View.GONE
            rl2.visibility = View.GONE
        } else {
            Fq = firestoreDB!!.collection("alumni")
        }
        firestoreListener = Fq.document(nim_s)
    }
}

```

```

.addSnapshotListener(EventListener { documentSnapshots, e ->
    if (e != null) {
        Log.e(TAG, "Listen failed!", e)
        pbLoading.visibility = View.GONE
        return@EventListener
    }

    if (documentSnapshots != null) {
        val document = documentSnapshots.data
        if (document?.get("image_url") != null)
            image_url = document.get("image_url").toString()
        if (document?.get("nama") != null)
            nama = document.get("nama").toString()
        if (document?.get("alamat") != null)
            alamat = document.get("alamat").toString()
        if (document?.get("email") != null)
            email = document.get("email").toString()
        if (document?.get("nohp") != null)
            nohp = document.get("nohp").toString()
        if (document?.get("latitude") != null)
            lat = document.get("latitude").toString()
        if (document?.get("longitude") != null)
            long = document.get("longitude").toString()
        if (document?.get("prodi") != null)
            prodi_id = document.get("prodi").toString()
        if (document?.get("angkatan") != null)
            angkatan = document.get("angkatan").toString()
    }

    context?.let{
        val placeholder = BitmapFactory.decodeResource(it.resources,
R.drawable.ic_user)
    }
}

```

```

    val circularBitmapDrawable = 
RoundedBitmapDrawableFactory.create(it.resources, placeholder)
    circularBitmapDrawable.isCircular = true
    Glide.with(it).load(image_url)
        .apply(
            RequestOptions()
                .circleCrop()
                .centerCrop()
                .error(circularBitmapDrawable)
                .placeholder(circularBitmapDrawable)
        )
        .apply(RequestOptions.circleCropTransform())
        .into(imageView)
    }

editDisable(textnim)
editDisable(textnama)
editDisable(textprodi)
editDisable(textangkatan)
editDisable(textnohp)
editDisable(textemail)
editDisable(textalamat)
editDisable(textlokasi)
editDisable(textnamainstansi)
editDisable(textalamatinstansi)
editDisable(textnotlpinstansi)
editDisable(textjabatan)

textnim.setText(nim_s)
textnama.setText(nome)
if (prodi_id.equals("") || prodi_id.equals("-")) {
    textprodi.setText("-")
} else{

```

```

        textprodi.setText(fkl.get(prodi_id).toString())
    }

    textangkatan.setText(angkatan)
    textnohp.setText(nohp)
    textemail.setText(email)
    textalamat.setText(alamat)
    textlokasi.setText(lat + ", " + long)
    pbLoading.visibility = View.GONE
}

firestoreListener = firestoreDB!!.collection("tracerstudy")
.document(nim_s)
.addSnapshotListener(EventListener { documentSnapshots, e ->
    if (e != null) {
        Log.e(TAG, "Listen failed!", e)
        pbLoading.visibility = View.GONE
        return@EventListener
    }
    if (documentSnapshots != null) {
        val document = documentSnapshots.data
        if (document?.get("F11_6") != null)
            s1 = document.get("F11_6").toString()
        if (document?.get("F11_7") != null)
            s2 = document.get("F11_7").toString()
        if (document?.get("F11_8") != null)
            s3 = document.get("F11_8").toString()
        if (document?.get("F11_9") != null)
            s4 = document.get("F11_9").toString()
    }
    textnamainstansi.setText(s1)
    textalamatinstansi.setText(s2)
    textnotlpinstansi.setText(s3)
}

```

```

        textjabatan.setText(s4)
        pbLoading.visibility = View.GONE
    })
}

fun dialog_about() {
    val inflater = context!!.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as
    LayoutInflator
    val formsView = inflater.inflate(R.layout.dialog_about, null, false)
    var btnhelp: Button
    var textclose: TextView
    var textversion: TextView
    btnhelp = formsView.findViewById(R.id.btnHelp)
    textclose = formsView.findViewById(R.id.textClose)
    textversion = formsView.findViewById(R.id.textVersion)
    textversion.text = "Version " + BuildConfig.VERSION_NAME
    val builder = AlertDialog.Builder(context!!).create()
    // builder.setTitle("Pengelolaan User")
    builder.setView(formsView)
    btnhelp.setOnClickListener {
        val uri = Uri.parse("http://fti.unibba.ac.id/")
        val intent = Intent(Intent.ACTION_VIEW, uri)
        startActivity(intent)
        builder.dismiss()
    }
    textclose.setOnClickListener {
        builder.dismiss()
    }
    builder.show()
}

```

```

fun dialog_rateus() {
    startActivity(Intent(Intent.ACTION_VIEW,
Uri.parse("market://details?id=" + BuildConfig.APPLICATION_ID)))
}

private fun dialogEdit(str:Boolean, title:String, key:String){
    val data = HashMap<String, Any>()
    val inflater = activity!!.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as
LayoutInflator
    val formsView = inflater.inflate(R.layout.dialog_profil_lokasi, null,
false)

    var textInputHint: TextInputLayout
    var textInput: TextInputEditText
    var fakultas: SearchableSpinner
    var prodi: SearchableSpinner

    textInputHint = formsView.findViewById(R.id.textInputLayout)
    textInput = formsView.findViewById(R.id.textInput)
    fakultas = formsView.findViewById(R.id.fakultas)
    prodi = formsView.findViewById(R.id.prodi)

    textInputHint.hint = title

    val builder = AlertDialog.Builder(context!!)
    builder.setTitle("Edit " + title)
    builder.setView(formsView)
    builder.setPositiveButton("Kembali", { dialog, which -> })
    builder.setNegativeButton("Simpan", { dialog, which ->
        if (str == true) {
            val text = textInput.text.toString().trim()
    
```

```

        data.put(key, text)
    }else{
        data.put("fakultas", fakultas.selectedItemPosition)
        data.put("prodi", prodi.selectedItemPosition)
    }
    savedata(data)
})
val dialog = builder.create()
builder.show()
}

private fun dialogLokasi(content: String, nim: String, nama: String, lat: Double, long: Double) {
    val inflater = activity!!.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflator
    val formsView = inflater.inflate(R.layout.dialog_profil_lokasi, null, false)
    var mapView: MapView
    var googleMap: GoogleMap
    var textInputHint: TextInputLayout
    var textInput: TextInputEditText
    var btnsetlokasi: Button
    var title = ""
    var dataprofil = HashMap<String, Any>()
    mapView = formsView.findViewById(R.id.mapView)
    textInputHint = formsView.findViewById(R.id.textInputLayout)
    textInput = formsView.findViewById(R.id.textInput)
    btnsetlokasi = formsView.findViewById(R.id.btnSetLokasi)
    MapsInitializer.initialize(context!!)
    title = "Update Lokasi (Latitude & Longitude)"
    textInputHint.hint = "Lokasi (Latitude & Longitude)"
}

```

```

if (!content.equals("-"))
    textView.setText(content)

textView.inputType = InputType.TYPE_NULL
textView.setTextIsSelectable(true)
textView.keyListener = null
btnsetlokasi.setOnClickListener {
    getlokasi(textView)
    textView.addTextChangedListener(object : TextWatcher {
        override fun afterTextChanged(s: Editable?) {
            val text = textView.text.toString().trim()
            if (!text.equals("")) {
                val k1 = text.split(",")[0].toDouble()
                val k2 = text.split(",")[1].toDouble()
                mapView.getMapAsync { mMap ->
                    googleMap = mMap
                    googleMap.uiSettings.isCompassEnabled = true
                    googleMap.uiSettings.isMyLocationButtonEnabled = true
                    googleMap.uiSettings.isRotateGesturesEnabled = true
                    // For dropping a marker at a point on the Map
                    googleMap.clear()
                    var koordinat: LatLng
                    koordinat = LatLng(k1, k2)
                    val cameraPosition = CameraPosition.Builder().target(koordinat).zoom(7f).build()

                    googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition))
                    googleMap.addMarker(MarkerOptions().position(koordinat).title(nim).snippet(name))
                    // For zooming automatically to the location of the marker
                    googleMap.setOnInfoWindowClickListener { marker -> }
                }
            }
        }
    })
}

```

```

        }

    }

}

override fun beforeTextChanged(s: CharSequence?, start: Int,
count: Int, after: Int) {

}

override fun onTextChanged(s: CharSequence?, start: Int, before:
Int, count: Int) {

}

})

}

val builder = AlertDialog.Builder(context!!)
builder.setTitle(title)
builder.setView(formsView)
builder.setPositiveButton("Kembali", { dialog, which -> })
builder.setNegativeButton("Simpan", { dialog, which ->

    val text =textInput.text.toString().trim()
    if (!text.equals("")) {
        val k1 = text.split(",")[0].toDouble()
        val k2 = text.split(",")[1].toDouble()
        dataprofil.put("latitude", k1.toString())
        dataprofil.put("longitude", k2.toString())
        Log.d(TAG, dataprofil.toString())
    }
    savedata(dataprofil)
})
val dialog = builder.create()
builder.show()

```

```

mapView.onCreate(dialog.onSaveInstanceState())
mapView.onResume()
mapView.getMapAsync { mMap ->
    googleMap = mMap
    googleMap.uiSettings.isCompassEnabled = true
    googleMap.uiSettings.isMyLocationButtonEnabled = true
    googleMap.uiSettings.isRotateGesturesEnabled = true
    // For dropping a marker at a point on the Map
    var koordinat: LatLng
    koordinat = LatLng(lat, long)
    val cameraPosition = CameraPosition.Builder().target(koordinat).zoom(7f).build()

    googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition))

    googleMap.addMarker(MarkerOptions().position(koordinat).title(nim).snippet(nama))
        googleMap.setMinZoomPreference(4F)
        // For zooming automatically to the location of the marker
        googleMap.setOnInfoWindowClickListener { marker -> }
    }

    fun savedata(data: HashMap<String, Any>) {
        var user = data
        user.remove("fakultas")
        user.remove("prodi")
        user.remove("angkatan")
        user.remove("alamat")
        user.remove("latitude")
        user.remove("longitude")
    }
}

```

```

if (!user.isEmpty) {
    firestoreDB!.collection("users")
        .document(nim_s)
        .set(user, SetOptions.merge())
        .addOnCompleteListener {
            Toast.makeText(context, "Simpan data berhasil...", Toast.LENGTH_LONG).show()
        }
}

if (akses_s.equals("0")) {
    firestoreDB!.collection("alumni")
        .document(nim_s)
        .set(data, SetOptions.merge())
        .addOnCompleteListener {}
}

}

private fun showPictureDialog() {
    val pictureDialog = AlertDialog.Builder(context!!)
    pictureDialog.setTitle("Unggah Foto")
    val pictureDialogItems = arrayOf("Ambil gambar", "Pilih dari galeri",
    "Batal")
    pictureDialog.setItems(
        pictureDialogItems
    ) { dialog, which ->
        when (which) {
            0 -> takePhotoFromCamera()
            1 -> choosePhotoFromGallary()
            2 -> dialog.dismiss()
        }
    }
}

```

```
        pictureDialog.show()
    }

    fun choosePhotoFromGallary() {
        runtimePermission {
            permissions(Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE) {
                checked {
                    val galleryIntent = Intent(
                        Intent.ACTION_PICK,
                        MediaStore.Images.Media.EXTERNAL_CONTENT_URI
                    )
                    startActivityForResult(galleryIntent, GALLERY)
                }
            }
        }
    }

    private fun takePhotoFromCamera() {
        runtimePermission {
            permissions(Manifest.permission.CAMERA) {
                checked {
                    val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
                    startActivityForResult(intent, CAMERA)
                }
            }
        }
    }
}
```

```

}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {

    super.onActivityResult(requestCode, resultCode, data)

    if (requestCode == GALLERY && resultCode == RESULT_OK &&
        data != null && data.data != null) {
        val contentURI = data.data
        try {
            val bitmap: Bitmap? =
                MediaStore.Images.Media.getBitmap(activity!!.contentResolver, contentURI)
            uploadimage(bitmap!!)
            context?.let {
                Glide.with(it).load(bitmap)
                    .apply(
                        RequestOptions()
                            .circleCrop()
                            .centerCrop()
                            .error(R.drawable.ic_user)
                            .placeholder(R.drawable.ic_user)
                    )
                    .apply(RequestOptions.circleCropTransform())
                    .into(imageview)
            }
        } catch (e: IOException) {
            e.printStackTrace()
            Toast.makeText(context!!, "Failed!", Toast.LENGTH_SHORT).show()
        }
    }
}

```

```

} else if (requestCode == CAMERA && resultCode == RESULT_OK
&& data != null) {
    val bitmap: Bitmap? = data.extras!![ "data" ] as Bitmap
    uploadimage(bitmap!!)
    context?.let{
        Glide.with(it).load(bitmap)
            .apply(
                RequestOptions()
                    .circleCrop()
                    .centerCrop()
                    .error(R.drawable.ic_user)
                    .placeholder(R.drawable.ic_user)
            )
            .apply(RequestOptions.circleCropTransform())
            .into(imageview)
    }
}
}

fun uploadimage(bitmap: Bitmap) {
    if (bitmap != null) {
        val baos = ByteArrayOutputStream()
        bitmap.compress(Bitmap.CompressFormat.JPEG, 50, baos)
        val data: ByteArray = baos.toByteArray()
        val ref = storageReference!!.child("images/users/" + nim_s + ".jpg")
        val uploadTask = ref.putBytes(data)
        uploadTask.addOnSuccessListener { task ->
            ref.downloadUrl
                .addOnSuccessListener { uri ->
                    saveimageurl(uri.toString())
                }
        }
    }
}

```

```

        }

    }

    fun saveimageurl(url: String) {
        val data = HashMap<String, Any>()
        data.put("image_url", url)
        if (akses_s.equals("0")) {
            firestoreDB!!.collection("alumni")
                .document(nim_s)
                .set(data, SetOptions.merge())
        }
        firestoreDB!!.collection("users")
            .document(nim_s)
            .set(data, SetOptions.merge())
    }

    companion object {
        private val IMAGE_DIRECTORY = "/tracer_unibba"
    }

    @SuppressLint("MissingPermission")
    fun getlokasi(inputText: EditText) {
        runtimePermission {
            permission(Manifest.permission.ACCESS_FINE_LOCATION) {
                granted {
                    val mFusedLocation = LocationServices.getFusedLocationProviderClient(context!!)
                    mFusedLocation!!.lastLocation.addOnSuccessListener {
                        location ->
                        onSuccess(location, inputText)
                    }
                }
            }
        }
    }
}

```

```

denied {
    Toast.makeText(context!!, "Permission denied",
Toast.LENGTH_LONG).show()
    openSettings()
}
}

}

fun onSuccess(location: Location?, inputText: EditText) {
    // Do it all with location
    Log.d(
        "My Current location",
        "Lat : ${location?.latitude} Long : ${location?.longitude}"
    )
    if (location != null) {
        inputText.setText(location.latitude.toString() + ", " +
location.longitude.toString())
    } else {
        Toast.makeText(
            context!!,
            "Set lokasi gagal, periksa kembali pengaturan handphone!!!",
            Toast.LENGTH_LONG
        ).show()
    }
}

fun openSettings() {
    val intent = Intent()
    intent.action =
    Settings.ACTION_APPLICATION_DETAILS_SETTINGS
    val uri = Uri.fromParts(

```

```
    "package",
    BuildConfig.APPLICATION_ID, null
)
intent.data = uri
intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK
startActivity(intent)
}

fun editDisable(edit: TextInputEditText) {
    edit.inputType = InputType.TYPE_NULL
    edit.keyListener = null
    edit.setTextIsSelectable(true)
}
}
```

8. Fragment_profil.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="60dp"
    android:id="@+id/root_layout"
    android:focusable="true"
    android:focusableInTouchMode="true">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBarLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```



```
        android:layout_height="match_parent"
        android:text="Kontak Saya"
        android:textSize="16sp"
        android:layout_margin="5dp"
        android:textColor="@color/colorPrimary"
        android:textStyle="bold"/>

<androidx.coordinatorlayout.widget.CoordinatorLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center"
        android:scaleType="fitXY"/>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|right"
        android:layout_margin="5dp"
        android:clickable="true"
        android:src="@drawable/ic_camera_alt_white_24dp"
        app:backgroundTint="@color/blue"
        app:fabSize="mini"/>
```

```
        app:elevation="5dp"
        app:borderWidth="0dp"
        app:layout_anchorGravity="end|bottom"
        app:layout_anchor="@+id/imageView"/>

    </androidx.coordinatorlayout.widget.CoordinatorLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:id="@+id/textNimLayout">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Nim"
            android:id="@+id/textNim"/>
    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:id="@+id/textNamaLayout">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Nama Lengkap"
            android:id="@+id/textNama"/>
    </com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dp"
    android:id="@+id/textProdiLayout">

<com.google.android.material.textfield.TextInputEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Program Studi"
    android:id="@+id/textProdi"/>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dp"
    android:id="@+id/textAngkatanLayout">

<com.google.android.material.textfield.TextInputEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Angkatan"
    android:id="@+id/textAngkatan"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dp"
    android:id="@+id/textNohpLayout">
```

```
<com.google.android.material.textfield.TextInputEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="No HP"
    android:id="@+id/textNohp"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dp"
    android:id="@+id/textEmailLayout">

<com.google.android.material.textfield.TextInputEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:id="@+id/textEmail"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dp"
    android:id="@+id/textAlamatLayout">

<com.google.android.material.textfield.TextInputEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Alamat"
    android:id="@+id/textAlamat"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:id="@+id/textLokasiLayout">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Lokasi"
        android:id="@+id/textLokasi"/>
    </com.google.android.material.textfield.TextInputLayout>
    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:id="@+id/textNamainstansiLayout">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Nama Instansi (*Bagi yang sudah
bekerja)"
            android:id="@+id/textNamainstansi"/>
        </com.google.android.material.textfield.TextInputLayout>
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="5dp"
            android:id="@+id/textAlamatinstansiLayout">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
        android:hint="Alamat Instansi (*Bagi yang sudah  
bekerja)"  
        android:id="@+id/textAlamatinstansi"/>  
</com.google.android.material.textfield.TextInputLayout>  
<com.google.android.material.textfield.TextInputLayout  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:layout_margin="5dp"  
        android:id="@+id/textNotlpinstansiLayout">  
  
<com.google.android.material.textfield.TextInputEditText  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:hint="No Tlp Instansi (*Bagi yang sudah  
bekerja)"  
        android:id="@+id/textNotlpinstansi"/>  
</com.google.android.material.textfield.TextInputLayout>  
<com.google.android.material.textfield.TextInputLayout  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:layout_margin="5dp"  
        android:id="@+id/textJabatanLayout">  
  
<com.google.android.material.textfield.TextInputEditText  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:hint="Jabatan (*Bagi yang sudah bekerja)"  
        android:id="@+id/textJabatan"/>  
</com.google.android.material.textfield.TextInputLayout>  
</LinearLayout>  
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
    android:id="@+id/card_view2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="Lainnya"
            android:textSize="16sp"
            android:layout_margin="5dp"
            android:textColor="@color/colorPrimary"
            android:textStyle="bold"/>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:clickable="true"
            android:layout_margin="5dp"
            android:background="?attr/selectableItemBackground"
            android:id="@+id/r11">
            <ImageView
                android:id="@+id/imageView1"
                android:layout_width="24dp"
                android:layout_height="24dp"
                android:layout_centerVertical="true"
                android:layout_alignParentLeft="true"
                android:src="@drawable/ic_edit_grey_24dp"
```

```
    android:layout_marginLeft="5dp"/>

    <TextView
        android:text="Update Data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:textColor="@color/black"
        android:textSize="16dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_toRightOf="@+id/imageView1"
        android:layout_toEndOf="@+id/imageView1"/>

    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"

        android:src="@drawable/ic_keyboard_arrow_right_grey_24dp"
        android:layout_marginLeft="5dp"/>

    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@color/gray"/>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
```

```
    android:clickable="true"
    android:layout_margin="5dp"
    android:background="?attr/selectableItemBackground"
    android:id="@+id/r12">
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="24dp"
    android:layout_height="24dp"
    android:layout_centerVertical="true"
    android:layout_alignParentLeft="true"
    android:src="@drawable/lokasi"
    android:layout_marginLeft="5dp"/>

<TextView
    android:text="Update Lokasi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:textColor="@color/black"
    android:textSize="16dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="5dp"
    android:layout_toRightOf="@+id/imageView2"
    android:layout_toEndOf="@+id/imageView2"/>
<ImageView
    android:layout_width="32dp"
    android:layout_height="32dp"
    android:layout_centerVertical="true"
    android:layout_alignParentRight="true"
```

```
    android:src="@drawable/ic_keyboard_arrow_right_grey_24dp"
        android:layout_marginLeft="5dp"/>

    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@color/gray"/>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:clickable="true"
        android:layout_margin="5dp"
        android:background="?attr/selectableItemBackground"
        android:id="@+id/rl3">
        <ImageView
            android:id="@+id/imageView3"
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:layout_centerVertical="true"
            android:layout_alignParentLeft="true"
            android:src="@drawable/ic_help_grey_24dp"
            android:layout_marginLeft="5dp"/>

        <TextView
            android:text="About"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:textColor="@color/black"
```

```
        android:textSize="16dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="5dp"
        android:layout_toRightOf="@+id/imageView3"
        android:layout_toEndOf="@+id/imageView3"/>
    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"

        android:src="@drawable/ic_keyboard_arrow_right_grey_24dp"
        android:layout_marginLeft="5dp"/>

    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@color/gray"/>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:clickable="true"
        android:layout_margin="5dp"
        android:background="?attr/selectableItemBackground"
        android:id="@+id/rl4">
        <ImageView
            android:id="@+id/imageView4"
            android:layout_width="24dp"
            android:layout_height="24dp"
```

```
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:src="@drawable/ic_star_grey_24dp"
        android:layout_marginLeft="5dp"/>

    <TextView
        android:text="Rate this app"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:textColor="@color/black"
        android:textSize="16dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="5dp"
        android:layout_toRightOf="@+id/imageView4"
        android:layout_toEndOf="@+id/imageView4"/>

    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"
        android:src="@drawable/ic_keyboard_arrow_right_grey_24dp"
        android:layout_marginLeft="5dp"/>

</RelativeLayout>
<View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@color/gray"/>
```

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:clickable="true"
    android:layout_margin="5dp"
    android:background="?attr/selectableItemBackground"
    android:id="@+id/rl5">

    <ImageView
        android:id="@+id/imageView5"
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:src="@drawable/ic_vpn_key_grey_24dp"
        android:layout_marginLeft="5dp"/>

    <TextView
        android:text="Ubah Password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:textColor="@color/black"
        android:textSize="16dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="5dp"
        android:layout_toRightOf="@+id/imageView5"
        android:layout_toEndOf="@+id/imageView5"/>

    <ImageView
        android:layout_width="32dp"
```

```
        android:layout_height="32dp"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"

    android:src="@drawable/ic_keyboard_arrow_right_grey_24dp"
        android:layout_marginLeft="5dp"/>
</RelativeLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="@color/gray"/>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:clickable="true"
    android:layout_margin="5dp"
    android:background="?attr/selectableItemBackground"
    android:id="@+id/r16">
<ImageView
    android:id="@+id/imageView6"
    android:layout_width="24dp"
    android:layout_height="24dp"
    android:layout_centerVertical="true"
    android:layout_alignParentLeft="true"
    android:src="@drawable/ic_exit_to_app_grey_24dp"
    android:layout_marginLeft="5dp"/>
<TextView
    android:text="Logout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
```

```
        android:layout_centerVertical="true"
        android:textColor="@color/black"
        android:textSize="16dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="5dp"
        android:layout_toRightOf="@+id/imageView6"
        android:layout_toEndOf="@+id/imageView6"/>
    <ImageView
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"
        android:src="@drawable/ic_keyboard_arrow_right_grey_24dp"
        android:layout_marginLeft="5dp"/>
    </RelativeLayout>
    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@color/gray"/>
    </LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
</ScrollView>
<ProgressBar
        android:id="@+id/pbLoading"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_centerInParent="true"
        android:layout_gravity="center"
        style="@style/AppTheme"/>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Lampiran 4 Kartu Bimbingan

PERBAIKAN PENELITIAN / SKRIPSI PEMBIMBING 1

| TANGGAL BIMBINGAN | BAB / BAGIAN YANG HARUS DIPERBAIKI | PARAF PEMBIMBING |
|-------------------|---|---|
| 29/2009 /April | §452.1. Pembahasan lansiran tentara harus sesuai pedoman |  |
| 4/2009 /Mei | §1. Lembar pertidaksengajaan sesua. |  |
| 8/2009 /Mei | Uraikan tentang penelitian dan jurnal penelitian |  |

| TANGGAL | BAB, BAGIAN YANG HARUS DIPERBAIKI | PARAF PEMBIMBING |
|----------------|--|------------------|
| 29 April | latar belakang penulis referensi masih banyak | ✓ |
| 11 May | perbaiki kelimana wasalaq dan hubungannya | ✓ |
| 18 May | data kurang lengkap grafe keracau, seharusnya pranagraf, laporan temuan | ✓ |
| 29 May | penulisan nomenklatur dapat diperbaiki | ✓ |
| 15 June | pembuktian teks ke dalam jurnal penulis formal pertama metode penelitian | ✓ |
| 4 July | rapor corak | ✓ |
| 6 August | penulisan kesimpulan dalam kesimpulan | ✓ |
| 13 August | ✓ | ✓ |
| 20 August | ✓ | ✓ |
| 23 August 2019 | perbaiki model yang suarce, (frontpage, abstract) | ✓ |

Lampiran 5 Biodata Penulis



Penulis bernama Deni Hidayat, dilahirkan pada tanggal 16 Desember 1993 di Bandung. Penulis adalah seorang anak dari pasangan suami istri yang bernama Bapak Andi Sukandi dan Ibu Omih Susilawati. Pada tahun 2004 penulis lulus Sekolah Dasar di SDN 1 Sukamanah, pada tahun 2007 penulis lulus Sekolah Menengah Pertama di SMPN 1 Paseh, pada tahun 2010 penulis lulus Sekolah Menengah Kejuruan di SMK KP 1 Majalaya dan pada tahun 2015 penulis masuk di UNIVERSITAS BALE BANDUNG (UNIBBA) yang insa aloh pada tahun ini dapat menyelesaikan dan mengantarkan penulis untuk mendapatkan gelar serjana strata satu. Demikian Biodata Penulis untuk sekedar diketahui.

Terima Kasih

Deni Hidayat