

**PENGEMBANGAN APLIKASI MOBILE I'M UII DENGAN
FRAMEWORK FLUTTER DAN LUMEN PADA FITUR
REMINDER**



Disusun Oleh:

N a m a : Rio Risqi Akbar Herlambang

NIM : 19523031

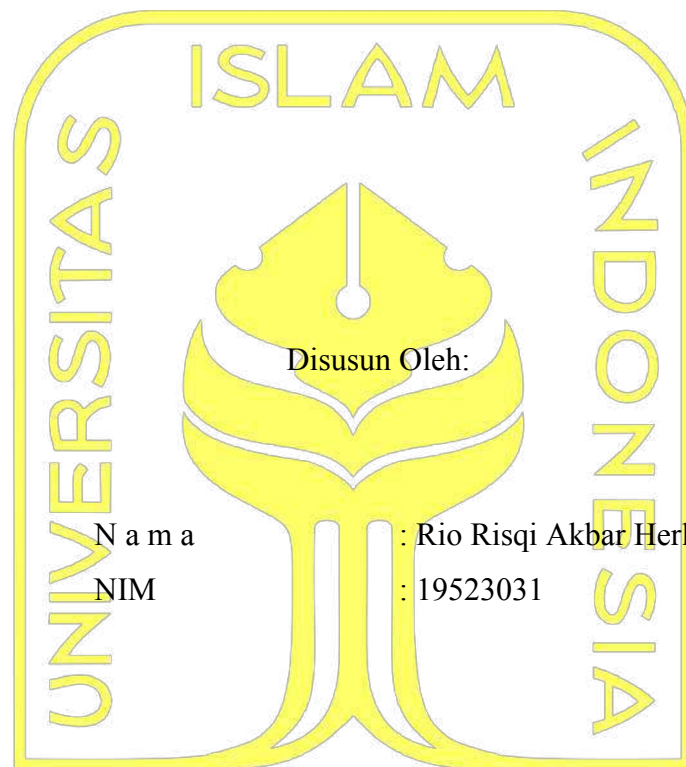
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2023

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN APLIKASI MOBILE I'M UII DENGAN
FRAMEWORK FLUTTER DAN LUMEN PADA FITUR
REMINDER**

TUGAS AKHIR JALUR MAGANG



N a m a

: Rio Risqi Akbar Herlambang

NIM

: 19523031

الجامعة الإسلامية
يogyakarta, 6 Juli 2023

Pembimbing,

A handwritten signature in blue ink, which appears to be 'Zainudin Zukhri', is written over the text 'Pembimbing,'.

(Zainudin Zukhri, S.T, M.IT)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN APLIKASI MOBILE I'M UII DENGAN
FRAMEWORK FLUTTER DAN LUMEN PADA FITUR
REMINDER**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 31 Juli 2023

Tim Penguji

Zainudin Zukhri, S.T, M.IT.

Anggota 1

Chandra Kusuma Dewa, S.Kom., M.Cs.,
Ph.D.

Anggota 2

Dr. Sri Kusumadewi, S.Si., M.T.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(DThomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Rio Risqi Akbar Herlambang

NIM : 19523031

Tugas akhir dengan judul:

PENGEMBANGAN APLIKASI MOBILE I'M UII DENGAN FRAMEWORK FLUTTER DAN LUMEN PADA FITUR REMINDER

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 6 Juli 2023



(Rio Risqi Akbar Herlambang)

HALAMAN PERSEMBAHAN

Dalam kesempatan ini, penulis dengan tulus ingin menyampaikan rasa syukur dan penghargaan yang mendalam kepada semua pihak yang telah memberikan dukungan, bimbingan, dan kontribusi dalam penyelesaian tugas akhir ini. Adapun pihak-pihak tersebut yaitu:

1. Allah SWT karena atas rahmat, hidayah, dan karunia-Nya yang selalu melimpah dalam setiap langkah perjalanan penulisan tugas akhir ini.
2. Keluarga tercinta, terutama kedua orang tua, yang selalu memberikan doa, dukungan moral, dan semangat dalam perjalanan pendidikan penulis. Kebersamaan, kehangatan, dan cinta dari keluarga menjadi motivasi tersendiri dalam menyelesaikan tugas akhir ini.

HALAMAN MOTO

“Ketahuilah bahwa kemenangan bersama kesabaran, kelapangan bersama kesempitan, dan kesulitan bersama kemudahan.”

(HR Tirmidzi)

“Dan Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatu apa pun, dan Dia memberimu pendengaran, penglihatan, dan hati agar kamu bersyukur.”

(Q.S An-Nahl: 78)

“Kualitas hidup sesungguhnya adalah seperti kualitas kopi dalam cangkir. Apa pun bentuknya dan kualitas cangkirnya. Kopi yang baik tetaplah terasa nikmat bagi siapa pun yang meminumnya.”

(Ahmad Zaky)

“Malam hari adalah waktu yang lebih tenang untuk bekerja. Ini membantu pemikiran.”

(Alexander Graham Bell)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur ke hadirat Allah SWT yang telah melimpahkan ridho, rahmat serta hidayah-Nya sehingga penulis bisa menyelesaikan tugas akhir yang berisi mengenai kegiatan magang yang dilakukan penulis di Badan Sistem Informasi Universitas Islam Indonesia. Tugas akhir ini dibuat demi memenuhi persyaratan dalam menjalani penjaluran magang yang telah ditempuh oleh penulis demi mendapatkan gelar sarjana. Dengan mengikuti penjaluran magang ini penulis bisa mendapatkan berbagai macam pengalaman yang berharga dan bermanfaat untuk masa depan nanti.

Dalam proses penyusunan laporan tengah ini telah banyak pihak-pihak yang membantu penulis, oleh karena itu tak lupa penulis mengucapkan terima kasih banyak kepada:

1. Allah SWT. karena dengan hidayahnya yang melancarkan segala aktivitas penulis selama melakukan program magang dan penulisan tugas akhir ini
2. Kedua Orang Tua dan Keluarga tercinta yang selalu memberikan dukungan do'a, semangat, dan kasih sayang.
3. Bapak Zainudin Zukhri, S.T., M.IT. selaku dosen pembimbing yang sudah memberikan banyak masukan serta arahan untuk pelaksanaan magang dan penulisan tugas akhir ini.
4. Senior-senior dan rekan-rekan Tim Akselerasi BSI UII yang telah membimbing dan mendampingi penulis selama mengikuti kegiatan magang.
5. Teman-teman penulis yang selalu memberikan dukungan semangat, do'a, dan motivasi.
6. Seluruh pihak yang tidak dapat saya sebutkan namanya satu persatu karena telah membantu dan mendukung penulis.

Semoga amal baik mereka mendapatkan balasan dari Allah SWT dengan balasan berlipat ganda dan semoga laporan tengah ini bisa menjadi manfaat bagi seluruh pihak yang telah membacanya.

Wassalamualaikum Wr. Wb.

Yogyakarta, 6 Juli 2023



(Rio Risqi Akbar Herlambang)

SARI

Badan Sistem Informasi (BSI) Universitas Islam Indonesia (UII) merupakan sebuah unit IT yang memberikan pelayanan di bidang teknologi dan informasi bagi UII. Saat ini BSI UII sedang mengembangkan sebuah aplikasi *mobile* bernama I'm UII yang ditujukan bagi mahasiswa UII untuk mempermudah proses perkuliahan mereka. Pada aplikasi ini terdapat beberapa fitur utama, yaitu fitur Schedule untuk manajemen jadwal kuliah mahasiswa, fitur What's On UII sebagai portal berita UII, fitur Reminder sebagai pengingat aktivitas perkuliahan bagi mahasiswa, dan fitur e-KTM atau Kartu Tanda Mahasiswa Elektronik sebagai upaya dalam mendigitalisasi KTM mahasiswa UII dan dapat diakses di mana saja melalui ponsel mereka.

Aktivitas magang dilakukan di dalam sebuah tim yang ada di BSI UII bernama Tim Akselerasi dan merupakan tim yang bertanggung jawab untuk pengembangan proyek aplikasi *mobile* I'm UII. Proyek ini dikembangkan menggunakan *framework* Flutter dan Lumen serta dengan menerapkan manajemen proyek berupa metode Waterfall. *Framework* Flutter digunakan karena kelebihanannya pada satu kode yang bisa digunakan untuk membuat aplikasi di banyak *device* yaitu Android dan iOS, sehingga *developer* tidak perlu mempelajari dua bahasa untuk dua *device* berbeda seperti Kotlin untuk Android dan Swift untuk iOS. *Framework* Lumen digunakan karena *framework* ini memang khusus untuk pembuatan API dan karena *framework* ini juga merupakan turunan dari *framework* Laravel menjadikannya lebih mudah untuk dipelajari dan digunakan. Pada proyek ini penulis bekerja untuk mengembangkan fitur Reminder, halaman profil, dan e-KTM pada aplikasi I'm UII serta pada bagian *backend* untuk membuat API dari seluruh fitur tersebut.

Pengujian pada aplikasi ini menggunakan BlackBox Testing dan hasil dari pengujian tersebut penulis berhasil mengembangkan fitur Reminder sehingga pengguna bisa melihat, menambah, mengedit, dan menghapus *reminder* serta beberapa fungsi lain di dalamnya, lalu penulis berhasil mengembangkan halaman profil yang berisi halaman personal data dan e-KTM di dalamnya. Selain itu pada bagian *backend* penulis berhasil membuat beberapa *endpoint* untuk fitur Reminder, halaman profil dan e-KTM. *Endpoint* tersebut seperti Get daftar *reminder*, Post *reminder*, Put *reminder*, dan Delete *reminder* untuk fitur Reminder, lalu *endpoint* Get *user* data yang digunakan untuk halaman profil dan e-KTM.

Kata kunci: aplikasi *mobile*, aplikasi I'm UII, *framework* Flutter, *framework* Lumen, API, metode Waterfall.

GLOSARIUM

| | |
|--|--|
| Android | Sistem operasi <i>mobile</i> yang dikembangkan oleh Google |
| Android Software Development Kit (SDK) | Kumpulan alat pengembangan perangkat lunak dengan berbagai komponen dan <i>library</i> yang disediakan oleh Google untuk mengembangkan aplikasi Android |
| <i>Backend Developer</i> | Seseorang yang bertanggung jawab untuk memproses logika bisnis, mengelola data, dan berinteraksi dengan <i>database</i> |
| <i>Background service Developer</i> | Layanan yang hanya berjalan jika aplikasi sedang berjalan Seseorang yang terlibat dalam proses pengembangan perangkat lunak atau aplikasi |
| <i>Environment</i> | Lingkungan kerja sistem |
| Firestore | Platform pengembangan aplikasi mobile dan web yang disediakan oleh Google serta menyediakan berbagai layanan dan alat yang dapat digunakan oleh pengembang untuk membangun, mengelola, dan menganalisis aplikasi |
| <i>Foreground service Framework</i> | Layanan yang bisa berjalan bahkan jika aplikasi sudah dihentikan Kerangka kerja atau rangkaian alat yang digunakan untuk membantu pengembang dalam membangun aplikasi atau sistem perangkat lunak |
| Google | Perusahaan teknologi multinasional yang berbasis di Amerika Serikat |
| Integrated Development Environment (IDE) | <i>Environment</i> pengembangan terpadu yang menggabungkan berbagai alat dan fitur yang diperlukan untuk pengembangan perangkat lunak dalam satu tempat. |
| iOS | Sistem operasi <i>mobile</i> yang dikembangkan oleh Apple Inc. dan digunakan pada perangkat Apple seperti iPhone, iPad, dan iPod Touch |
| JavaScript Object Notation (JSON) | Format pertukaran data ringan dan terbaca manusia yang digunakan untuk mengirim dan menerima data antara aplikasi atau sistem |
| <i>Mobile Developer</i> | Seseorang yang terlibat dalam pengembangan aplikasi mobile untuk platform seperti Android dan iOS. |

| | |
|--------------------------|--|
| Native | Pengembangan aplikasi yang dikodekan khusus untuk platform tertentu |
| <i>Software Engineer</i> | Profesional yang terlibat dalam pengembangan perangkat lunak |
| User Experience | Interaksi pengguna dengan produk, sistem, atau layanan |
| User Interface | Bagian aplikasi yang berfungsi sebagai titik interaksi antara pengguna dan sistem |
| User Persona | Representasi fiksi dari pengguna target atau pelanggan yang digunakan dalam proses desain |
| Version Control | Proses dan sistem yang digunakan untuk mengelola perubahan dalam kode sumber atau berkas-berkas proyek selama pengembangan perangkat lunak |

DAFTAR ISI

| | |
|---|------|
| HALAMAN JUDUL | i |
| HALAMAN PENGESAHAN DOSEN PEMBIMBING..... | ii |
| HALAMAN PENGESAHAN DOSEN PENGUJI..... | iii |
| HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR | iv |
| HALAMAN PERSEMBAHAN | v |
| HALAMAN MOTO | vi |
| KATA PENGANTAR | vii |
| SARI | viii |
| GLOSARIUM..... | ix |
| DAFTAR ISI..... | xi |
| DAFTAR TABEL..... | xiii |
| DAFTAR GAMBAR | xiv |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Ruang Lingkup..... | 3 |
| 1.3 Tujuan..... | 4 |
| 1.4 Manfaat..... | 4 |
| 1.5 Sistematika Penulisan..... | 5 |
| BAB II LANDASAN TEORI DAN KAJIAN PUSTAKA | 6 |
| 2.1 Pengembangan Aplikasi Berbasis Mobile..... | 6 |
| 2.2 <i>Framework</i> Flutter..... | 6 |
| 2.3 <i>Framework</i> Lumen | 7 |
| 2.4 Aplikasi I'm UII | 8 |
| 2.5 Metode Waterfall..... | 9 |
| 2.6 Tinjauan Pustaka | 10 |
| BAB III PELAKSANAAN MAGANG..... | 13 |
| 3.1 Manajemen Proyek..... | 13 |
| 3.2 Aktivitas Magang | 14 |
| 3.2.1 Induksi Magang BSI UII | 16 |
| 3.2.2 Survey Kebutuhan Calon Pengguna Aplikasi I'm UII..... | 17 |
| 3.2.3 Perancangan Basis Data | 19 |
| 3.2.4 Pengembangan <i>Backend</i> | 22 |

| | | |
|--|---|----|
| 3.2.5 | Pengembangan Fitur Reminder, Profil, dan e-KTM | 26 |
| 3.3 | Hasil Pelaksanaan Magang..... | 41 |
| 3.3.1 | Hasil Survey Pengguna | 41 |
| 3.3.2 | Hasil Pengembangan Fitur Reminder, Profil, dan e-KTM..... | 42 |
| 3.3.3 | Hasil Pengembangan <i>Backend</i> | 52 |
| BAB IV REFLEKSI PELAKSANAAN MAGANG | | 58 |
| 4.1 | Relevansi Akademik..... | 58 |
| 4.2 | Pembelajaran Magang | 59 |
| 4.3 | Kendala, Hambatan, dan Tantangan..... | 59 |
| BAB V PENUTUP | | 62 |
| 5.1 | Kesimpulan..... | 62 |
| 5.2 | Saran..... | 62 |
| DAFTAR PUSTAKA | | 64 |
| LAMPIRAN..... | | 66 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Ringkasan penelitian sebelumnya..... | 11 |
| Tabel 3.1 <i>Tools</i> dan <i>Framework</i> yang digunakan..... | 15 |
| Tabel 3.2 Ringkasan aktivitas magang di BSI UII | 15 |
| Tabel 3.3 Runtutan acara induksi BSI UII..... | 16 |
| Tabel 3.4 Daftar pertanyaan wawancara..... | 18 |
| Tabel 3.5 Tabel hasil pengujian | 43 |
| Tabel 3.6 Tabel hasil pengujian <i>endpoint</i> | 52 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1.1 Bagan Struktur Organisasi BSI UII | 2 |
| Gambar 1.2 Gedung Rektorat UII..... | 3 |
| Gambar 2.1 Tahapan Waterfall Model | 10 |
| Gambar 3.1 Suasana <i>meeting</i> Tim Akselerasi | 14 |
| Gambar 3.2 API <i>contract</i> untuk GET daftar <i>reminder</i> | 21 |
| Gambar 3.3 Rancangan tabel basis data untuk fitur <i>reminder</i> di Phpmyadmin | 22 |
| Gambar 3.4 Basis data untuk aplikasi I'm UII melalui DBeaver | 23 |
| Gambar 3.5 Kode untuk mengulang <i>reminder</i> berdasarkan nilai pada kolom <i>repeated</i> | 25 |
| Gambar 3.6 Kode untuk <i>query</i> DELETE <i>reminder</i> | 26 |
| Gambar 3.7 Testing <i>endpoint</i> untuk DELETE <i>reminder</i> | 26 |
| Gambar 3.8 Tampilan desain halaman profil dan e-KTM..... | 29 |
| Gambar 3.9 Bagian dari tampilan desain halaman <i>reminder</i> | 29 |
| Gambar 3.10 Halaman daftar <i>reminder</i> dan menu pilihan ketika aksi <i>swipe</i> | 31 |
| Gambar 3.11 Halaman tambah <i>reminder</i> baru dan halaman untuk tambah orang | 32 |
| Gambar 3.12 Halaman detail dan ketika ingin mengubah <i>reminder</i> | 32 |
| Gambar 3.13 Dialog konfirmasi ketika melakukan aksi hapus | 33 |
| Gambar 3.14 Menu pilihan <i>sidebar</i> | 34 |
| Gambar 3.15 Tampilan halaman profil | 35 |
| Gambar 3.16 Halaman data profil pengguna | 36 |
| Gambar 3.17 Halaman e-KTM dan tampilan ketika menekan tombol memperbesar | 37 |
| Gambar 3.18 Implementasi <i>request</i> GET menggunakan Dio pada Flutter..... | 38 |
| Gambar 3.19 Penerapan <i>request</i> POST pada Dio di Flutter | 39 |
| Gambar 3.20 <i>Request</i> edit <i>reminder</i> pada fungsi..... | 40 |
| Gambar 3.21 <i>Request</i> hapus pada fungsi di Dio | 41 |
| Gambar 3.22 User persona..... | 42 |
| Gambar 3.23 Halaman lihat <i>reminder</i> dan detail <i>reminder</i> | 49 |
| Gambar 3.24 Halaman buat dan ubah <i>reminder</i> | 49 |
| Gambar 3.25 Menu dialog hapus <i>reminder</i> | 50 |
| Gambar 3.26 <i>Sidebar</i> | 50 |
| Gambar 3.27 Halaman profil dan personal data pengguna..... | 51 |
| Gambar 3.28 Halaman e-KTM pengguna..... | 51 |
| Gambar 3.29 GET <i>reminder list</i> | 54 |

| | |
|---|----|
| Gambar 3.30 GET <i>reminder</i> detail | 55 |
| Gambar 3.31 POST <i>reminder</i> | 55 |
| Gambar 3.32 PUT edit <i>user reminder</i> | 56 |
| Gambar 3.33 PUT edit <i>reminder completed status</i> | 56 |
| Gambar 3.34 PUT edit <i>reminder pinned status</i> | 56 |
| Gambar 3.35 DELETE <i>user reminder</i> | 57 |
| Gambar 3.36 GET <i>user data</i> | 57 |

BAB I

PENDAHULUAN

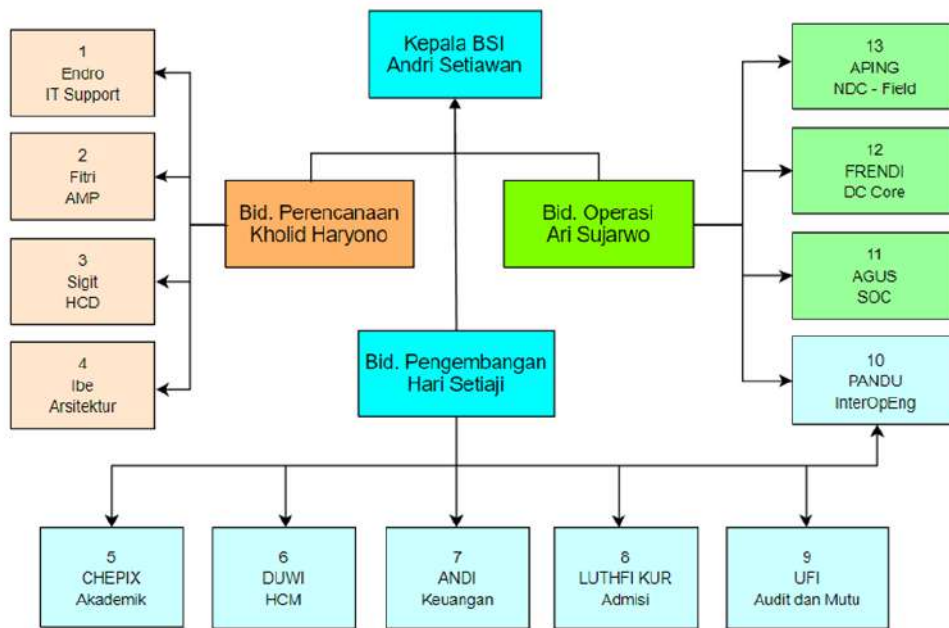
1.1 Latar Belakang

Badan Sistem Informasi (BSI) merupakan sebuah unit *Information Technology* (IT) yang berada di bawah naungan Universitas Islam Indonesia. BSI secara umum memiliki peran di UII untuk bertugas memberikan pelayanan bagi segala bidang yang terdapat di UII. BSI juga didukung dengan berbagai sumber daya profesional di bidang teknologi informasi yang terdiri dari tim dengan memiliki tugas khusus untuk melakukan analisis, *programmer*, *engineer*, dan juga konsultan. Seluruh tim tersebut berperan untuk mengurus seluruh sistem layanan teknologi informasi di UII. Untuk bidang-bidang yang dilayani oleh BSI sendiri untuk UII, yaitu bidang perencanaan sistem informasi, bidang pengembangan sistem informasi, divisi operasi sistem informasi, divisi layanan pengguna, divisi administrasi dan manajemen proyek.

Jumlah pengguna yang menggunakan layanan BSI UII sudah mencapai kurang lebih sekitar 30.000 (tiga puluh ribu) pengguna yang terdiri dari berbagai macam golongan mulai dari mahasiswa, dosen, tenaga kependidikan, dan para pemaku kepentingan lainnya. Dengan menggunakan rumusan tiga peran BSI, yaitu melayani, mendampingi dan mengakselerasi, BSI sudah siap memberikan pelayanan terbaik bagi penggunanya yang terbilang memiliki jumlah cukup banyak.

BSI UII dikepalai oleh Bapak Mukhammad Andri Setiawan dan memiliki tiga bidang utama di dalamnya. Pertama adalah bidang perencanaan yang dikepalai oleh Bapak Kholid Haryono, kedua adalah bidang pengembangan yang dikepalai oleh Bapak Hari Setiaji, dan terakhir yang ketiga adalah bidang operasional yang dikepalai oleh Bapak Ari Sujarwo. Untuk struktur organisasi BSI UII bisa dilihat lebih lengkap pada Gambar 1.1 Bagan Struktur Organisasi BSI UII yang berupa bagan hierarki.

BSI mengadakan program magang yang bertujuan untuk membantu mahasiswa mendapatkan sebuah kesempatan untuk mengikuti kerja di industri dengan proyek yang nyata, lalu mendapatkan talenta profesional pada bidang IT yang siap untuk terjun berkarya dan berinovasi sehingga diharapkan bisa memberikan tenaga kerja yang mampu untuk memperkuat tim di BSI dalam mewujudkan visi transformasi digital di UII. Selain itu di BSI juga terdapat tim-tim yang bertugas menangani berbagai macam proyek baik itu untuk BSI ataupun untuk UII.



Gambar 1.1 Bagan Struktur Organisasi BSI UII

Pertama ada Tim Admisi yang bertanggung jawab untuk menangani segala urusan admisi seperti pendaftaran atau penerimaan mahasiswa baru UII, Tim Keuangan yang menangani bidang finansial, lalu Tim Interoperability yang bertugas dalam informasi dan keamanan, Tim HCM atau Human Capital Management untuk menangani sumber daya manusia, Tim Network dan Data Center yang bertanggung jawab dalam jaringan dan data, Tim SOC atau Security Operation Center yang bertanggung jawab dalam masalah keamanan, dan Tim Akademik yang bertugas untuk menangani bagian akademik UII.

Pemangag bekerja langsung di kantor BSI yang letaknya berada di Gedung GBPH Prabuningrat – Rektorat Universitas Islam Indonesia (Lantai 4) seperti yang terlihat pada Gambar 1.2 Gedung Rektorat UII. Jadwal kerja dilakukan setiap hari Senin hingga Jumat yang dimulai pada pukul 08.00 WIB hingga waktu selesai kerja pada jam 12.00 WIB. Karena pemangag hanya mendapat jatah waktu setengah hari dibandingkan dengan pekerja tetap di BSI UII maka jam istirahat yang biasanya dilakukan pada pukul 12.00 WIB, untuk para pemangag jam tersebut menjadi jam selesai kerja. Selain jam-jam yang telah ditentukan tersebut, terkadang terdapat juga kegiatan-kegiatan lain seperti rapat kerja yang dilakukan di luar jam kerja karena terbatasnya waktu untuk pekerja lain sehingga perlu menentukan waktu tertentu di luar waktu kerja tetap.

Dari berbagai proyek yang dikerjakan oleh BSI UII, salah satu dari proyek-proyek tersebut adalah proyek aplikasi *mobile* I'm UII. Proyek ini sedang dikembangkan oleh salah satu tim yang berada di BSI UII, yaitu Tim Akselerasi dan pemegang mendapatkan tugas dengan peran magang sebagai *software engineer* untuk dimasukkan ke dalam tim tersebut dan membantu proses pengembangan proyek aplikasi dengan durasi waktu selama delapan bulan. Aplikasi *mobile* I'm UII sendiri merupakan sebuah aplikasi dengan target penggunanya adalah mahasiswa UII dan memiliki tujuan yang diharapkan dapat membantu mahasiswa UII melaksanakan kegiatan perkuliahannya dengan berbagai macam sarana fitur utama di dalam aplikasi I'm UII. Pada proses pengembangannya, fase pertama pada proyek ini adalah aplikasi dikembangkan untuk *device* Android terlebih dahulu dan setelah itu pada fase berikutnya akan diikuti oleh *device* lain, yaitu iOS. Tugas utama yang diberikan adalah untuk menangani fitur Reminder, halaman profil dan e-KTM serta mengerjakan bagian *backend* dari fitur tersebut. Lalu terdapat juga beberapa pekerjaan tambahan lainnya, yaitu untuk mengimplementasikan Firebase Analytics dan Firebase Dynamic Links ke dalam aplikasi I'm UII.



Gambar 1.2 Gedung Rektorat UII

1.2 Ruang Lingkup

Durasi program magang di BSI UII dilakukan selama 8 bulan yang dimulai dari bulan September 2022 hingga bulan April 2023. Pemegang ditempatkan di sebuah tim khusus untuk pengembangan proyek aplikasi *mobile*, tim ini bernama Tim Akselerasi dan berjumlah delapan orang yang sudah termasuk dengan anggota magang. Bagian-bagian dari tim ini terdiri dari Supervisor untuk tim berasal dari kepala bidang pengembangan, yaitu Bapak Hari Setiaji.

Kemudian untuk pembagian tugasnya, terdapat dua orang yang mengerjakan UI/UX. Satu orang ditempatkan untuk pengerjaan *backend* dan satu orang ditambah tiga anggota magang untuk pengerjaan *mobile*. Untuk anggota magang juga mendapatkan tugas dalam menangani bagian *backend* untuk masing-masing fitur yang dikembangkan.

Tugas Akhir ini akan berfokus pada proyek aplikasi I'm UII. Proyek ini merupakan sebuah aplikasi *mobile* bernama I'm UII yang sedang dikembangkan oleh BSI UII dengan tujuan untuk memberikan fasilitas pembantu bagi mahasiswa yang dapat diakses dengan mudah melalui *smartphone* masing-masing mahasiswa dengan harapan dapat membantu mahasiswa dalam menjalani proses perkuliahan di UII. Proyek aplikasi *mobile* ini dikembangkan dengan menggunakan Visual Studio Code sebagai IDE utamanya, lalu *framework* Flutter untuk bagian pengembangan *mobile*-nya dan *framework* Lumen sebagai bagian *backend*-nya untuk mengurus bagian pengolahan data di dalam basis data dengan menggunakan API sebagai perantara penghubung antara aplikasi *mobile* dengan basis data. Pembahasan akan berfokus pada proses pengembangan aplikasi I'm UII menggunakan Flutter dan bagaimana proses pengolahan datanya dengan menggunakan Lumen.

1.3 Tujuan

Berdasarkan latar belakang di atas, adapun tujuan dari penulisan laporan akhir magang ini, yaitu sebagai berikut:

- a. Menjelaskan cara proses pengembangan aplikasi I'm UII pada fitur Reminder, halaman profil dan e-KTM menggunakan *framework* Flutter.
- b. Menjelaskan cara pembuatan API untuk fitur Reminder, halaman profil dan e-KTM dengan menggunakan *framework* Lumen dan cara mengintegrasikannya pada Flutter.

1.4 Manfaat

Berdasarkan tujuan di atas, laporan akhir ini diharapkan bisa memberikan manfaat sebagai:

- a. Dokumen cara pengembangan aplikasi *mobile* dengan menggunakan Visual Studio Code dan *framework* Flutter.
- b. Dokumen cara pengembangan bagian *backend* untuk aplikasi *mobile* dengan menggunakan *framework* Lumen.

1.5 Sistematika Penulisan

Untuk sistematika penulisan pada laporan akhir ini adalah sebagai berikut:

- a. BAB I membahas tentang latar belakang, ruang lingkup, tujuan, serta manfaat dalam pengembangan aplikasi *mobile* menggunakan *framework* Flutter dan Lumen
- b. BAB II membahas tentang dasar-dasar teori yang berkaitan dengan aplikasi *mobile*, *framework* Flutter dan Lumen.
- c. BAB III menjelaskan tentang proses pelaksanaan magang, mulai dari manajemen proyek yang diterapkan pada proyek aplikasi *mobile* I'm UII di Tim Akselerasi, aktivitas – aktivitas magang yang dilakukan pada pengerjaan proyek, dan hasil yang didapatkan setelah melalui aktivitas magang tersebut.
- d. BAB IV menjelaskan mengenai refleksi pembelajaran magang seperti manfaat yang didapatkan, kendala, hambatan dan tantangan yang dihadapi ketika menjalani aktivitas magang di BSI UII serta relevansi teori – teori yang berlaku dengan fakta yang terjadi di lapangan.
- e. BAB V menjelaskan kesimpulan yang dapat diambil setelah menjalani berbagai rangkaian proses kegiatan magang di BSI UII dan saran yang dapat diberikan untuk kedepannya.

BAB II LANDASAN TEORI DAN KAJIAN PUSTAKA

2.1 Pengembangan Aplikasi Berbasis Mobile

Aplikasi *mobile* atau sering disebut juga dengan aplikasi seluler, adalah perangkat lunak yang dirancang khusus untuk dijalankan pada perangkat *mobile* seperti *smartphone* dan tablet. Aplikasi *mobile* sendiri berasal dari dua kata, yaitu aplikasi dan *mobile*. Secara istilah, aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi untuk pengguna atau aplikasi yang lain sedangkan *mobile* adalah perpindahan dari suatu tempat ke tempat yang lain (Muktasim & Buyens, 2001). Pada masa ini aplikasi *mobile* telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari karena memungkinkan penggunaannya untuk melakukan berbagai aktivitas dan mendapatkan akses ke berbagai layanan dengan mudah dan cepat. Aplikasi *mobile* menawarkan berbagai fitur dan fungsi, mulai dari media sosial, hiburan, komunikasi, hingga keuangan dan produktivitas.

Salah satu keuntungan utama aplikasi *mobile* adalah kemampuannya untuk memberikan aksesibilitas yang tinggi kepada pengguna. Menurut (Holla & Katti, 2012), pada perangkat *mobile* seperti Android, SDK-nya menyediakan kumpulan antarmuka pemrograman aplikasi (API) yang luas, modern, dan tangguh, selain itu layanan inti dari *device* Android dapat diakses langsung oleh semua aplikasi. Ketika diberikan izin yang sesuai, aplikasi Android dapat saling berbagi data dan mengakses sumber daya bersama dengan aman. Holla & Katti pun menuturkan Android memberikan kesempatan bagi para pengembang untuk membuat aplikasi yang sangat kaya dan inovatif dengan platformnya yang bersifat *open development*. Para developer bebas memanfaatkan perangkat keras, mengakses informasi lokasi, menjalankan layanan latar belakang, mengatur alarm, menambahkan pemberitahuan, dan masih banyak lagi.

2.2 Framework Flutter

Flutter adalah sebuah *framework open-source* pengembangan aplikasi *multi-platform* yang dikembangkan oleh Google, hanya dengan satu *codebase* atau satu kode saja sudah bisa digunakan untuk membuat banyak aplikasi di berbagai *devices* (Thomas, 2019). Definisi *framework* sendiri merupakan standar *coding*, *best practice* dan kumpulan fungsi umum untuk memudahkan pembuatan suatu aplikasi (Wijayanto et al., 2018). Dengan menggunakan Flutter, pengembang dapat membuat aplikasi yang konsisten dan responsif di berbagai *device* seperti iOS, Android, desktop dan bahkan web. Salah satu keunggulan utama dari Flutter adalah

kemampuannya untuk membuat antarmuka pengguna yang menarik, indah, dan responsif dengan cepat (Adservio, 2022). Framework ini menggunakan bahasa pemrograman Dart yang dioptimalkan untuk membuat aplikasi *mobile* modern.

Salah satu fitur penting dari Flutter adalah *hot reload* yang memungkinkan pengembang untuk melihat perubahan yang dilakukan pada kode secara langsung dalam waktu nyata tanpa perlu menjalankan ulang *build* aplikasi yang membutuhkan waktu lama (Dagne, 2019). Hal ini mempercepat proses pengembangan dan memungkinkan pengembang untuk menguji dan memperbaiki kesalahan dengan lebih efisien. Selain itu, Flutter juga menyediakan kustom *widget* yang kuat dan fleksibel, sehingga memungkinkan pengembang untuk membuat antarmuka pengguna sesuai keinginan mereka sendiri (Dagne, 2019). Framework ini juga dilengkapi dengan banyak *plugin* dan paket yang mempermudah integrasi dengan layanan pihak ketiga seperti API, penyimpanan data, dan autentikasi pengguna. Dengan kemampuan dan fitur yang lengkap, Flutter telah menjadi pilihan populer bagi pengembang untuk mengembangkan aplikasi *mobile* yang cepat, indah, dan dapat diandalkan.

2.3 Framework Lumen

Framework Lumen merupakan sebuah *framework* turunan dari Laravel yang juga merupakan *web application framework* namun untuk Lumen lebih dikhususkan pada pembuatan Web API atau Web Service. Web API sendiri terdiri dari dua kata, yaitu *web* dan API, API merupakan singkatan dari Application Programming Interface, yaitu kumpulan dari berbagai macam fungsi perintah dan protokol yang bertugas sebagai jembatan antara satu aplikasi dengan aplikasi yang lainnya seperti menerima permintaan/*request* dan memberikan respon (Wijayanto et al., 2018).

Application Programming Interface (API) memiliki cara kerjanya sendiri dalam menangani proses *request* dan memberikan respon yang tepat, yaitu Proses dimulai dengan aplikasi klien mengirim permintaan (*request*) ke API. Permintaan ini berisi informasi yang diperlukan untuk melakukan operasi tertentu, seperti metode HTTP (GET, POST, PUT, DELETE), *endpoint* atau URL tujuan, parameter, *header*, dan data *payload* (Dicoding Indonesia, n.d.-a).

API menerima permintaan dan memprosesnya sesuai dengan aturan dan logika yang ditentukan oleh API tersebut. Pada tahap ini, API akan memvalidasi permintaan, memeriksa otorisasi dan keamanan, serta memeriksa ketersediaan sumber daya yang diminta. Jika ada masalah, seperti permintaan yang tidak valid atau otorisasi yang gagal, API akan mengirimkan

respons yang sesuai dengan status kode yang relevan (misalnya, 400 untuk permintaan yang tidak valid atau 401 untuk otorisasi yang gagal) (Dicoding Indonesia, n.d.-a). Setelah permintaan diverifikasi dan valid, API akan mengakses sumber daya yang diperlukan. Sumber daya ini dapat berupa data dalam basis data, *file*, layanan eksternal, atau komponen lainnya. API akan melakukan operasi yang diperlukan untuk mendapatkan, mengubah, atau memanipulasi sumber daya ini sesuai dengan permintaan yang diterima.

Setelah API berhasil mengakses dan memanipulasi sumber daya, langkah berikutnya adalah menghasilkan respons (*response*) yang akan dikirimkan kembali ke aplikasi klien. Respons ini berisi informasi yang diminta, seperti status kode HTTP (misalnya, 200 untuk permintaan yang berhasil atau 404 untuk sumber daya yang tidak ditemukan), *header* yang relevan, dan data yang dikirimkan dalam respons (Dicoding Indonesia, n.d.-a). API mengirimkan respons kembali ke aplikasi klien melalui jaringan menggunakan protokol yang telah ditentukan, seperti HTTP atau HTTPS. Respons ini dikemas dalam format yang sesuai, seperti JSON, XML, atau format lainnya (Dicoding Indonesia, n.d.-a).

2.4 Aplikasi I'm UII

I'm UII merupakan sebuah aplikasi *mobile* yang sedang dikembangkan oleh BSI UII dan ditujukan untuk mahasiswa Universitas Islam Indonesia agar mempermudah mereka dalam menjalankan perkuliahan dan aktivitas kampus lainnya. Pada aplikasi *mobile* ini terdapat beberapa fitur utama yang nantinya bisa digunakan oleh mahasiswa, yaitu fitur Reminder, fitur Schedule, fitur What's On UII atau portal berita UII dan fitur e-KTM (Kartu Tanda Mahasiswa Elektronik).

Fitur Schedule merupakan sebuah fitur kalender untuk manajemen jadwal kuliah mahasiswa UII, sehingga mahasiswa bisa melihat jadwal lengkap seperti jam kuliah, ruangan kuliah, dan lainnya, fitur What's On adalah fitur untuk menampilkan berita dan informasi penting yang ada di UII, fitur Reminder merupakan sebuah fitur yang berfungsi sebagai pengingat hal-hal penting mengenai kegiatan mahasiswa, nantinya fitur ini akan diintegrasikan dengan tagihan UII sehingga dapat memunculkan notifikasi pengingat bagi mahasiswa tentang tagihan perkuliahan melalui *smartphone*-nya masing-masing, terakhir fitur e-KTM, yaitu merupakan sebuah fitur untuk memvirtualisasi kartu tanda mahasiswa UII ke dalam aplikasi *mobile* sehingga pengguna tidak perlu takut kelupaan membawa KTM fisik karena sudah ada KTM yang bisa diakses secara digital. Adapun kontribusi yang diberikan pada proyek ini adalah untuk mengembangkan fitur Reminder, halaman profil, dan e-KTM.

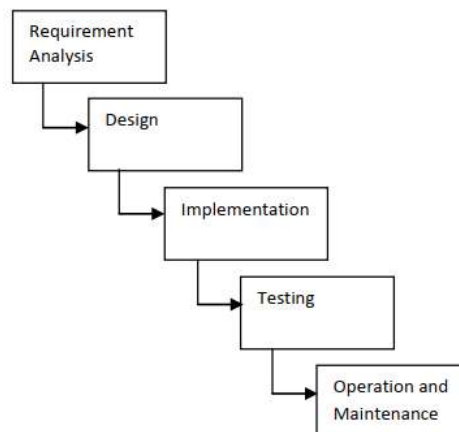
Pada fitur reminder ini terdapat beberapa aksi yang bisa pengguna lakukan, pertama pengguna bisa membuat *reminder* yang diinginkan, di dalam pembuatannya pengguna bisa melakukan kustomisasi seperti mengubah warna pengingat agar bisa membedakan dengan pengingat yang lainnya, pengguna juga bisa mengatur kapan pengingat tersebut akan diulang, apakah setiap hari, setiap minggu, dan lainnya. Selain itu, pengguna juga bisa menentukan prioritas pengingat, jika pengingat yang dibuat memiliki urgensi yang tinggi, pengguna dapat menekan opsi pin untuk membuat pengingat tersebut menjadi tampil pada daftar teratas dan memiliki sistem notifikasi dengan prioritas tinggi. Pengingat yang memiliki prioritas tinggi dalam sistem notifikasinya akan ditampilkan paling atas pada bagian *notification bar*, untuk yang prioritas rendah akan tetap muncul notifikasi namun jika muncul notifikasi untuk pengingat dengan prioritas tinggi akan menggeser pengingat dengan prioritas rendah ke bawah pada *notification bar*. Pengguna juga bisa mengundang pengguna lain untuk ikut dalam satu pengingat yang sama, jika ada kegiatan yang diikuti bersama, pengguna bisa mengundang pengguna lain tersebut. Fitur Reminder dibutuhkan pada aplikasi ini karena sebagai upaya untuk mengimplementasikan notifikasi untuk tagihan mahasiswa UII pada aplikasi *smartphone* dan dalam hal ini melalui aplikasi I'm UII. Untuk halaman profil pengguna dapat melihat informasi pribadi seperti *email* UII, NIM, fakultas, jurusan, dan lain-lain, lalu pada e-KTM pengguna bisa mengakses fitur ini untuk melihat KTM mereka secara digital, e-KTM dibuat dengan mereplika KTM asli UII menggunakan *widget-widget* yang ada pada *framework* Flutter.

2.5 Metode Waterfall

Proses pengembangan aplikasi I'm UII menggunakan Waterfall Model. Model ini merupakan sebuah model statis dengan menggunakan pendekatan secara linear dan berurutan, sehingga satu pekerjaan perlu untuk diselesaikan terlebih dahulu sebelum lanjut ke tahap pekerjaan lainnya (Adenowo & Adenowo, 2013).

Waterfall model termasuk sebagai metode yang sudah lama muncul dalam manajemen proyek dan sudah banyak buku maupun penelitian yang menjelaskan mengenai tahapan di dalamnya. Terdapat sebuah buku yang menjelaskan tahapan pada metode Waterfall dipecah menjadi *requirement analysis*, desain, *coding*, dan testing (Flower, 2003). Buku lain terkait pembahasan metode ini juga menjelaskan mengenai tahap metode Waterfall yang dibagi menjadi lima bagian, yaitu pertama komunikasi (inisiasi proyek dan mengumpulkan *resource*), kedua *planning* (estimasi, penjadwalan, dan *tracking*), ketiga modeling (analisis dan desain),

keempat konstruksi (*coding* dan *testing*), dan terakhir yang kelima adalah *deployment* (*delivery*, *support*, dan *feedback*) (Pressman, 2005). Dari penjelasan tahapan – tahapan tersebut (Adenowo & Adenowo, 2013) menyimpulkan bahwa metode Waterfall terdiri dari lima tahap utama, yaitu *requirement analysis*, desain, implementasi atau *coding*, *testing*, dan *operation/maintenance*. Untuk runtutan tahapan-tahapan dari Waterfall Model dapat dilihat pada Gambar 2.1 Tahapan Waterfall Model.



Gambar 2.1 Tahapan Waterfall Model

Sumber: (Adenowo & Adenowo, 2013)

2.6 Tinjauan Pustaka

Terdapat beberapa penelitian sebelumnya yang telah dilakukan terkait penggunaan *framework* Flutter dalam pengembangan sebuah aplikasi *mobile*. Terdapat sebuah penelitian mengenai usaha dalam menyempurnakan sistem informasi di Politeknik Sahid dengan menggunakan aplikasi berbasis Android (Hakim et al., 2019). Pada penelitian tersebut dijelaskan mengenai metode yang digunakan dalam proses pengembangan dan penggunaan *testing*, yaitu Black Box Testing, hasil dari penelitian tersebut pun seluruh fungsional sistem sudah mencapai target yang diharapkan. Hanya saja pada penelitian ini tidak terlalu dijelaskan mengenai penerapan Flutter untuk pengembangan aplikasinya.

Selanjutnya terdapat penelitian lain yang membahas mengenai proses konversi yang dilakukan dari sistem informasi akademik sebuah universitas yang berbasis *web* ke aplikasi berbasis *mobile* menggunakan *framework* Flutter dengan fitur dan fungsionalitas yang sama seperti yang ada pada *web* (Pratama & Kamisutara, 2021). Penelitian ini memiliki konsep penelitian yang sama pada penjelasan di paragraf sebelumnya namun metode penelitian yang digunakan berbeda karena menggunakan metode yang sama pada tugas akhir ini, yaitu metode Waterfall. Pada pengembangan aplikasi *mobile* di penelitian ini juga menggunakan API yang

sama pada sistem informasi akademik berbasis *web* mereka, namun tidak dijelaskan mengenai cara penerapannya menggunakan Flutter.

Penelitian lain yang lebih berfokus pada pengembangan untuk *backend*-nya terdapat pada sebuah penelitian yang membahas mengenai pengembangan sebuah *web service* untuk sistem informasi akademik yang ada di Universitas Pasundan (Widarno & Putra, 2023). Pengembangan *Web service* ini dilakukan dengan tujuan supaya layanan yang digunakan bisa dipakai oleh aplikasi lain seperti aplikasi *mobile*, *desktop*, dan lainnya, sebab sistem informasi akademik yang digunakan tidak menggunakan *web service* dan hanya bisa diakses melalui versi *website*-nya saja. Pada penelitian ini lebih berfokus pada pembuatan sebuah *endpoint* di API menggunakan *framework* Lumen.

Pada tabel berikut ini tersaji ringkasan mengenai bahasa atau *framework* apa saja yang digunakan beserta metode dan hasil penelitiannya dari penjelasan pada paragraf – paragraf sebelumnya.

Tabel 2.1 Ringkasan penelitian sebelumnya

| No. | Judul Penelitian | Bahasa/ <i>Framework</i> Pemrograman | Metode Penelitian | Hasil Penelitian |
|-----|---|---|---|---|
| 1. | Pengembangan Sistem Informasi Akademik Berbasis Android Menggunakan Flutter di Politeknik | Flutter | Analisis sistem berjalan, evaluasi sistem berjalan, perancangan sistem, implementasi, pengujian | Aplikasi <i>mobile</i> , hasil uji Black Box |
| 2. | Pengembangan Sistem Informasi Akademik Berbasis Mobile Menggunakan Flutter di Universitas Narotama Surabaya | Flutter | Metode Waterfall | Aplikasi <i>mobile</i> Android |
| 3. | Penyediaan Layanan Web Service Menggunakan Framework Lumen Untuk Berbagi Layanan Data (Studi Kasus: SITU Akademik Universitas Pasundan) | PHP, Lumen | Identifikasi masalah, pengumpulan data, pembangunan perangkat lunak | <i>Web service</i> , dokumentasi <i>endpoint</i> API, hasil uji Black Box |

| | | | | |
|--|--|--|---|--|
| | | | (<i>web service implementation</i>), kesimpulan dan saran | |
|--|--|--|---|--|

Berdasarkan pada Tabel 2.1, aplikasi *mobile* I'm UII dikembangkan dengan mengimplementasikan beberapa gabungan dari penelitian – penelitian tersebut, karena penelitian yang disebutkan masih belum memiliki penjelasan secara lengkap pada beberapa bagian. Sebagai contoh belum adanya penjelasan tentang cara integrasi *web service* atau API yang telah dibuat ke dalam aplikasi yang mereka buat dengan menggunakan *framework* Flutter dan bagaimana cara kerjanya (Pratama & Kamisutara, 2021). Cara integrasi API ke sebuah aplikasi *mobile* merupakan hal yang penting untuk dibahas karena proses tersebut tentu berbeda jika menggunakan bahasa pemrograman lainnya seperti pada Kotlin ataupun integrasi API pada aplikasi lainnya seperti aplikasi yang berbasis *web*. Maka dari itu, pada tugas akhir ini akan dijelaskan bagaimana proses pengembangan aplikasi I'm UII dengan menggunakan Flutter pada fitur Reminder, halaman profil dan e-KTM serta bagaimana cara membuat sebuah *endpoint* API untuk fitur – fitur tersebut dan implementasinya di dalam aplikasi I'm UII.

BAB III

PELAKSANAAN MAGANG

3.1 Manajemen Proyek

Tim Akselerasi menerapkan metode manajemen proyek untuk aplikasi *mobile* I'm UII dengan metode Waterfall, karena Tim Akselerasi tidak mengikuti tim lain di BSI yang sudah menggunakan SCRUM dalam pengembangan proyeknya. Tim Akselerasi bekerja hanya pada waktu selesai jam kerja wajib di BSI, yaitu pada jam 08.00-16.00 WIB, sehingga proses pengerjaan proyek tidak akan terlalu efektif jika menggunakan SCRUM yang memerlukan anggotanya untuk aktif bersama seperti jika saat melakukan *sprint planning* dan *sprint review*. Dengan metode Waterfall proses pengembangan aplikasi I'm UII dilakukan secara bertahap.

Proses inisiasi dilakukan oleh anggota Tim Akselerasi yang dipimpin oleh Bapak Hari Setiaji selaku supervisor dan yang menjadi inisiator proyek aplikasi I'm UII ini bersama dengan Mas Elbo Shindi Pangestu. Pada tahap ini juga pemagang turut ikut andil dalam proses mengumpulkan data dari mahasiswa mengenai perkuliahan mereka di UII untuk nanti dijadikan acuan dalam memberikan gambaran umum mengenai tujuan dari proyek aplikasi I'm UII oleh Tim Akselerasi dan dapat lanjut ke tahapan berikutnya. Pada tahap ini pemagang akan melakukan wawancara kepada mahasiswa UII dan dilakukan selama tiga minggu serta akan memiliki *output* berupa User Persona. User Persona ini akan diberikan kepada tim produk, yaitu Mas Anwaruddin Ridho Novianto dan Mas Muhammad Habib Izdhihar Syaf sebagai bahan dalam perancangan UI/UX I'm UII.

Selanjutnya pada tahap implementasi, pemagang diberikan tugas untuk mengerjakan fitur-fitur utama pada aplikasi I'm UII yang desainnya sudah dirancang oleh tim produk. Pada Tim Akselerasi terdapat tiga orang anggota magang, dan setiap pemagang mendapatkan tugas pengerjaan fitur yang berbeda-beda, pertama Saudara Alif Maulana Rizqi yang mengerjakan fitur Schedule, lalu Saudara Fahrudin Nasikh Az Zuhdu yang mengerjakan fitur What's On UII atau portal berita UII, lalu Rio Risqi Akbar Herlambang yang mengerjakan fitur Reminder dan e-KTM. Selain dari pengerjaan fitur-fitur utama tersebut, pemagang juga mengerjakan bagian *backend* dari masing-masing fitur yang dikerjakan dan dibantu oleh Mas Elbo yang memandu pemagang dalam pengerjaan *backend*. Kedepannya pemagang juga akan mendapatkan tugas tambahan untuk melengkapi aplikasi I'm UII ini, yaitu menambahkan *service* Firebase ke dalam I'm UII dan akan memanfaatkan Firebase Messaging untuk sistem notifikasi aplikasi, Firebase Dynamic Links untuk fitur *share* seperti *share* berita yang ada pada What's On UII,

dan Firebase Analytics untuk melihat data analitik aplikasi, adapun pada profil pengguna akan dilakukan integrasi dengan basis data UII sehingga user melakukan login dengan menggunakan akun UII dan menampilkan data profil serta e-KTM berdasarkan data pengguna.

Meeting yang dilakukan Tim Akselerasi diadakan setiap 1 – 2 kali setiap bulannya untuk membahas proses perkembangan proyek aplikasi I'm UII. Dalam *meeting* tersebut pemegang memaparkan hasil pekerjaannya pada proyek ini, pemegang juga perlu menjelaskan masalah-masalah apa saja yang dialami selama proses pengembangan aplikasi, dan di *meeting* ini akan dicari solusinya dan diharapkan bisa melanjutkan pekerjaannya. Suasana *meeting* Tim Akselerasi bisa dilihat pada Gambar 3.1 yang membahas proses implementasi fitur aplikasi ke Flutter.



Gambar 3.1 Suasana *meeting* Tim Akselerasi

3.2 Aktivitas Magang

Aktivitas magang yang dilakukan di BSI UII sudah terhitung terlaksana selama empat bulan terhitung sejak 5 September 2022 hingga Desember 2022. Di BSI UII pemegang diberikan posisi sebagai *software engineer* dan bertanggung jawab di Tim Akselerasi untuk mengurus bagian *mobile* dan *backend*. Pemegang diberikan sebuah tugas untuk menyelesaikan proyek aplikasi *mobile* Tim Akselerasi, proyek tersebut bernama I'm UII, yaitu sebuah aplikasi mobile yang ditujukan untuk mahasiswa Universitas Islam Indonesia agar mempermudah dalam menjalankan perkuliahan dan aktivitas kampus lainnya. Pada aplikasi *mobile* ini terdapat beberapa fitur utama yang nantinya bisa digunakan oleh mahasiswa, yaitu fitur Reminder, fitur Schedule, fitur What's On UII dan fitur e-KTM (Kartu Tanda Mahasiswa Elektronik). Adapun

tools dan *framework* yang digunakan untuk melakukan pengerjaan aplikasi mobile terdapat pada Tabel 3.1:

Tabel 3.1 *Tools* dan *Framework* yang digunakan

| No. | Bagian | Tools/Framework |
|-----|---|---------------------------------------|
| 1. | <i>Mobile</i> | <i>Framework</i> Flutter |
| 2. | <i>Backend</i> | <i>Framework</i> Lumen |
| 3. | <i>Text Editor/IDE</i> (Integrated Development Environment) | Visual Studio Code dan Android Studio |
| 4. | <i>Backend</i> Testing | Postman |
| 5. | Manajemen Basis Data | Dbeaver |
| 6. | <i>Version Control</i> | Gitlab |

Sebelum pemegang mulai dimasukkan ke dalam tim akselerasi, pemegang diwajibkan untuk mengikuti induksi terlebih dahulu yang diadakan oleh BSI UII dan dilaksanakan selama tiga hari mulai dari 5 September 2022 hingga 7 September 2022, dilanjutkan dengan mulai dimasukkan ke dalam tim di BSI dan memulai pengerjaan proyek yang sedang dikerjakan tim masing-masing, ada juga pemegang di BSI mengikuti kegiatan khusus BSI UII, yaitu Tech Talk yang diadakan setiap seminggu sekali pada hari Kamis dan akan membahas teknologi-teknologi baru yang saat ini sedang banyak digunakan, selain itu di akhir program magang juga ada presentasi akhir, yaitu pemegang harus mempresentasikan apa saja kegiatan yang dilakukannya selama mengikuti program ini di BSI UII, pemegang juga mempresentasikan hasil dari aplikasi yang sudah dibuat, lalu dilanjutkan dengan evaluasi yang dilakukan oleh manajemen dan Tim Human Capital Development untuk menilai setiap pemegang berdasarkan hasil presentasi dan performa yang dikeluarkan dalam melakukan aktivitas magang. Tabel 3.2 menunjukkan ringkasan aktivitas magang yang dilakukan oleh pemegang selama delapan bulan:

Tabel 3.2 Ringkasan aktivitas magang di BSI UII

| No. | Aktivitas | Durasi Waktu |
|-----|--|--------------------------------|
| 1. | Induksi magang | 5 – 7 September 2022 |
| 2. | Survey Kebutuhan Calon Pengguna Aplikasi I'm UII | 8 – 30 September 2022 |
| 3. | Pengembangan aplikasi I'm UII | 1 Oktober 2022 – 30 April 2023 |

| | | |
|----|--------------------------------------|----------------------|
| 5. | <i>Meeting</i> Tim Akselerasi | 1– 2 kali perbulan |
| 6. | Tech talk BSI | 1 kali setiap minggu |
| 7. | Presentasi akhir dan evaluasi magang | 13 April 2023 |

3.2.1 Induksi Magang BSI UII

Sebelum masuk ke proyek utama tim akselerasi, pemegang mengikuti kegiatan wajib yang diadakan oleh BSI terlebih dahulu, yaitu kegiatan induksi yang dilaksanakan pada tanggal 5 September 2022 dan berlangsung selama 3 hari hingga tanggal 7 September 2022. Kegiatan ini wajib diikuti oleh seluruh anggota magang dan karyawan baru BSI.

Kegiatan induksi magang BSI berlokasi di Gedung GBPH Prabuningrat – Rektorat UII pada Ruang Erasmus lantai 2. Di dalam kegiatan ini pemegang diberikan bekal ilmu serta pengetahuan tentang BSI dan budaya kerja di dalamnya. Pada hari pertama materi induksi yang diberikan adalah pengetahuan tentang budaya UII di BSI, tentang BSI UII serta regulasi dan etika di dalamnya, lalu hari kedua dilanjut dengan pemberian materi tentang metode SCRUM yang digunakan BSI untuk manajemen proyek yang dilakukan lalu materi tentang *tools* yang dipakai untuk mendukung metode SCRUM seperti Jira, Confluence, Kanban, dan *tools* lainnya, serta materi tentang karakteristik seorang profesional dan kultur Islam di BSI. Terakhir di hari ketiga diberikan materi tentang beberapa proyek yang telah dikerjakan oleh BSI sekaligus pengenalan tim yang ada di BSI yang mengerjakan proyek-proyek tersebut. Untuk runtutan acara induksi yang dilaksanakan oleh BSI UII dapat dilihat pada Tabel 3.3 Runtutan acara induksi BSI UII.

Tabel 3.3 Runtutan acara induksi BSI UII

| Senin, 5 September 2022 | | | | |
|-------------------------|--|---------------|---------------------------------|------------------------------|
| Topic: ke-UII-an | | | | |
| No. | Materi | Waktu | Pemateri | Tempat |
| 1 | Opening | 09.30 – 10.00 | Kholid Haryono, S.T., M.Kom. | Ruang Erasmus Lantai 2 |
| 2 | BSI product knowledge (Support) dan layanan prima | 10.00 – 11.00 | Endro Mustofa, S.Kom. | |
| 3 | Regulation and Ethics | 11.00 – 12.00 | Fitria Prihatini, S.T. | |
| 4 | Ke-UII-an | 13.00 – 15.00 | Dr. Mukhammad Andri Setiawan | |

| Selasa, 6 September 2022 | | | | |
|--------------------------|----------------------------|---------------|---------------------------------|------------------------------|
| Topic: Working Culture | | | | |
| No. | Materi | Waktu | Pemateri | Tempat |
| 1 | SCRUM Philosopy | 09.30 – 10.45 | Scrum Master | Ruang Erasmus Lantai 2 |
| 2 | Kanban & Jira - Confluence | 10.45 – 12.00 | Scrum Master | |
| 3 | Develop a Champion Culture | 13.00 – 14.30 | Kholid Haryono, S.T., M.Kom. | |
| 4 | Islamic Culture | 14.30 – 16.00 | Ari Sujarwo, S.Kom., M.I.T. | |

| Rabu, 7 September 2022 | | | | |
|---|--|---------------|---|------------------------------|
| Topic: Competency (Development Environment) | | | | |
| No. | Materi | Waktu | Pemateri | Tempat |
| 1 | UIIGateway (Dev Architecture & Microservice) | 09.00 – 10.30 | Andi Purwanto, S.T. | Ruang Erasmus Lantai 2 |
| 2 | DB & Interoperability (Infra Dev) | 10.30 – 12.00 | Pandu Bangun Asmoro, S.Kom. | |
| 3 | NDC & OPS (Security) | 13.00 – 14.30 | Akhmad Rafiuddin, S.Kom. Frendi Yusroni Romadhona, S.Kom. | |
| 4 | Closing | 14.30 – 16.00 | Ari Sujarwo, S.Kom., M.I.T. | |

3.2.2 Survey Kebutuhan Calon Pengguna Aplikasi I'm UII

Setelah pemegang mengikuti kegiatan induksi pada tiga hari pertama, tanggal 8 September pemegang mengikuti kegiatan *meeting* yang diadakan oleh supervisor untuk menentukan tim mana yang akan dimasuki pemegang agar bisa segera memulai pengerjaan proyek BSI. Setelah selesai pelaksanaan *meeting* dengan supervisor, hasilnya menentukan pemegang untuk masuk ke dalam tim akselerasi dan bisa mulai mengerjakan proyek aplikasi *mobile* I'm UII, yaitu sebuah aplikasi yang diperuntukkan bagi mahasiswa Universitas Islam Indonesia dan saat ini sedang dikerjakan oleh tim akselerasi BSI UII.

Tugas pertama yang didapat oleh pemegang adalah untuk mempelajari *framework* Flutter dan melakukan survey pengalaman pengguna untuk menentukan gambaran umum aplikasi yang akan dikembangkan dan menentukan fitur mana yang akan diangkat ke dalam aplikasi. Survey pengalaman pengguna merupakan tahap inisiasi dari proyek yang dikerjakan pemegang

dan durasi yang diberikan untuk melakukan survey adalah selama tiga minggu. Tujuan dari melakukan survey ini adalah untuk mengetahui informasi dari mahasiswa UII yang nanti akan menjadi pengguna dari aplikasi *mobile* ini, informasi-informasi ini akan dibentuk menjadi sebuah data dengan keluaran sebagai User Persona. User Persona merupakan karakter fiksi yang digunakan untuk merepresentasikan tujuan, kebutuhan, dan karakteristik dari banyak pengguna dan setiap persona akan mewakili pengguna dengan karakter yang mirip (Dicoding Indonesia, n.d.-b), nantinya User Persona ini akan menjadi bahan acuan bagi tim produk untuk menentukan fitur-fitur mana yang akan dikembangkan dan dimasukkan ke dalam aplikasi *mobile*.

Pada proses melakukan survey pengalaman pengguna, pemegang menggunakan metode wawancara secara langsung ke calon pengguna aplikasi nanti, yakni para mahasiswa UII. Sebelum melakukan wawancara, pemegang membuat daftar pertanyaan terlebih dahulu agar wawancara yang dilakukan terarah, daftar pertanyaan dibuat dengan menggunakan metode Hook Model. Hook Model merupakan sebuah metode yang bertujuan untuk menciptakan keterlibatan pengguna pada suatu produk (Lukyanchikova et al., 2023), sehingga mereka akan terus datang kembali ke produk tersebut tanpa perlu diiklankan secara agresif atau mahal. Dengan demikian, pengguna akan terlibat secara aktif dengan produk tersebut tanpa perlu dipaksa atau diajak. Terdapat empat tahap di dalam hook model, pertama ada *trigger* merupakan hal apa saja yang menyebabkan seseorang untuk melakukan suatu aksi, kedua ada *action*, yaitu aksi apa yang dilakukan oleh pengguna, lalu yang ketiga *reward* berupa hasil yang didapat setelah melakukan aksi tersebut, dan terakhir yang keempat adalah *investment* merupakan hal yang dilakukan seseorang untuk meningkatkan aksi yang dilakukan pada *cycle* berikutnya (Tantular Tezar et al., 2017). Berdasarkan metode Hook Model tersebut dan berpacu pada empat tahap di dalamnya pemegang membuat daftar pertanyaan yang menjadi bahan untuk wawancara. Daftar pertanyaan yang telah dibuat oleh pemegang dapat dilihat pada Tabel 3.4 dan akan menjadi acuan untuk wawancara ke calon pengguna aplikasi.

Tabel 3.4 Daftar pertanyaan wawancara

| No. | Pertanyaan |
|-----|--|
| 1. | Apa keinginan Anda saat memutuskan kuliah di UII? |
| 2. | Pernahkah Anda lupa untuk mengikuti sesi kuliah daring maupun luring? |
| 3. | Jika pernah, kenapa Anda bisa lupa? Jika tidak, apa yang Anda lakukan untuk mengingat sesi kuliah itu? |

| | |
|-----|---|
| 4. | Apa yang anda khawatirkan pada saat melakukan kegiatan kuliah? |
| 5. | Sebagai mahasiswa, apa kendala yang anda alami selama kegiatan perkuliahan? |
| 6. | Bagaimana caramu untuk mengatasi kendala tadi? |
| 7. | Dari berbagai layanan digital mahasiswa yang ditawarkan UII, layanan apa saja yang sering anda gunakan? |
| 8. | Kenapa Anda memilih menggunakan layanan tersebut? |
| 9. | Kapan dan Kenapa Anda biasa mencari informasi tentang perkuliahan? (Contoh : KKN, PNDI, ONDI, UTS, dan UAS) |
| 10. | Biasanya, apa yang Anda lakukan untuk mencari informasi tersebut? |
| 11. | Jika ada aplikasi smartphone untuk mahasiswa UII, apa saja fitur yang anda harapkan? |
| 12. | Untuk apa anda menginginkan fitur itu? |

Setelah membuat pertanyaan, pemegang melakukan wawancara kepada calon pengguna aplikasi *mobile*, yaitu mahasiswa UII. Saat pelaksanaan wawancara pemegang saling membagi tugas ke anggota yang lain untuk berpencar melakukan wawancara, pemegang mendapat tugas untuk mewawancarai mahasiswa yang berada di Fakultas Hukum, Fakultas Teknologi Industri, Fakultas Matematika dan Ilmu Pengetahuan Alam, dan Fakultas Teknik Sipil dan Perencanaan. Wawancara dilakukan dengan menggunakan pedoman pertanyaan yang telah dibuat sebelumnya menggunakan metode Hook Model dengan tujuan agar bisa mengetahui kendala yang sedang dirasakan mahasiswa, apa saja upaya mahasiswa untuk mengatasi kendala tersebut, tujuan-tujuan mahasiswa melakukan perkuliahan, serta fitur-fitur apa saja yang diharapkan oleh mahasiswa untuk ada di aplikasi *mobile* dalam hal membantu perkuliahan mereka. Setelah mendapatkan informasi dari berbagai narasumber dengan latar belakang yang berbeda-beda, pemegang mengumpulkan informasi-informasi tersebut ke dalam dokumen Excel dan membuat kesimpulannya dengan *output* berupa User Persona.

3.2.3 Perancangan Basis Data

Fitur-fitur yang ada di aplikasi I'm UII tidak akan berjalan apabila tidak ada proses CRUD di dalamnya, maka dari itu basis data diperlukan. Sebelum mulai melakukan pemrograman *backend* dengan menggunakan Lumen, pemegang perlu merancang basis datanya terlebih dahulu. Hal pertama yang dilakukan ketika merancang basis data adalah

dengan membuat rancangan API *contract*, rancangan ini merupakan sebuah struktur *body* JSON yang nanti akan digunakan untuk ditampilkan ke bagian *mobile* agar dapat diolah. Dengan rancangan API *contract* ini juga pemegang bisa tahu apa saja tabel yang akan dibuat di basis data nanti dan kolom-kolom apa saja yang akan dimasukkan ke dalam tabel tersebut. Dalam pembuatan API *contract* pemegang perlu mengetahui dengan betul variabel apa saja yang perlu ditampilkan saat memunculkan struktur JSON ini ke bagian *mobile* untuk digunakan di sana, karena ada sebagian variabel yang tidak boleh ditunjukkan ke bagian *mobile* karena masalah keamanan contohnya adalah tidak boleh menunjukkan variabel ID apapun ke bagian *mobile* karena khawatir dengan ID tersebut bisa digunakan untuk melakukan hal berbahaya ke dalam basis data. Setelah selesai dalam pembuatan API *contract*, tugas ini perlu untuk di-*review* terlebih dahulu oleh *team leader* akselerasi untuk dicari bagian apa saja yang sekiranya belum tepat serta solusinya.

Proses *review* yang dilakukan berlangsung selama satu pertemuan yang biasa dilakukan saat *progress meeting*, setelah proses *review* selesai saatnya pemegang untuk mengimplementasikan hasil dari API *contract* tersebut ke dalam basis data. Hasil tangkapan layar dari bagian API *contract* untuk mengambil atau GET daftar *reminder* dapat dilihat pada Gambar 3.2 API *contract* untuk GET daftar *reminder*. Metode GET dalam hal ini merupakan sebuah *endpoint* yang akan digunakan untuk mengambil *list* atau daftar *reminder* milik pengguna, selanjutnya terdapat parameter yang digunakan ketika menggunakan *endpoint* ini, yaitu parameter *user_uuid* yang merupakan *uuid* milik user untuk digunakan ketika mengambil daftar *reminder* sehingga daftar yang diambil hanya milik pengguna dengan *uuid* itu saja. Terakhir ada *response* yaitu, struktur *body* JSON yang akan ditampilkan nanti di *mobile*.

Untuk perancangan basis data pemegang membuatnya berdasarkan *response*, dari API *contract* yang telah dibuat dan untuk *tools* yang digunakan untuk memulainya adalah dengan menggunakan XAMPP dan Phpmyadmin untuk merancang basis data. Pembuatan rancangan basis data ini merupakan hal yang penting untuk dilakukan karena saat melakukan komunikasi dengan API akan mengirimkan *request* berupa *query* ke dalam basis data untuk melakukan metode seperti GET ketika ingin mengambil daftar data. Pemegang membuat basis data dengan merancang tabel yang ingin dibuat, tabel basis data merupakan koleksi dari data yang disimpan dalam format tabel berupa baris dan kolom pada basis data, bagian baris akan menyimpan *record* dari data dan kolom akan menyimpan atribut dari *record* tersebut. Pemegang mulai membuat tabel dengan menentukan tabel-tabel yang hendak dibuat, yaitu tabel untuk *reminder*, tabel untuk *user*, dan terakhir tabel untuk relasi *user* dan *reminder*.

:: GET List Reminder

Mengambil list reminder

Parameter:

- ▶ `user_uuid` dalam bentuk `string`

Response:

```
{
  "data" :[
    {
      "uuid": "123123423894",
      "title": "Tagihan SPP 2022/2023",
      "category": "Tagihan",
      "start_time": "2022-10-16 09:00:00.000Z",
      "end_time": "2022-11-10 12:00:00.000Z",
      "color": "0xFF68C29",
      "is_completed": false,
      "is_editable": false,
      "is_pinned": true
    }
  ]
}
```

Gambar 3.2 API *contract* untuk GET daftar *reminder*

Tabel *reminder* akan digunakan untuk fitur *reminder* dan tabel *user* akan digunakan untuk mengurus pengguna aplikasi, sedangkan untuk tabel relasi *user* dan *reminder* akan digunakan sebagai relasi dari *reminder* yang ada dengan penggunanya seperti untuk proses mendapatkan daftar *reminder* untuk pengguna, menghapus *reminder* milik pengguna, dan lain-lain. Kolom GPA pada tabel *user* merupakan kolom yang digunakan untuk menyimpan nilai IPK mahasiswa UII, kolom ini akan selalu di-*update* oleh BSI selaku yang memegang basis data inti UII dan kolom *user_update* yang menjadi acuan keamanan bahwa GPA akan di-*update* otomatis oleh sistem sesuai dengan basis data inti BSI UII sehingga kolom *user_update* akan terisi sistem yaitu dalam artian di-*update* oleh BSI UII. Untuk kolom *user_update* dan *updated_at* berlaku bagi setiap tabel di basis data yang digunakan pada aplikasi I'm UII karena sebagai bentuk standar keamanan di BSI untuk mengetahui siapa yang merubah data dan kapan dirubahnya, begitu pun dengan pembuatan data terdapat kolom *created_at* dan *user_input* untuk mengetahui kapan data dibuat dan siapa yang membuat data tersebut, sehingga jika ditemukan sebuah anomali jika ada data yang tidak sesuai diharapkan bisa terbantu dengan kolom-kolom tersebut. Lalu kolom *is_completed* dibuat sebagai penanda apakah *reminder* milik pengguna sudah selesai dilakukan atau belum, pada aplikasi pengguna diberikan opsi tombol sebagai

penanda apakah *reminder* tersebut sudah selesai atau belum, jika *reminder* tersebut sudah lewat dari waktu dan tanggal yang telah ditentukan maka dari kode Flutter telah diatur untuk menyelesaikan *reminder* tersebut.

Selanjutnya setelah menentukan tabel apa saja yang akan dibuat, selanjutnya adalah menentukan kolom apa saja yang akan dimasukkan pada setiap tabel, terakhir adalah menentukan relasi untuk setiap tabel. Relasi ini akan berguna ketika proses mengolah data, seperti jika ingin menghapus sebuah data, jika data tersebut terdapat relasi dengan tabel lain maka di tabel yang lain juga bisa ikut terhapus. Seluruh proses pembuatan basis data ini dipandu langsung oleh *team leader* jadi pemegang tidak perlu kebingungan apabila nanti terdapat kendala dalam pembuatannya. Contoh hasil dari pembuatan tabel basis data yang dibuat oleh pemegang bisa dilihat pada Gambar 3.3 Rancangan tabel basis data untuk fitur *reminder* di Phpmyadmin yang merupakan tabel beserta relasinya untuk bagian fitur Reminder.

The image shows three database tables from Phpmyadmin:

- mobile_reminder.user_reminder**:
 - id : bigint(20)
 - uuid : varchar(128)
 - user_uuid : varchar(128)
 - reminder_uuid : varchar(128)
 - title : varchar(255)
 - color : varchar(50)
 - repeated : varchar(50)
 - description : text
 - time_repeated : datetime
 - time : varchar(255)
 - user_type : enum('owner','guest')
 - is_editable : tinyint(1)
 - is_pinned : tinyint(1)
 - is_completed : tinyint(1)
 - user_input : varchar(128)
 - created_at : timestamp
 - user_update : varchar(128)
 - updated_at : timestamp
- mobile_reminder.reminder**:
 - id : bigint(20)
 - uuid : varchar(128)
 - user_input : varchar(128)
 - created_at : timestamp
 - user_update : varchar(128)
 - updated_at : timestamp
- mobile_reminder.user**:
 - id : bigint(20)
 - uuid : varchar(128)
 - name : varchar(255)
 - nim : varchar(255)
 - image_url : varchar(255)
 - email : varchar(255)
 - birth_date : datetime
 - faculty : varchar(255)
 - study_program : varchar(255)
 - gpa : float
 - address : text
 - barcode : varchar(255)
 - fcm : text
 - created_at : timestamp
 - updated_at : timestamp
 - user_input : varchar(128)
 - user_update : varchar(128)

Gambar 3.3 Rancangan tabel basis data untuk fitur *reminder* di Phpmyadmin

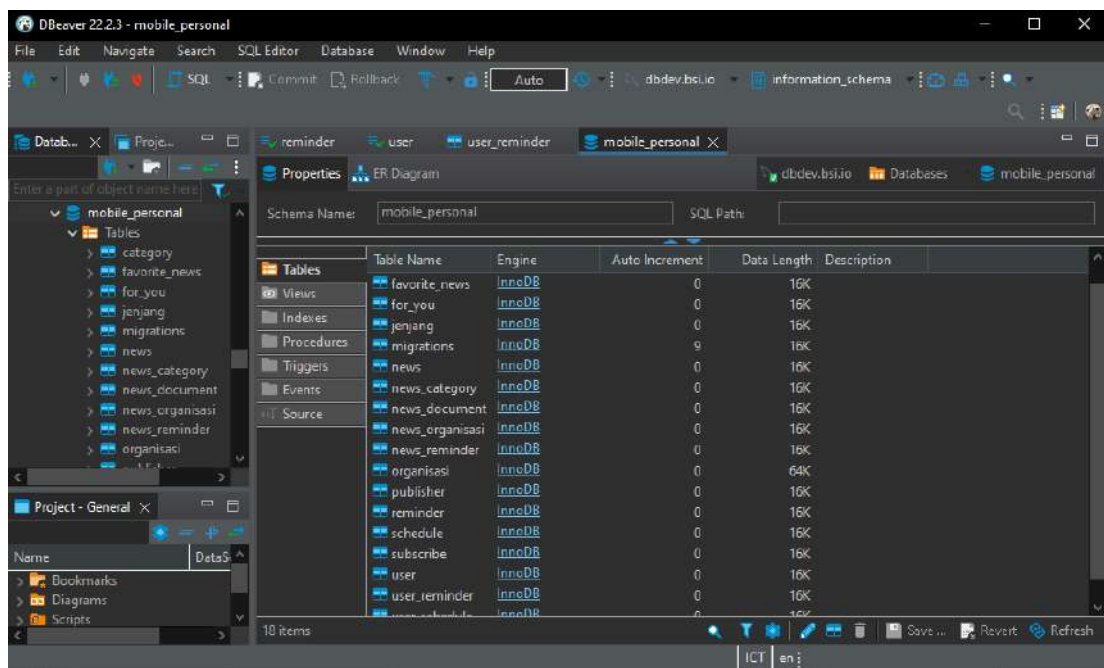
3.2.4 Pengembangan *Backend*

Sebelum lanjut untuk mengerjakan *backend* dengan Lumen, *team leader* menginformasikan untuk men-*deploy* tabel-tabel basis data yang ada pada Phpmyadmin ke DBDev. DBDev merupakan *environment* pengembangan milik BSI UII yang digunakan sebagai tempat menyimpan basis data untuk proyek-proyek yang ada di BSI, namun *environment* ini hanyalah khusus untuk pada tahap pengembangan saja jadi tidak

mempengaruhi *environment* basis data utama yang digunakan oleh BSI dalam menjalankan seluruh proyek-proyeknya. Untuk akses dan manajemen ke *environment* DBDev pemegang menggunakan sebuah *tools* bernama DBeaver yang merupakan aplikasi *multi-platform* sebagai *tools* untuk manajemen basis data. Tangkapan layar pada aplikasi DBeaver untuk basis data aplikasi I'm UII setelah di-*deploy* ke DBDev dapat dilihat pada Gambar 3.4.

Selanjutnya setelah berhasil melakukan *deploy* basis data ke *environment* DBDev, pemegang melanjutkan tugas dalam pembuatan API yang kali ini menggunakan *framework* Lumen. Dalam pembuatan API terdapat metode-metode HTTP yang digunakan untuk memproses *request* yang diminta ke basis data, API sendiri merupakan sebuah mekanisme yang memungkinkan dua komponen perangkat lunak untuk saling berkomunikasi satu sama lain. Untuk metode HTTP *request*-nya terdapat empat macam yang digunakan oleh pemegang untuk pembuatan API aplikasi I'm UII pada fitur *reminder* dan profil, yaitu:

- Metode GET yang digunakan untuk meminta data dari basis data.
- Metode POST untuk mengirim data baru ke server dan server membuat data baru di basis data.
- Metode PUT yang hampir mirip seperti POST hanya saja metode ini untuk memperbarui data yang sudah ada di basis data.
- Metode DELETE untuk menghapus data yang ditentukan di basis data.



Gambar 3.4 Basis data untuk aplikasi I'm UII melalui DBeaver

Untuk metode GET terdapat tiga jenis *request* yang akan digunakan pada fitur yang pemegang kembangkan, yaitu GET daftar *reminder*, GET detail *reminder*, dan GET data profil pengguna untuk digunakan pada halaman profil dan e-KTM. Selanjutnya adalah metode POST untuk menambahkan *reminder* baru milik pengguna, lalu metode PUT terdapat tiga *request* yang akan dikirimkan, yaitu PUT untuk mengedit *reminder*, PUT untuk pin *reminder*, dan PUT untuk menyelesaikan *reminder*. Terakhir adalah metode DELETE yang digunakan untuk menghapus *reminder*, terdapat dua jenis dalam metode DELETE ini, yang pertama adalah pengguna bisa menghapus seluruh *reminder* yang berulang atau yang kedua pengguna bisa menghapus hanya salah satu dari *reminder* yang berulang tersebut.

Pada metode PUT *reminder* terdapat kolom yang perlu diisi yaitu kolom *repeated*, kolom ini berfungsi untuk menentukan apakah *reminder* yang dibuat berulang atau tidak, jika berulang apakah berulang setiap hari, setiap minggu, atau setiap bulan. Untuk kolom *repeated* nanti akan diisi dengan nilai *varchar* berupa setiap hari, setiap minggu, atau setiap bulan, dari nilai-nilai tersebut nanti akan dimasukkan ke dalam kode logika pada *framework* Lumen untuk dilakukan *query* yang akan diinput berulang ke basis data sesuai dengan tanggal dan waktu yang dimasukkan ke dalam *reminder*. Hasil kode logika pemrograman di Lumen untuk kolom *repeated* sebuah *reminder* dapat dilihat pada Gambar 3.5 Kode untuk mengulang *reminder* berdasarkan nilai pada kolom *repeated*.

```

$startTimeFirst = null;
if($data['repeated'] == 'Tidak diulangi') {
    $startTimeFirst = strtotime($data['time']);
}
else {
    $startTimeFirst = strtotime($firstReminder->information->time);
}

$startFormatted = date('Y-m-d H:i:s', $startTimeFirst);
$startDate = new DateTime($startFormatted);
$copyStartTime = clone $startDate;
$intervalDay = null;
$dateRange = null;

if($data['repeated'] == 'Tiap hari') {
    $intervalDay = $copyStartTime->add(new DateInterval('P30D'));
    $dateRange = new DatePeriod(
        $startDate, new DateInterval('P1D'), $intervalDay);
}
else if($data['repeated'] == 'Tiap minggu') {
    $intervalDay = $copyStartTime->add(new DateInterval('P30D'));
    $dateRange = new DatePeriod(
        $startDate, new DateInterval('P7D'), $intervalDay);
}
else if($data['repeated'] == 'Tiap bulan') {
    $intervalDay = $copyStartTime->add(new DateInterval('P1Y'));
    $dateRange = new DatePeriod(
        $startDate, new DateInterval('P1M'), $intervalDay);
}
else if($data['repeated'] == 'Tiap tahun') {

```

```

    $intervalDay = $copyStartTime->add(new DateInterval('P10Y'));
    $dateRange = new DatePeriod(
        $startDate, new DateInterval('P1Y'), $intervalDay);
}
else if($data['repeated'] == 'Tidak diulangi') {
    $intervalDay = $copyStartTime->add(new DateInterval('P1D'));
    $dateRange = new DatePeriod(
        $startDate, new DateInterval('P1D'), $intervalDay);
}
}

```

Gambar 3.5 Kode untuk mengulang *reminder* berdasarkan nilai pada kolom *repeated*

Dalam proses pembuatannya pemegang menggunakan aplikasi *code editor* berupa Visual Studio Code, sedangkan untuk melakukan pengujian API yang telah dibuat menggunakan aplikasi Postman. Postman merupakan aplikasi yang sering digunakan oleh developer untuk membantu melakukan pengembangan serta pengujian *backend*. Dengan Postman pemegang bisa melakukan testing API dengan mengirimkan *request* apapun dan melihat hasilnya pada bagian *response* yang disediakan oleh Postman, jika ada *error* yang terjadi pada Lumen akan terlihat pada bagian *response* di Postman untuk detail dari masalah *error*-nya. Contohnya pada Gambar 3.6 Kode untuk *query DELETE reminder*, terdapat sebuah kode untuk melakukan *query* ke basis data dalam melakukan proses hapus *reminder* milik pengguna.

```

public function remove(
    UserPVReminderDomainModel $reminder, string $user_reminder_uuid,
    string $delete_type, string $user_uuid, string $user_type
)
{
    if($delete_type == 'all-event') {
        if($user_type == 'guest') {
            DB::table('user_reminder')
                ->where(
                    'reminder_uuid',
                    $reminder->information->reminder_uuid
                )
                ->where('user_uuid', $user_uuid)
                ->delete();
        }
        else if($user_type == 'owner') {
            DB::table('reminder')
                ->where('uuid', $reminder->information->reminder_uuid)
                ->delete();

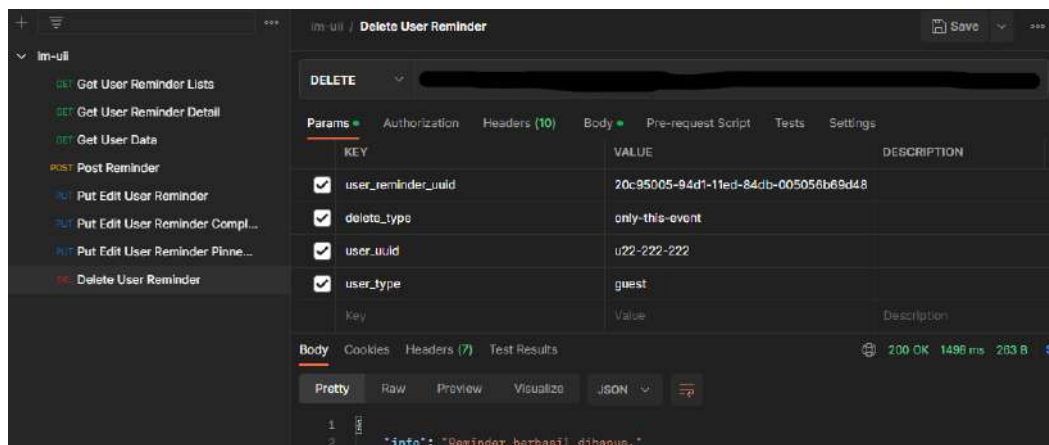
            DB::table('user_reminder')
                ->where(
                    'reminder_uuid',
                    $reminder->information->reminder_uuid
                )
                ->delete();
        }
    }
    else if($delete_type == 'only-this-event') {
        DB::table('user_reminder')
            ->where('uuid', $user_reminder_uuid)
            ->delete();
    }
}

```


Gambar 3.6 Kode untuk *query* DELETE *reminder*

Pada sebuah *reminder* terdapat fitur untuk menambahkan orang ke dalam *reminder* tersebut, untuk orang yang membuat *reminder*-nya disebut sebagai *owner* dan orang yang ditambahkan ke dalam *reminder* tersebut dinamakan sebagai *guest*, pada basis data nanti akan tersimpan data *uuid* pengguna untuk *owner* yang membuat *reminder* tersebut dan *uuid* pengguna untuk *guest* yang diundang ke dalam *reminder* tersebut, kedua tipe *owner* dan *guest* ini nantinya akan berdampak ke proses hapus *reminder* yang terjadi di aplikasi. Untuk pengguna dengan status *owner* dapat menghapus seluruh *reminder* yang ada dan akan berdampak ke *guest* yang ditambahkan, yaitu *reminder* yang ada di *guest* juga akan ikut terhapus apabila *reminder* tersebut merupakan *reminder* yang dibuat oleh *owner*, sedangkan untuk *guest* jika menghapus seluruh *reminder* tidak akan berdampak ke *guest* yang lainnya serta *owner* dan hanya akan menghapus *reminder* yang ada di *guest* tersebut saja. Begitupun dengan jika *owner* atau *guest* hanya ingin menghapus satu *reminder* saja. Untuk dokumentasi API biasanya dari BSI dilakukan pada platform seperti Jira dan Confluence namun karena pemegang tidak diberikan akses khusus pada platform Confluence maka pemegang hanya bisa membuat dokumentasi API menggunakan Postman.

Seperti yang telah dijelaskan sebelumnya untuk melakukan testing API, pemegang menggunakan aplikasi Postman seperti yang terlihat pada Gambar 3.7 Testing *endpoint* untuk DELETE *reminder* di bawah ini dalam melakukan testing *endpoint* untuk DELETE *reminder*.

Gambar 3.7 Testing *endpoint* untuk DELETE *reminder*

3.2.5 Pengembangan Fitur Reminder, Profil, dan e-KTM

Tim Akselerasi memberikan tugas lain yang diberikan ke pemegang, yaitu mempelajari Flutter serta melakukan pemasangan IDE dan *framework* Flutter ke laptop pemegang masing-

masing. IDE atau *Integrated Development Environment* yang digunakan oleh pemegang adalah Visual Studio Code dan Android Studio, kedua IDE tersebut sudah sangat bagus untuk memulai mengembangkan aplikasi menggunakan *framework* Flutter. Namun, sebelum memulai menggunakan IDE tersebut perlu memasang *plugin* terlebih dahulu, yaitu *plugin* khusus untuk *framework* Flutter dan bahasa pemrograman Dart.

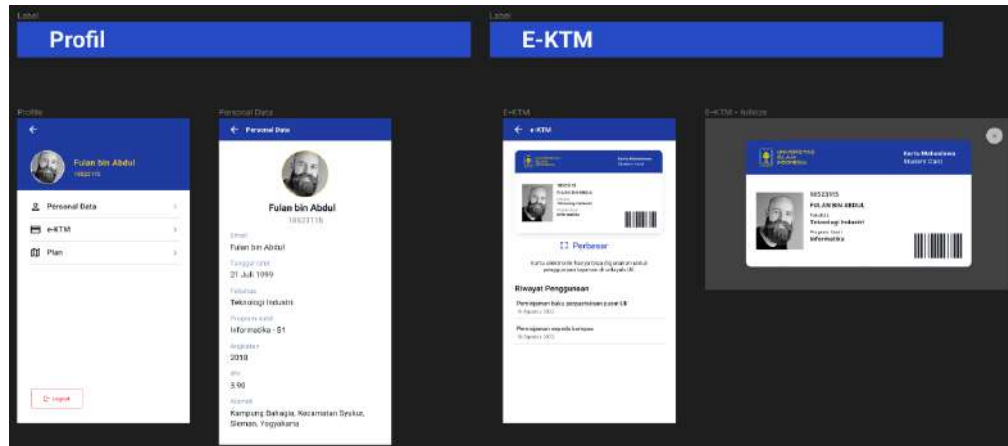
Alasan dari pemegang untuk memulai mempelajari Flutter, yaitu karena pemegang sebelumnya belum pernah mengembangkan aplikasi menggunakan Flutter sehingga supervisor memberikan waktu tiga minggu untuk mempelajari *framework* Flutter sambil melakukan survey ke calon pengguna aplikasi. Untuk awalan pemegang mulai mempelajari dari fundamentalnya terlebih dahulu seperti bagian-bagian di dalam Flutter, *widget*, *stateful* dan *stateless widget*, *navigation*, *routing*, *state management*, dan lain-lain. Pemegang mempelajari fundamental Flutter tersebut di berbagai macam sumber, seperti platform Dicoding dengan mempelajari dua *course* pertama Flutter untuk pemula, yaitu *course* belajar membuat aplikasi Flutter untuk pemula dan *course* belajar fundamental aplikasi Flutter. Selain Dicoding pemegang juga banyak mempelajari Flutter dari video-video yang tersebar banyak di Youtube mengenai Flutter.

Selain mempelajari fundamental Flutter dari berbagai macam sumber yang berada di Internet, pemegang juga melakukan latihan membuat aplikasi sederhana untuk mengimplementasikan hasil dari mempelajari fundamental Flutter. Aplikasi sederhana ini bernama Kutubuku, yaitu sebuah aplikasi untuk melihat daftar buku. Di dalam aplikasi ini pengguna bisa melihat daftar buku, menambah buku baru, mengedit data buku yang sudah ada, dan menghapus buku, untuk bagian *backend* ini pemegang menggunakan ExpressJS dalam mengoperasikan CRUD (Create, Read, Update, Delete). Pada Flutter pemegang mengimplementasikan hasil pembelajaran dengan membangun tampilan antarmuka menggunakan kombinasi *widget* hingga mengimplementasikan API untuk proses CRUD dengan menggunakan *library* khusus untuk Flutter, yaitu *library* HTTP.

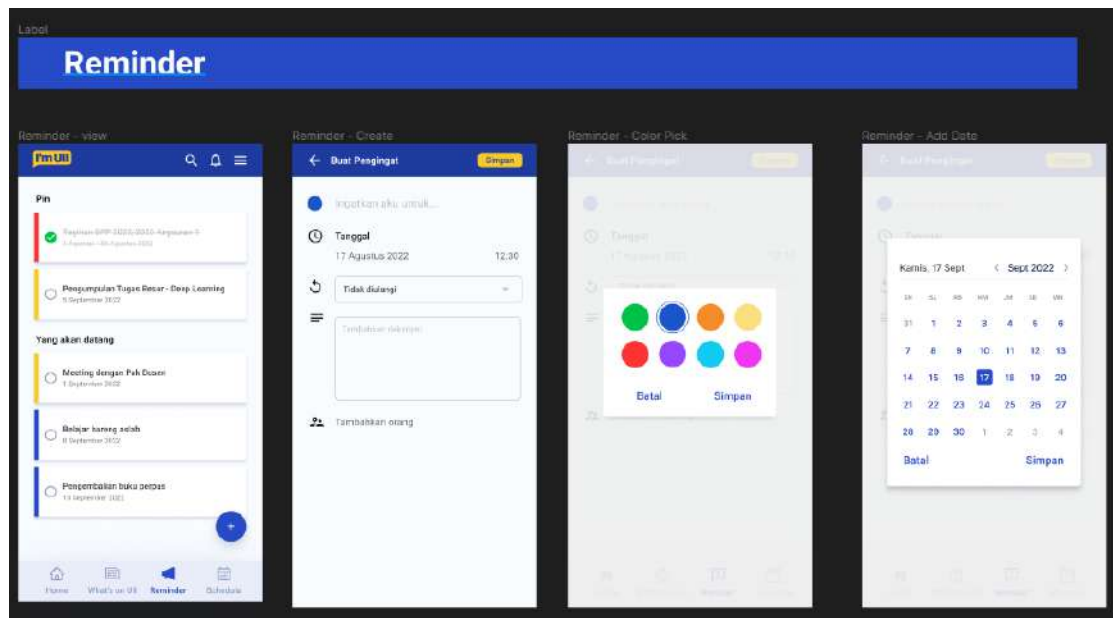
Setelah berhasil mempelajari Flutter, pemegang mulai melakukan tugas utama di tim akselerasi, yaitu mengembangkan aplikasi I'm UII. Setelah *output* dari hasil survey ke calon pengguna aplikasi diberikan ke anggota tim UX, tim UX memberikan rancangan desain antarmuka berupa *high fidelity wireframe* untuk diimplementasikan ke *frontend*, yaitu Flutter. Pada rancangan desain yang diberikan, terdapat tiga fitur utama yang akan diimplementasikan ke dalam aplikasi, yaitu yang pertama adalah fitur *what's on* merupakan fitur portal berita untuk seluruh informasi yang ada di UII, yang kedua adalah fitur *schedule* seperti kalender agar

pengguna bisa melakukan penjadwalan yang berisi beragam jadwal kegiatan mahasiswa UII, lalu yang ketiga adalah fitur *reminder* yang berfungsi sebagai pengingat hal-hal penting bagi mahasiswa UII. Selain dari ketiga desain fitur utama tersebut terdapat juga desain antarmuka untuk halaman *login* pengguna, halaman *splash screen*, halaman profil, dan halaman e-KTM (Kartu Tanda Mahasiswa Elektronik).

Desain dari tampilan halaman profil dan e-KTM yang telah dibuat oleh anggota bagian UX dapat dilihat pada Gambar 3.8 Tampilan desain halaman profil dan e-KTM. Pada halaman profil tersebut menampilkan data dari pengguna yang memakai aplikasi seperti nama, NIM, fakultas, dan lain-lain, namun sebelum masuk ke halaman data profil pengguna terdapat juga menu lain, yaitu e-KTM dan *plan*. Untuk fitur *plan* merupakan sebuah fitur yang direncanakan sebagai tempat bagi pengguna aplikasi, yaitu mahasiswa untuk membuat berbagai macam rencana perkuliahan yang mereka inginkan selama berkuliah di UII, namun fitur tersebut masih bagian dari rencana fitur yang akan diimplementasikan ke dalam aplikasi I'm UII dan belum ada desain tampilan antarmuka aplikasi yang dibuat oleh anggota bagian UX. Lalu untuk desain tampilan antarmuka fitur *reminder* dapat dilihat pada Gambar 3.9 Bagian dari tampilan desain halaman *reminder*. Pada fitur *reminder* tersebut pengguna akan diarahkan pertama kali ke bagian halaman daftar dari *reminder* yang pengguna miliki, nanti pengguna bisa melakukan aksi pin atau memasang *reminder* yang ada di daftar menjadi paling teratas sehingga memiliki makna kalau *reminder* yang dipin merupakan hal yang sangat penting. Selanjutnya, jika pengguna ingin membuat sebuah *reminder* akan diarahkan ke halaman buat *reminder* dan di dalamnya pengguna bisa mengisi judul, tanggal, setiap kapan saja *reminder* tersebut diulang, deskripsi dari *reminder* tersebut, pengguna juga bisa menambahkan orang lain ke dalam *reminder* yang dibuat agar orang lain juga bisa mendapatkan *reminder* yang sama, dan terakhir pengguna dapat memilih warna dari *reminder* yang dibuat sesuka hatinya.



Gambar 3.8 Tampilan desain halaman profil dan e-KTM

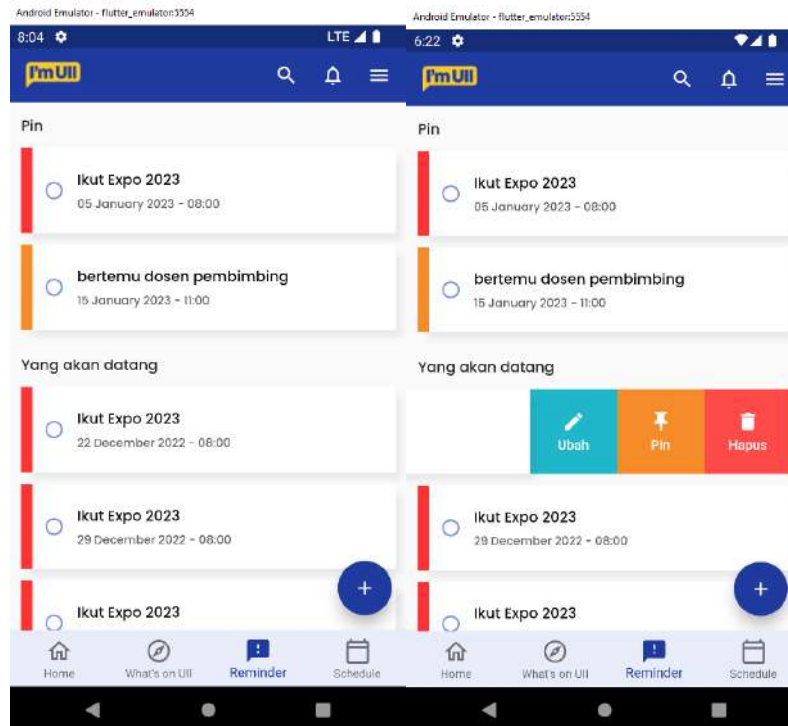


Gambar 3.9 Bagian dari tampilan desain halaman *reminder*

Dari seluruh rancangan desain yang diberikan, pemegang beserta rekan tim akselerasi lainnya mengkonfirmasi desain antarmuka tersebut dan mulai menjalani tugas untuk mengimplementasi rancangan desain antarmuka aplikasi I'm UII ke *framework* Flutter. Untuk tugas kali ini pemegang mendapatkan bagian mengerjakan bagian fitur *reminder*, e-KTM, dan halaman profil. Bagian pertama yang pemegang kerjakan adalah untuk membuat fitur *reminder*, pada fitur ini pemegang perlu untuk membuat bagian dari halaman daftar *reminder*, halaman untuk membuat dan mengedit *reminder* serta halaman untuk menambah dan mengedit orang yang ingin dimasukkan ke dalam *reminder*, fungsi untuk pin *reminder*, dan fungsi untuk hapus *reminder*. Untuk awalan pengguna menggunakan data *dummy* atau data buatan

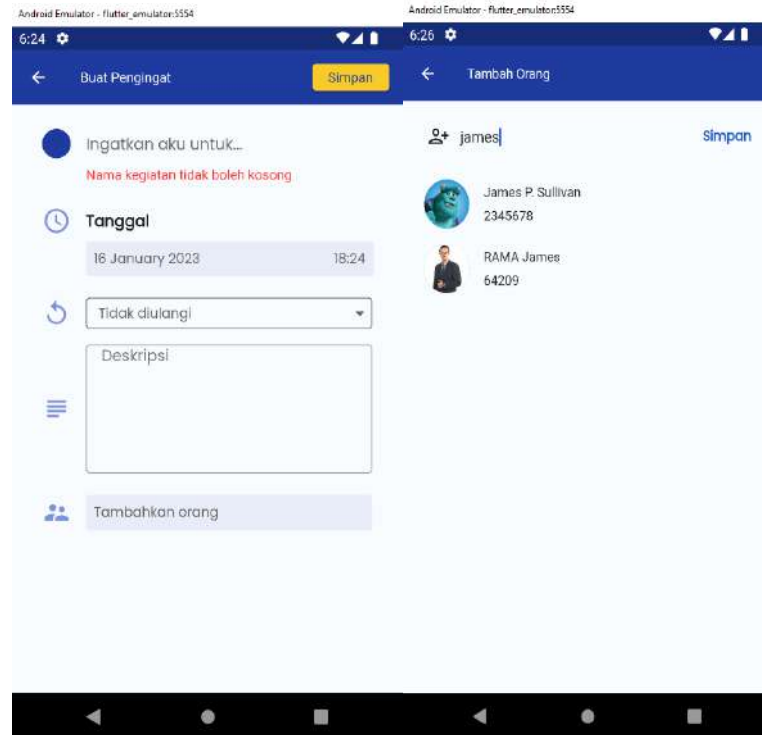
menggunakan *file* berformat JSON untuk tes menampilkan daftar *reminder*, *file* berformat JSON ini digunakan karena belum dibuat sebuah API yang bisa digunakan untuk mengambil data daftar *reminder* milik pengguna yang telah masuk ke aplikasi. Setelah dibuat *file* JSON, *file* tersebut dimasukkan ke dalam Flutter dengan dibuatkan modelnya terlebih dahulu agar bisa dibaca oleh Flutter dan setelah itu ditampilkan ke dalam tampilan antarmuka seperti yang terlihat pada Gambar 3.10 yang merupakan gambar dari tampilan antarmuka setelah diimplementasikan oleh pemegang pada Flutter. Pada halaman utama *reminder* menggunakan tampilan berupa *list* atau daftar karena sesuai dengan *design* yang diberikan oleh tim UI/UX untuk aplikasi I'm UII ini, penggunaan *list* ini juga untuk memanfaatkan fitur *swipe item* pada sebuah *list* yang menampilkan tombol opsi *reminder* dan juga untuk mempermudah pengguna dalam menerima informasi *reminder* berupa judul, tanggal beserta waktu *reminder*-nya.

Selanjutnya setelah berhasil membuat tampilan daftar *reminder*, pemegang melanjutkan untuk membuat tampilan menambahkan *reminder* baru. Untuk membuat halaman ini, pemegang hanya membuat tampilannya saja terlebih dahulu bagi pengguna untuk mengisi data *reminder*-nya seperti judul, tanggal, waktu diulangnya, deskripsi *reminder*, tombol simpan, dan tombol untuk mengarahkan ke halaman menambahkan atau mengedit orang ke dalam *reminder*. Sama halnya dengan halaman untuk mengedit *reminder* karena API untuk metode PUT atau *edit reminder* milik pengguna belum ada, maka pemegang hanya membuat halaman tampilannya saja yang isi halamannya sama seperti ketika pengguna melakukan aksi tambah *reminder* baru. Untuk tampilan halaman bagian tambah dan edit *reminder* yang telah diimplementasikan ke dalam Flutter bisa dilihat pada Gambar 3.11 dan Gambar 3.12.

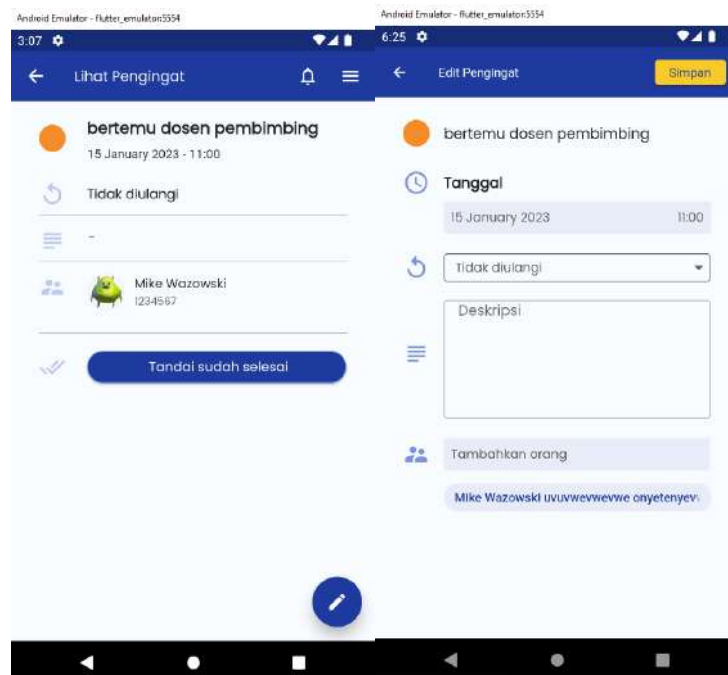


Gambar 3.10 Halaman daftar *reminder* dan menu pilihan ketika aksi *swipe*

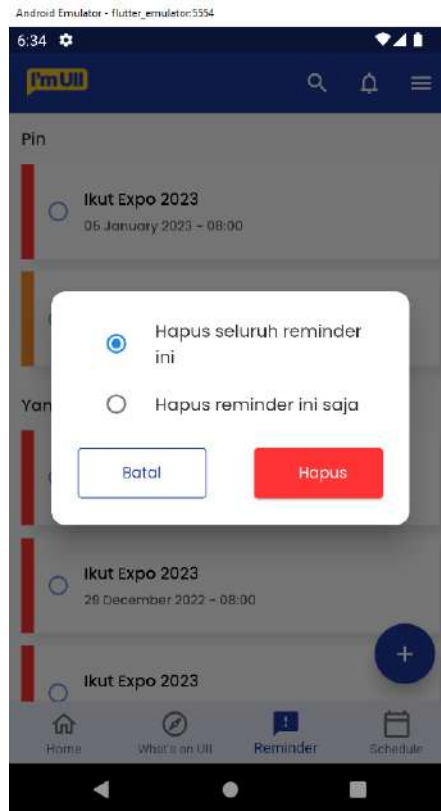
Setelah membuat halaman-halaman tersebut, selanjutnya pemegang membuat fungsi untuk menghapus *reminder* dengan membuat sebuah dialog konfirmasi yang diarahkan untuk pengguna ketika ingin melakukan aksi hapus *reminder* seperti yang terlihat pada Gambar 3.13, sehingga jika pengguna melakukan aksi *swipe* atau menggerakkan jari pada satu *item reminder* ke kiri seperti pada Gambar 3.10 Halaman daftar *reminder* dan menu pilihan ketika aksi *swipe* akan ada opsi aksi yang diberikan ke pengguna. Untuk konfirmasi dialog hapus tersebut pengguna akan diarahkan pada dua pilihan, jika ada *reminder* yang berulang seperti *reminder* yang berulang setiap minggunya, maka akan ada pilihan pertama, yaitu hapus satu *reminder* yang dipilih saja lalu yang kedua adalah pilihan untuk menghapus seluruh *reminder* yang berulang setiap minggunya tersebut. Lalu, fungsi terakhir adalah fungsi untuk pengguna melakukan aksi pin *reminder* agar ditaruh di paling atas pada halaman daftar *reminder*. Tombol untuk aksi pin ini ditaruh pada posisi seperti tombol hapus seperti yang terlihat pada Gambar 3.10 Halaman daftar *reminder* dan menu pilihan ketika aksi *swipe*.



Gambar 3.11 Halaman tambah *reminder* baru dan halaman untuk tambah orang



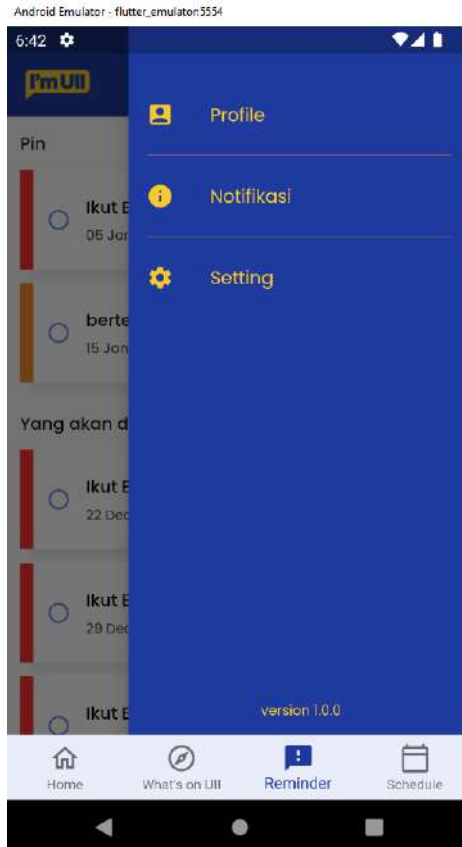
Gambar 3.12 Halaman detail dan ketika ingin mengubah *reminder*



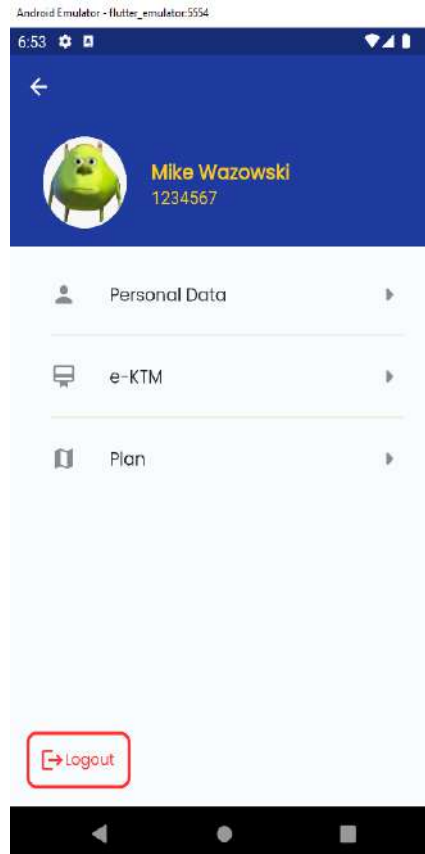
Gambar 3.13 Dialog konfirmasi ketika melakukan aksi hapus

Selesai mengimplementasikan desain tampilan antarmuka ke Flutter, selanjutnya pemegang melanjutkan pengerjaan untuk bagian kedua, yaitu membuat halaman data profil pengguna aplikasi dan e-KTM. Untuk profil pengguna pemegang membuat tampilan menu terlebih dahulu untuk memilih menu data profil pengguna, e-KTM dan *plan*, sama seperti desain yang telah dirancang. Untuk masuk ke menu tersebut pengguna perlu menekan tombol *sidebar* dengan *icon* garis tiga seperti terlihat pada Gambar 3.10 Halaman daftar *reminder* dan menu pilihan ketika aksi *swipe* yang terletak pada pojok kanan atas. Setelah tombol *sidebar* ditekan akan memunculkan tiga menu yang diberikan ke pengguna, yaitu menu *profile*, notifikasi, dan *setting* atau pengaturan. Untuk pilihan menu dapat dilihat pada Gambar 3.14 Menu pilihan *sidebar* dan jika pengguna menekan pilihan menu *profile*, pengguna akan diarahkan ke halaman menu profil yang menawarkan menu data profil pengguna, e-KTM, dan *plan* yang terlihat pada Gambar 3.15 Tampilan halaman profil.

Selanjutnya pemegang membuat tampilan untuk halaman data profil pengguna, pada halaman ini berisi data-data pengguna seperti gambar profil, nama pengguna, nomor induk mahasiswa, dan lain – lain. Sama seperti *reminder*, pemegang menggunakan data *dummy* dengan menggunakan *file* JSON dalam pembuatan tampilan halaman data profil pengguna.



Gambar 3.14 Menu pilihan *sidebar*

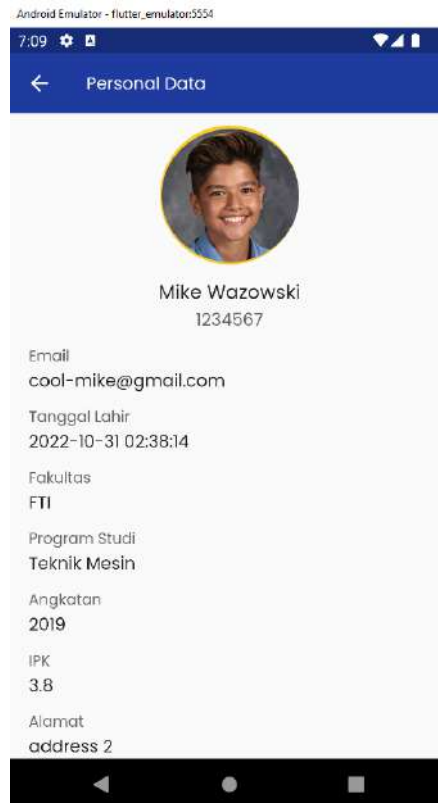


Gambar 3.15 Tampilan halaman profil

Selanjutnya pemegang membuat tampilan untuk halaman data profil pengguna, pada halaman ini berisi data-data pengguna, yaitu gambar profil, nama pengguna, nomor induk mahasiswa, alamat surat elektronik atau *e-mail*, tanggal lahir, fakultas, program studi, angkatan, nilai indeks prestasi kumulatif, dan alamat tinggal pengguna. Sama seperti *reminder*, pemegang menggunakan data *dummy* dengan menggunakan *file* JSON dalam pembuatan tampilan halaman data profil pengguna ini.

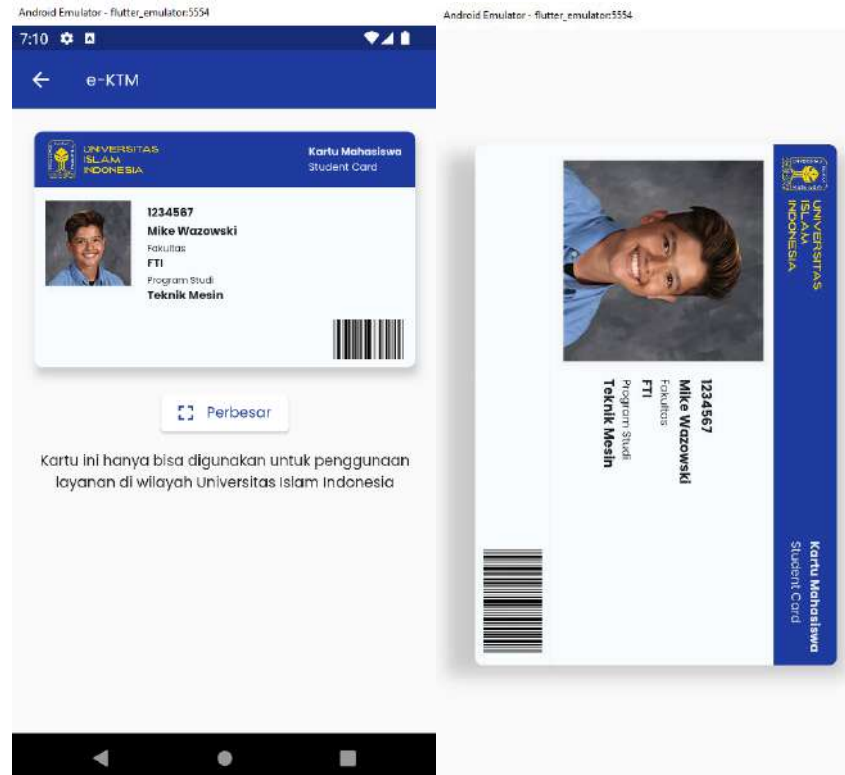
Setelah membuat halaman untuk profil dan data profil pengguna, selanjutnya pemegang membuat halaman untuk menu e-KTM atau kartu tanda mahasiswa elektronik. Pada halaman ini akan berisi tampilan dengan gambar KTM yang biasa dipakai di lingkungan UII, KTM ini juga memiliki *barcode* yang dapat digunakan mirip seperti KTM fisik milik mahasiswa. Tampilan dari e-KTM menggunakan Kartu Tanda Mahasiswa UII fisik sebagai tampilan utamanya untuk memvirtualisasikannya secara digital, sehingga informasi seperti untuk tahun masuk dan waktu valid dari KTM tersebut tidak ditampilkan karena belum ada pada tampilan di KTM fisik. Selain itu, terdapat juga tombol perbesar untuk memperbesar gambar dari KTM menjadi mode *landscape* sehingga pengguna dapat melihat e-KTM dengan jelas pada layar *smartphone* masing-masing.

Tampilan dari halaman data profil pengguna dan e-KTM yang telah diimplementasikan dapat dilihat pada Gambar 3.16 Halaman data profil pengguna dan Gambar 3.17 Halaman e-KTM dan tampilan ketika menekan tombol memperbesar



Gambar 3.16 Halaman data profil pengguna

Setelah pembuatan API sudah selesai, pemegang melanjutkan ke tugas berikutnya, yaitu melakukan implementasi hasil dari API yang sudah dibuat untuk dihubungkan dengan Flutter agar aplikasi *mobile* I'm UII bisa menampilkan dan mengolah data dari basis data. Pada Flutter, untuk melakukan komunikasi dengan API diperlukan sebuah *library* untuk membantu proses komunikasi dan melakukan berbagai macam *request* ke API, tim akselerasi telah menentukan *library* yang digunakan untuk membantu proses pengembangan pada tahap ini, yaitu *library* Dio. Dio merupakan sebuah *library* pada *framework* Flutter untuk melakukan *request* HTTP ke server dan menerima respon yang diberikan dari server. Dengan *library* ini, pemegang bisa melakukan berbagai macam *request* seperti GET, POST, PUT, dan DELETE.



Gambar 3.17 Halaman e-KTM dan tampilan ketika menekan tombol memperbesar

Hal pertama yang pemegang lakukan adalah dengan mengimplementasi metode *request* GET terlebih dahulu. Untuk melakukan proses *request* GET pada Dio, diperlukan sebuah fungsi pada Flutter yang dapat menangani proses tersebut, fungsi pada Dio tersebut bernama `dio().get()`. Fungsi `dio().get()` merupakan sebuah metode yang digunakan untuk mengirimkan *request* GET ke server. Pada fitur yang pemegang kerjakan, yaitu *reminder*, data profil, dan e-KTM, pemegang perlu mengimplementasikan tiga jenis *request* GET, yaitu GET daftar *reminder*, GET detail *reminder*, dan GET data profil. Khusus untuk GET data profil pengguna akan digunakan untuk dua halaman sekaligus, yaitu halaman data profil pengguna dan halaman e-KTM. Fungsi `dio().get()` juga bisa menerima *query* parameter, *header*, dan lainnya apabila *request* yang diminta memerlukan hal-hal tersebut, dalam kasus pemegang, pada fitur *reminder* diperlukan *user_uuid* atau kode unik yang hanya dimiliki oleh pengguna untuk *request* daftar *reminder*-nya, sehingga daftar *reminder* yang didapatkan adalah daftar *reminder* dengan *user_uuid* milik pengguna yang telah *login* pada aplikasi *mobile* I'm UII. Gambar 3.18 Implementasi *request* GET menggunakan Dio pada Flutter berikut merupakan hasil pemegang untuk melakukan *request* GET daftar *reminder* milik pengguna dengan menggunakan fungsi `dio().get()` pada *library* Dio di Flutter.

```

Future<ReminderModel> getListReminder(BuildContext context, String userUuid)
async {
  if (await Utils.checkConnection()) {
    try {
      String path = "${url}";
      final response = await Dio().get(
        path,
        queryParameters: {
          'user_uuid': userUuid,
        },
        options: Options(
          headers: {
            "x-app": "reminder",
            "x-menu": "reminder"
          }
        )
      );
      if(response.statusCode == 200){
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        return ReminderModel.fromJson(resData);
      } else {
        print('${response.statusCode} : ${response.data.toString()}');
        throw response.statusCode!;
      }
    }
  }
}

```

Gambar 3.18 Implementasi *request* GET menggunakan Dio pada Flutter

Setelah berhasil membuat *request* GET dengan Dio, pemegang membuat *class* model baru yang akan digunakan untuk menerima hasil *response* yang didapatkan setelah melakukan *request*. Model yang dibuat adalah untuk data daftar *reminder* yang kembalian *response* dari servernya dalam bentuk *List*, lalu model untuk data detail dari *reminder*, dan terakhir adalah model untuk pengguna. Tahap selanjutnya adalah dengan memanggil model tersebut ke dalam *widget* di Flutter untuk ditampilkan hasilnya ke pengguna, dalam implementasinya pemegang menggunakan *widget* bernama *FutureBuilder* untuk melakukan pemanggilan *request* yang sebelumnya telah dibuat dengan Dio lalu memunculkannya pada Flutter. Saat melakukan pemanggilan pemegang akan membuat kondisi yang jika *request* gagal dilakukan akan menampilkan *error* sedang terjadi ke pengguna, lalu kondisi untuk *request* yang berhasil namun tidak ada data yang muncul yang artinya pengguna belum membuat *reminder*, dan terakhir adalah kondisi saat *request* berhasil dilakukan dan data muncul, maka data tersebut akan ditampilkan. Untuk hasil dari data yang berhasil ditampilkan kurang lebih sama seperti pada untuk daftar *reminder*, Gambar 3.12 Halaman detail dan ketika ingin mengubah *reminder* untuk detail dari sebuah *reminder*, dan Gambar 3.16 Halaman data profil pengguna serta Gambar 3.17 Halaman e-KTM dan tampilan ketika menekan tombol memperbesar untuk data profil pengguna dan e-KTM.

Selanjutnya pemegang melanjutkan implementasi API pada Flutter untuk metode *request* POST. Fungsi yang akan digunakan pada Dio untuk mengatasi *request* POST adalah fungsi `dio().post()`, yaitu sebuah fungsi yang mirip seperti `dio().get()` namun perbedaannya adalah pada fungsi ini memerlukan sebuah *body* berformat JSON untuk mengirim data berupa *reminder* seperti judul *reminder*, deskripsi *reminder*, waktu berulang, tanggal, dan lainnya. Gambar 3.19 menunjukkan seperti apa implementasi POST yang menggunakan *body* untuk mengirim data.

```

try {
  String path = "${ReminderApiService.url}";
  final response = await Dio().post(
    path,
    options: Options(
      headers: {
        "x-app": "reminder",
        "x-menu": "reminder"
      }
    ),
    data: {
      "title": title,
      "color": color,
      "category": "personal",
      "description": description,
      "repeated": repeated,
      "time": time,
      "is_editable": true,
      "user_input": userInput,
      "user_update": userUpdate,
      "guests_uuid": guest,
    }
  );
}

```

Gambar 3.19 Penerapan *request* POST pada Dio di Flutter

Pada Gambar 3.19 Penerapan *request* POST pada Dio di Flutter terlihat terdapat parameter baru berupa data yang berfungsi untuk mengisikan data *reminder* yang nantinya akan dikirim ke alamat *endpoint* untuk *request* POST dalam format berupa JSON. Nanti server akan menerima respon tersebut dan akan mengirimkan isi data baru ke dalam basis data.

Selanjutnya adalah implementasi metode PUT untuk fitur *reminder*, yaitu PUT untuk mengedit *reminder*, PUT untuk pin *reminder*, dan PUT untuk menyelesaikan *reminder*. Pada Dio, fungsi yang digunakan adalah fungsi `dio().put()` yang mirip seperti fungsi `dio().post()` dengan membutuhkan sebuah data *body* yang akan dikirimkan ke server, hanya saja PUT digunakan untuk memperbarui data yang sudah ada sehingga dibutuhkan parameter berupa UUID atau kode unik milik *reminder* yang hendak diubah isinya. Begitupun dengan PUT pin *reminder* dan PUT menyelesaikan *reminder*, di dalam tabel basis data akan ada kolom bernama *is_pinned* dan *is_completed* yang berfungsi sebagai status apakah sebuah *reminder* memiliki

status pin atau tidak dan apakah sebuah *reminder* memiliki status sudah selesai atau belum. Bedanya dengan proses pada fungsi `dio().get()` untuk PUT edit *reminder* dengan PUT status pin dan status selesai adalah pada parameter datanya, untuk PUT status pin *reminder* hanya akan berisi data *is_pinned* saja dan begitu pula dengan PUT status selesai *reminder* hanya akan berisi data *is_completed* saja. Contoh hasil dari implementasi pemegang untuk API *request* PUT bisa dilihat pada berikut ini.

```
try {
    String path =
        "${ReminderApiService.url}public/api/v1/reminder/$userReminderUuid";

    final response = await Dio().put(
        path,
        options: Options(
            headers: {
                "x-app": "reminder",
                "x-menu": "reminder"
            }
        ),
        data: {
            "update_type": updateType,
            "title": title,
            "color": color,
            "category": "personal",
            "description": description,
            "repeated": repeated,
            "time_repeated": timeRepeated,
            "time": time,
            "is_editable": true,
            "user_input": userInput,
            "user_update": userUpdate,
            "guests_uuid": guest,
        }
    );
}
```

Gambar 3.20 *Request* edit *reminder* pada fungsi

Selanjutnya yang terakhir adalah metode *request* untuk DELETE, pada Dio akan menggunakan fungsi `dio().delete()` dengan parameter tambahan berupa UUID *reminder* yang ingin dihapus, tipe hapus yang akan dipakai, UUID milik pengguna, dan tipe pengguna apakah pengguna tersebut *guest* ataukah *owner*. Untuk tipe hapus yang akan digunakan adalah tipe hapus seperti pada Gambar 3.13 Dialog konfirmasi ketika melakukan aksi hapus yang berarti apakah hapus tersebut ingin menghapus seluruh *reminder* atau *reminder* yang dipilih saja. *Query* yang akan dijalankan nanti saat dikirimkan oleh server ke basis data adalah seperti yang dijelaskan pada Pengembangan *Backend* serta Gambar 3.6 Kode untuk *query* DELETE *reminder*, saat melakukan proses hapus akan melakukan cek sesuai kondisi dari data yang dikirimkan dari aplikasi. Contoh hasil dari implementasi pemegang bisa dilihat pada Gambar 3.21 *Request* hapus pada fungsi di Dio berikut ini.

```
try {
    String path = "${ReminderApiService.url}";
}
```

```

final response = await Dio().delete(
  path,
  queryParameters: {
    "user_reminder_uuid": userReminderUuid,
    "delete_type": deleteType,
    "user_uuid": userUuid,
    "user_type": userType,
  },
  options: Options(
    headers: {
      "x-app": "reminder",
      "x-menu": "reminder"
    }
  )
);

```

Gambar 3.21 *Request* hapus pada fungsi di Dio

3.3 Hasil Pelaksanaan Magang

3.3.1 Hasil Survey Pengguna

Hasil yang didapatkan oleh pemegang selama tiga minggu menjalani tugas untuk melakukan survey fitur aplikasi kepada calon pengguna aplikasi *mobile*, yaitu mahasiswa Universitas Islam Indonesia, ditemukan berbagai data dari banyak mahasiswa di berbagai fakultas dan jurusan. Data-data yang berhasil didapatkan tersebut merupakan hasil dari wawancara yang dilakukan menggunakan metode Hook Model. Karena seluruh pemegang di BSI melakukan survey secara berpencar ke seluruh fakultas yang ada di UII, data-data tersebut perlu untuk dikumpulkan dan dipilah terlebih dahulu untuk bisa lanjut ke tahap selanjutnya.

Data yang didapatkan berisi berbagai macam keluhan kesah dari mahasiswa, apa saja yang telah mereka lakukan dalam menangani keluhan kesah tersebut, hasil yang mereka selama ini dapatkan, saran untuk masalah yang dialami, serta fitur apa saja yang mereka inginkan jika ada sebuah aplikasi *mobile* yang diharapkan bisa membantu mereka dalam menjalani perkuliahan. Contoh keluhan kesah yang mereka berikan ketika diwawancarai seperti banyak mahasiswa yang ketiduran di tempat tinggal mereka sehingga ketinggalan sesi perkuliahan sesuai dengan jadwal yang telah ditentukan, lalu ada juga yang kebingungan dalam mencari informasi tentang perkuliahan terutama bagi mahasiswa baru karena informasi masih tersebar di banyak tempat seperti di Instagram UII, *website* UII, ataupun dari kabar mulut ke mulut. Dari keluhan-keluhan yang diutarakan oleh mahasiswa didapatkan juga tujuan yang mereka inginkan dalam menangani keluhan tersebut, seperti ingin mendapatkan sebuah pengingat untuk jadwal perkuliahan mereka, ingin mudah dalam mencari informasi terkait UII, dan lain-lain.

Dari berbagai data yang didapatkan tersebut dibuat sebuah *user persona*, yaitu sebagai karakter fiksi yang merepresentasikan calon pengguna dari aplikasi *mobile* ini. Contoh dari

user persona yang dibuat oleh pemegang bisa dilihat pada Gambar 3.22 *User persona*. *User persona* bertujuan dalam memahami lebih dalam mengenai siapa, perilaku, serta kebutuhan apa yang diinginkan oleh pengguna aplikasi nantinya, selain itu *user persona* ini juga dibuat untuk mewakili seluruh kebutuhan pengguna dari beberapa mahasiswa yang telah diwawancarai (Sparklin, 2019). *User persona* akan sangat membantu bagi anggota *UX designer* di tim akselerasi dalam mendesain aplikasi *mobile I'm UII*.



Sultan Niko

Umur: 20
Pendidikan: Mahasiswa Teknik Industri
Domisili: Yogyakarta
Status Menikah: Belum Menikah
Pekerjaan: Mahasiswa

“Berkuliahlah dengan serius dan dampingi dengan kegiatan berorganisasi.”

Goals

- Kemudahan mengakses informasi UKM dan organisasi.
- Sistem presensi yang lebih praktis.
- Membuka informasi website seperti gateway dan lainnya tanpa selalu login

Frustrations

- Informasi tentang UKM dan organisasi masih tercecer.
- Proses akses website kampus terlalu rumit karena harus selalu login.
- Jadwal berubah terlalu mendadak dan tanpa notifikasi.

Niko adalah seorang mahasiswa yang baru saja mengikuti perkuliahan luring. Saat ini ia ingin mengikuti UKM dan membutuhkan informasi tentang UKM dan organisasi yang ada di kampus. Selain itu sebagai mahasiswa aktif, ia juga sering berurusan dengan website-website yang ada di kampus.

Gambar 3.22 *User persona*

3.3.2 Hasil Pengembangan Fitur Reminder, Profil, dan e-KTM

Setelah melalui proses pengembangan fitur Reminder pada aplikasi *I'm UII*, selanjutnya masuk ke dalam proses pengujian atau testing. Testing pada aplikasi *I'm UII* dilakukan dengan menggunakan metode Black Box Testing dan testing ini dilakukan untuk memastikan apakah setiap fungsionalitas fitur Reminder, profil dan juga e-KTM di dalam aplikasi berjalan sesuai dengan skenario yang diinginkan ketika aplikasi digunakan oleh pengguna. Testing dilakukan oleh anggota developer *I'm UII* dari Tim Akselerasi yang berjumlah lima orang dan sudah termasuk tiga orang pemegang.

Fungsionalitas aplikasi digambarkan melalui *usecase* yang berisi skenario untuk setiap kondisi dalam penggunaan fitur Reminder, profil dan e-KTM di aplikasi *I'm UII*. *Usecase* fungsionalitas dapat dilihat pada Tabel 3.5. Pada Tabel tersebut akan menyajikan setiap

usecase yang dilakukan uji coba terhadap ketiga fitur yang telah disebutkan di paragraf sebelumnya.

Tabel 3.5 Tabel hasil pengujian

| Usecase | Skenario | Target Pengujian | Hasil Pengujian |
|--|--|---|------------------------|
| UC.1 Melihat daftar <i>reminder</i> | Membuka halaman <i>reminder</i> | Pengguna masuk dan dapat melihat halaman <i>reminder</i> utama yang akan berisi daftar <i>reminder</i> milik pengguna | Berhasil |
| | Melihat daftar <i>reminder</i> | Pengguna bisa melihat <i>reminder</i> yang ada atau melihat pemberitahuan jika tidak ada <i>reminder</i> | Berhasil |
| UC.2 Menambah <i>reminder</i> baru | Membuka halaman tambah <i>reminder</i> | Pengguna menekan tombol tambah di halaman <i>reminder</i> dan masuk ke dalam halaman tambah <i>reminder</i> | Berhasil |
| | Mengisi kolom untuk menambah <i>reminder</i> | Pengguna bisa mengisi kolom yang tersedia untuk tambah <i>reminder</i> baru seperti judul, waktu <i>reminder</i> , waktu berulang, dan lainnya. | Berhasil |
| | Membuka halaman tambah orang lain ke dalam <i>reminder</i> | Pengguna menekan tombol tambahkan orang dan masuk ke dalam halaman tambah orang | Berhasil |
| | Mencari pengguna lain dan menambahnya | Pengguna dapat mencari pengguna lain | Berhasil |

| | | | |
|--|---|--|----------|
| | ke daftar <i>guest reminder</i> | dan menambahkannya ke dalam <i>guest list</i> untuk ditambahkan ke dalam <i>reminder</i> | |
| | Menyimpan pengguna yang sudah ditambahkan ke dalam <i>reminder</i> | Pengguna menekan tombol simpan untuk menyimpan perubahan pengguna ke dalam <i>reminder</i> atau menekan tombol kembali untuk membatalkan perubahan | Berhasil |
| | Menyimpan <i>reminder</i> yang sudah dibuat | Pengguna menekan tombol simpan lalu menekan tombol iya untuk menyimpan <i>reminder</i> baru | Berhasil |
| | Kembali ke halaman daftar <i>reminder</i> dari halaman tambah <i>reminder</i> | Pengguna menekan tombol batal setelah menekan tombol simpan dan menekan tombol kembali untuk kembali ke halaman awal <i>reminder</i> dan membatalkan tambah <i>reminder</i> baru | Berhasil |
| UC.3 Melihat detail <i>reminder</i> | Membuka halaman detail <i>reminder</i> | Pengguna menekan salah satu <i>item reminder</i> yang ada di halaman <i>reminder</i> utama dan masuk ke dalam halaman detail dari <i>reminder</i> tersebut | Berhasil |

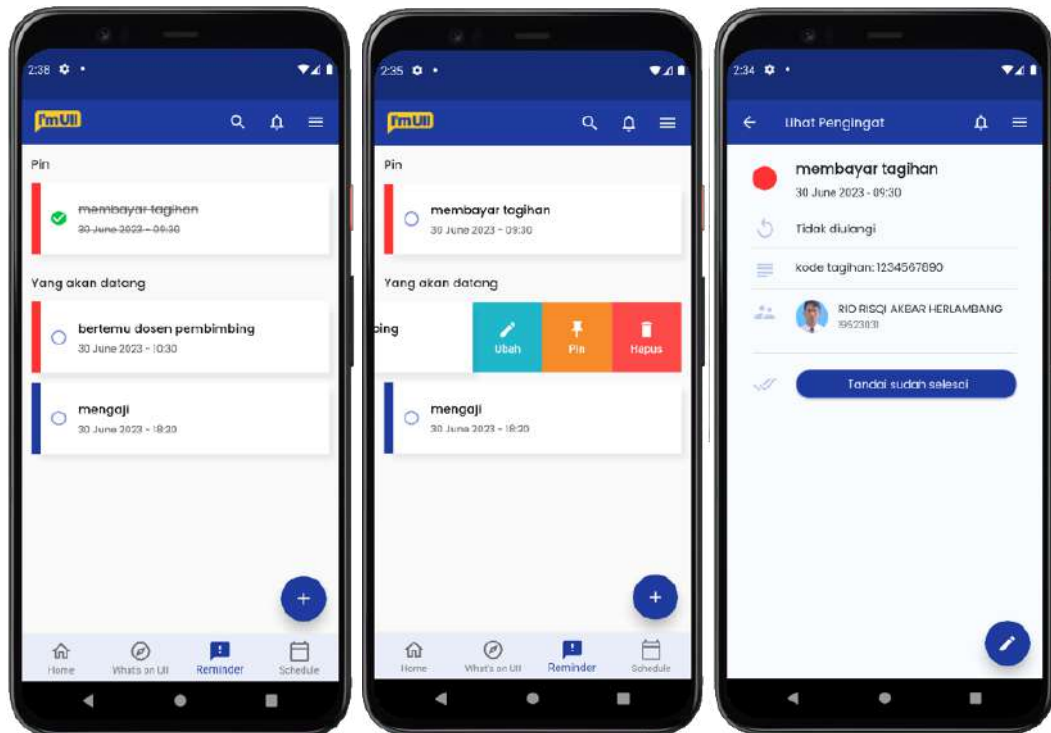
| | | | |
|--|--|--|----------|
| | Kembali ke halaman daftar <i>reminder</i> | Pengguna menekan tombol kembali dan masuk ke halaman daftar <i>reminder</i> | Berhasil |
| UC.4 Melakukan edit <i>reminder</i> | Melakukan aksi <i>swipe</i> pada salah satu <i>item reminder</i> di halaman daftar <i>reminder</i> dan masuk ke halaman edit <i>reminder</i> melalui tombol edit | Pengguna bisa melakukan aksi <i>swipe</i> ke kiri dan melihat menu yang disajikan dan salah satunya adalah edit <i>reminder</i> , lalu pengguna menekan tombol edit <i>reminder</i> dan masuk ke dalam halaman edit <i>reminder</i> tersebut | Berhasil |
| | Menekan tombol edit <i>reminder</i> pada halaman detail <i>reminder</i> dan masuk ke halaman edit <i>reminder</i> | Pengguna dapat menekan tombol edit pada halaman detail <i>reminder</i> dan masuk ke dalam halaman edit <i>reminder</i> tersebut | Berhasil |
| | Mengubah kolom dari <i>reminder</i> yang dipilih | Pengguna bisa mengubah kolom-kolom dari <i>reminder</i> yang dipilih seperti mengubah judul, waktu, dan lainnya | Berhasil |
| | Mengubah atau menambahkan pengguna lain ke dalam <i>reminder</i> | Pengguna dapat menekan tombol tambah orang untuk masuk ke halaman tambah orang dan mengubah pengguna lain yang terdapat pada <i>reminder</i> tersebut | Berhasil |

| | | | |
|--|---|---|----------|
| | | seperti menambahkan atau menghapus pengguna lain, lalu menyimpannya | |
| | Menyimpan perubahan <i>reminder</i> | Pengguna menekan tombol simpan dan memilih apakah menyimpan perubahan untuk seluruh <i>reminder</i> yang sama jika <i>reminder</i> tersebut berulang atau hanya <i>reminder</i> yang sedang diedit saja | Berhasil |
| | Kembali ke halaman sebelumnya dan batal merubah <i>reminder</i> | Pengguna menekan tombol batal setelah menekan tombol simpan dan menekan tombol kembali untuk kembali ke halaman sebelumnya, halaman detail jika masuk dengan menekan tombol edit di halaman detail <i>reminder</i> dan halaman daftar <i>reminder</i> jika menekan tombol edit dari aksi <i>swipe</i> | Berhasil |
| UC.5 Merubah status <i>reminder</i> apakah sudah selesai atau belum | Mengubah status <i>reminder</i> apakah sudah selesai atau belum | Pengguna masuk ke dalam halaman detail <i>reminder</i> dan menekan tombol tandai sudah selesai atau tandai belum selesai | Berhasil |

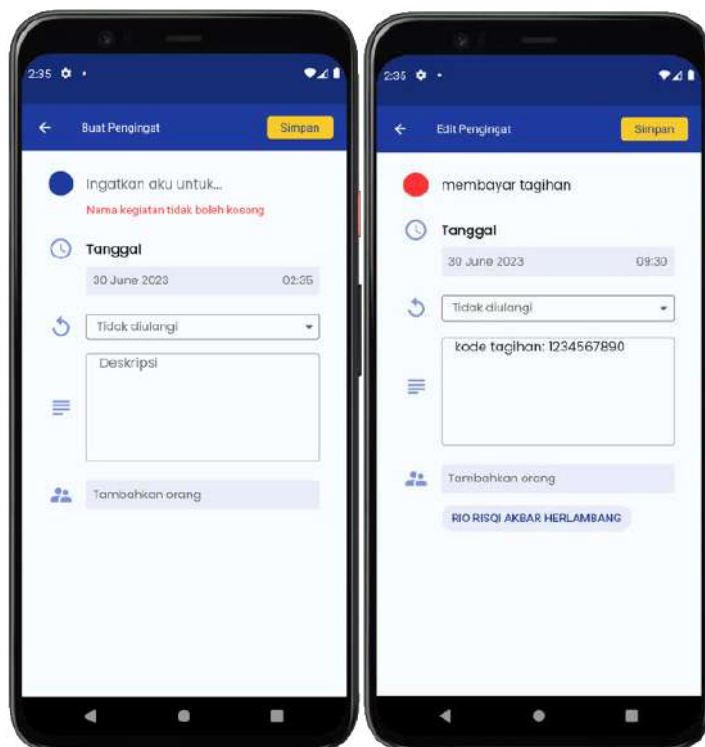
| | | | |
|---|--|---|-----------------|
| <p>UC.6 Melakukan hapus <i>reminder</i></p> | <p>Melakukan aksi <i>swipe</i> pada salah satu <i>item reminder</i> di halaman daftar <i>reminder</i> dan melihat dialog konfirmasi hapus melalui tombol hapus</p> | <p>Pengguna dapat melakukan aksi <i>swipe</i> dan melihat beberapa menu dan salah satunya adalah hapus <i>reminder</i>, lalu pengguna dapat menekan tombol hapus dan melihat dialog konfirmasi untuk memilih apakah menghapus satu <i>reminder</i> saja atau seluruh <i>reminder</i> jika <i>reminder</i> tersebut berulang</p> | <p>Berhasil</p> |
| <p>UC.7 Melakukan Pin <i>reminder</i></p> | <p>Melakukan aksi <i>swipe</i> pada salah satu <i>item reminder</i> di halaman daftar <i>reminder</i> dan menekan tombol pin</p> | <p>Pengguna dapat melakukan aksi <i>swipe</i> dan melihat beberapa menu dan salah satunya adalah pin <i>reminder</i>, lalu pengguna dapat menekan tombol pin dan <i>reminder</i> yang dipilih menjadi berada pada daftar paling atas di kolom khusus pin</p> | <p>Berhasil</p> |
| <p>UC.8 Melihat profil pengguna</p> | <p>Membuka <i>sidebar</i></p> | <p>Pengguna dapat membuka <i>sidebar</i> setelah menekan tombol burger</p> | <p>Berhasil</p> |
| | <p>Membuka halaman profil</p> | <p>Pengguna menekan tombol <i>profile</i> pada <i>sidebar</i> dan masuk ke halaman profil utama</p> | <p>Berhasil</p> |

| | | | |
|--|--|---|----------|
| | Membuka halaman personal data | Pengguna menekan tombol personal data dan masuk ke dalam halaman personal data | Berhasil |
| | Melihat data pengguna | Pengguna bisa melihat data mereka seperti foto profil, nama, nim, fakultas, tanggal lahir, dan lainnya di halaman personal data | Berhasil |
| | Membuka halaman e-KTM | Pengguna menekan tombol e-KTM di <i>sidebar</i> dan masuk ke halaman e-KTM | Berhasil |
| | Melihat e-KTM pengguna | Pengguna bisa melihat e-KTM mereka lengkap dengan data serta <i>barcode</i> -nya | Berhasil |
| | Melihat e-KTM secara <i>fullscreen</i> | Pengguna dapat menekan tombol perbesar untuk melihat e-KTM secara <i>fullscreen</i> | Berhasil |

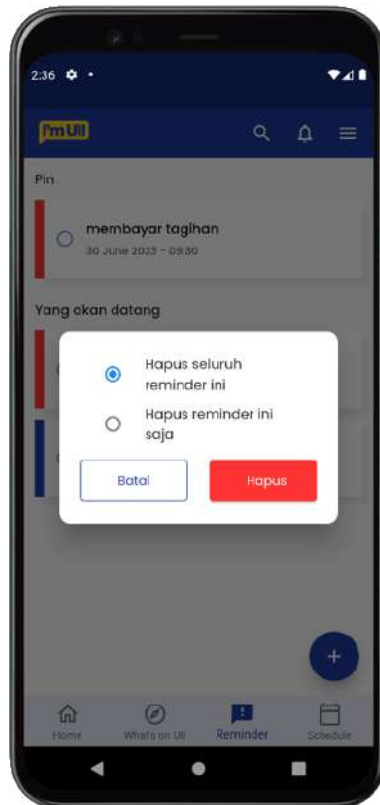
Selanjutnya pada Gambar 3.23 Halaman lihat *reminder* dan detail *reminder* hingga Gambar 3.28 Halaman e-KTM pengguna merupakan hasil tangkapan layar aplikasi I'm UII pada fitur Reminder, halaman profil, dan e-KTM yang sudah diintegrasikan dengan basis data melalui API, hasil tangkapan layar ini diambil menggunakan *emulator* yang telah disediakan oleh IDE Android Studio dan sudah melalui uji coba yang telah dijelaskan pada Tabel 3.5 Tabel hasil pengujian.



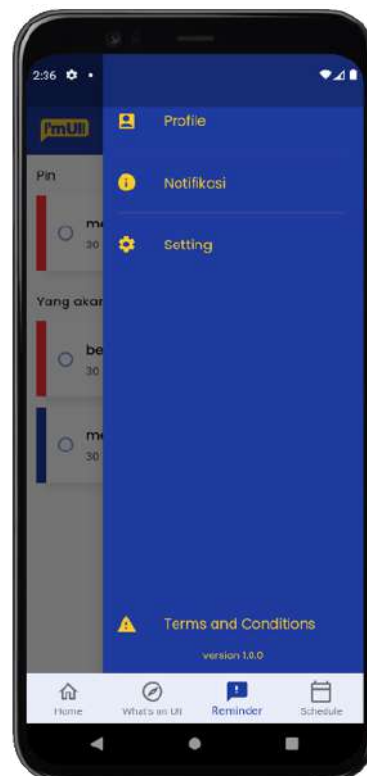
Gambar 3.23 Halaman lihat *reminder* dan detail *reminder*



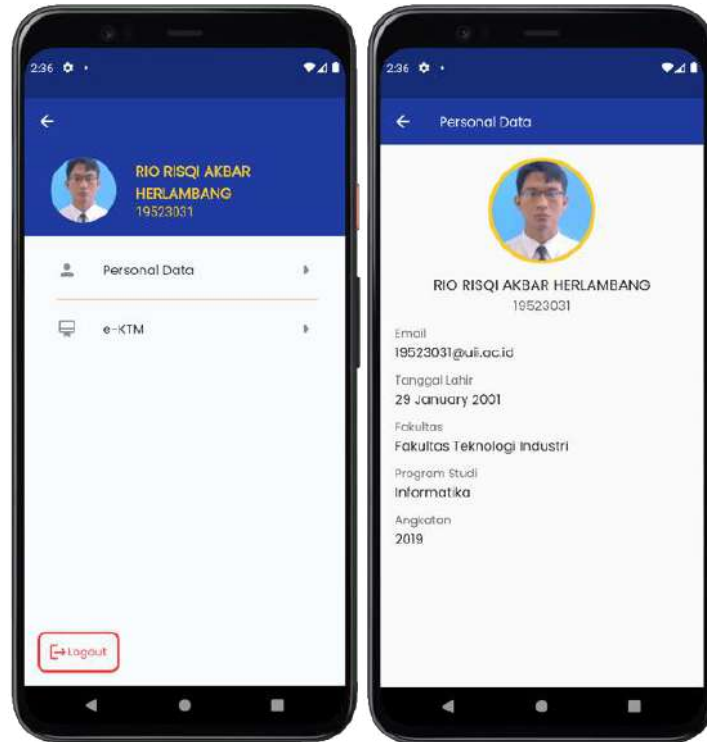
Gambar 3.24 Halaman buat dan ubah *reminder*



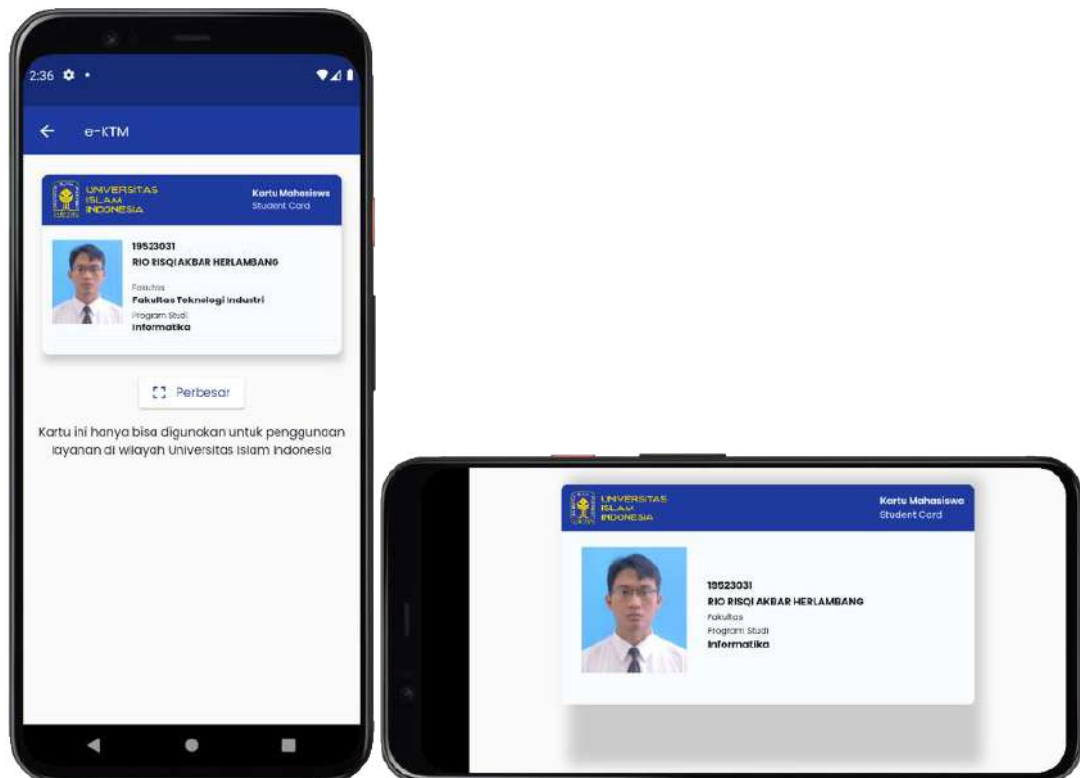
Gambar 3.25 Menu dialog hapus *reminder*



Gambar 3.26 *Sidebar*



Gambar 3.27 Halaman profil dan personal data pengguna



Gambar 3.28 Halaman e-KTM pengguna

3.3.3 Hasil Pengembangan *Backend*

Bagian *backend* pada aplikasi I'm UII digunakan untuk mendukung fitur-fitur yang ada pada aplikasi tersebut agar bisa berjalan. Pada bagian ini pemegang membuat *endpoint* yang sesuai dengan fitur yang dikerjakan masing-masing dan testing yang dilakukan pun sama, yaitu menggunakan Black Box Testing. Pada pengujian ini dilakukan untuk mengetahui apakah *endpoint* yang telah dibuat berfungsi dengan semestinya jika pengguna berinteraksi dengan aplikasi I'm UII, misal jika pengguna ingin melihat data maka *endpoint* berupa GET yang bersangkutan akan berjalan, dan begitu pun dengan *endpoint* lainnya. Selengkapnya hasil pengujian *endpoint* API ini dapat dilihat pada Tabel 3.6.

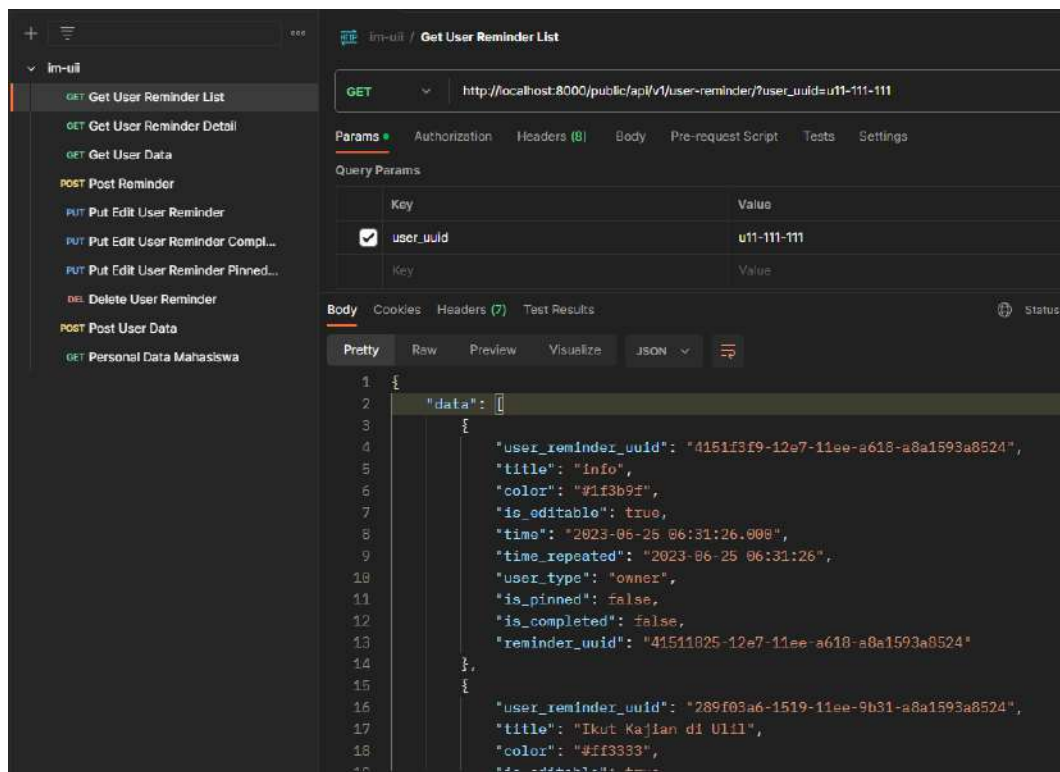
Tabel 3.6 Tabel hasil pengujian *endpoint*

| Usecase | Skenario | Target Pengujian | Hasil Pengujian |
|---|---|--|-----------------|
| UC.1 <i>Endpoint GET user reminder</i> | Mendapatkan respon daftar <i>reminder</i> milik pengguna | Memasukkan parameter yang dibutuhkan oleh <i>endpoint</i> , yaitu UUID milik pengguna lalu mengirimkannya dan mendapatkan respon berupa daftar <i>reminder</i> milik pengguna | Berhasil |
| UC.2 <i>Endpoint GET detail reminder</i> | Mendapatkan respon detail dari <i>reminder</i> yang dipilih | Memasukkan parameter yang dibutuhkan oleh <i>endpoint</i> , yaitu UUID dari <i>reminder</i> lalu mengirimkannya dan mendapatkan respon berupa detail dari <i>reminder</i> dengan UUID tersebut | Berhasil |
| UC.3 <i>Endpoint POST reminder</i> | Berhasil mengirim data <i>reminder</i> baru | Mengisi <i>body</i> dalam format JSON yang berisi data <i>reminder</i> yang ingin dibuat lalu mengirimkannya dan | Berhasil |

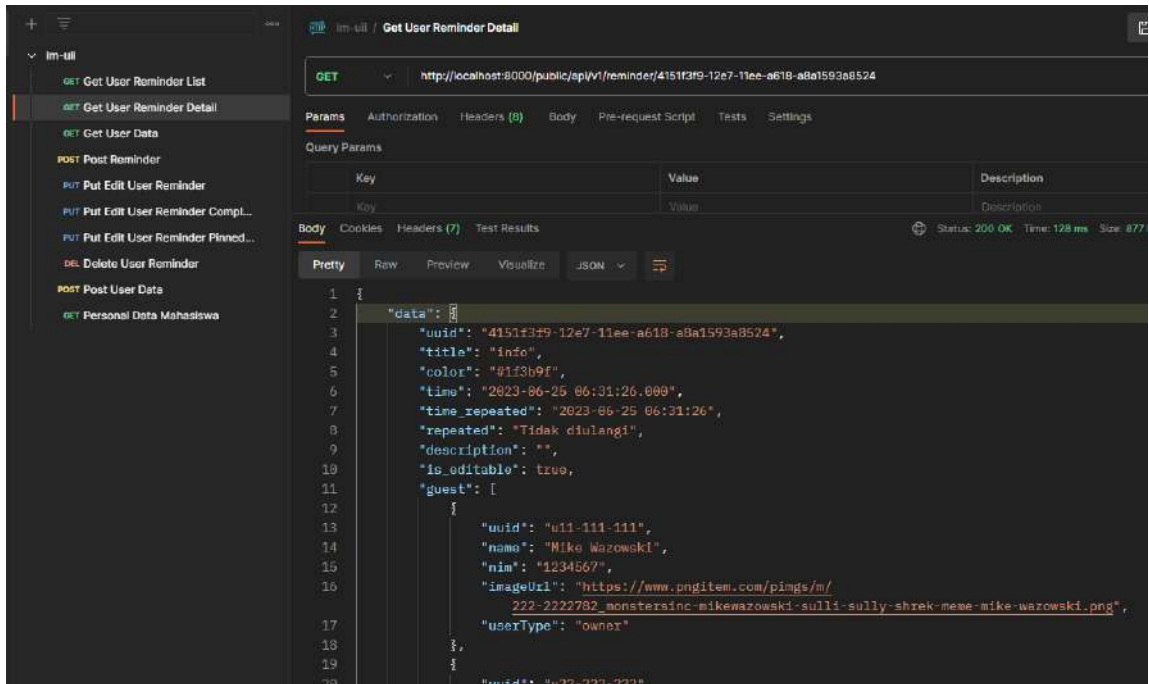
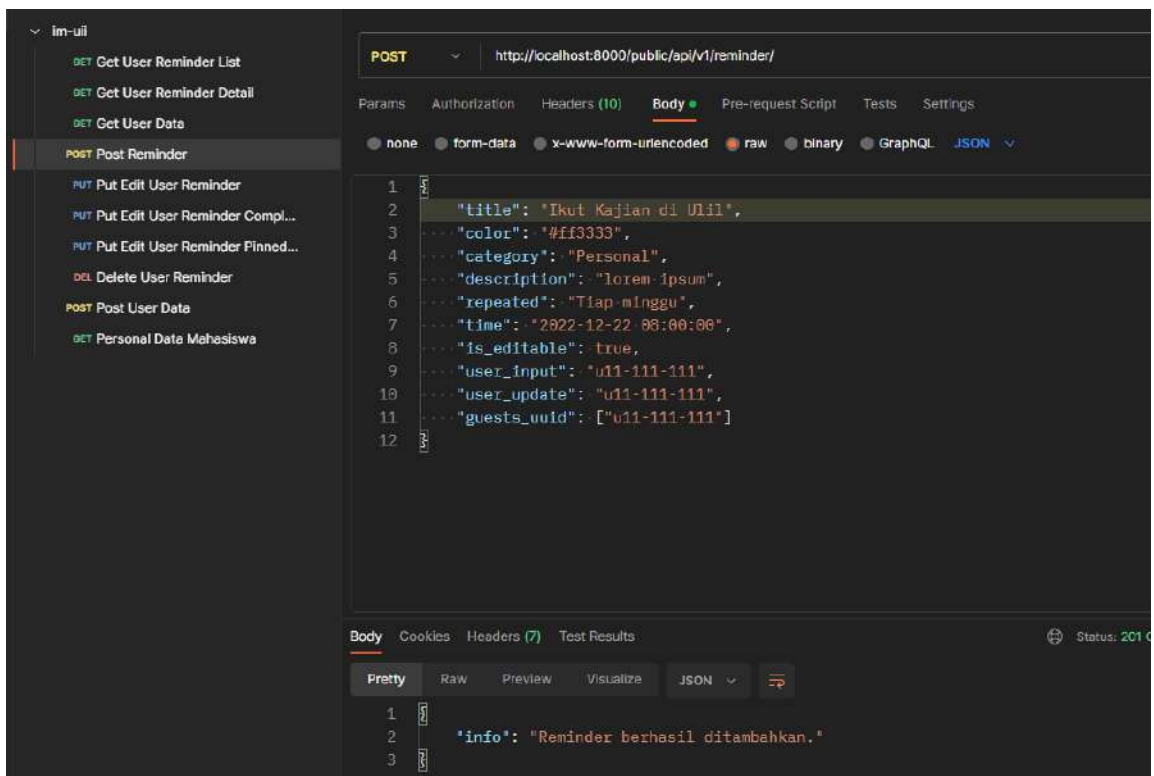
| | | | |
|--|---|---|--|
| | | mendapatkan respon <i>reminder</i> berhasil ditambahkan | |
| <i>UC.4</i> <i>Endpoint PUT</i> <i>reminder</i> | Berhasil mengubah isi data <i>reminder</i> yang dipilih | Mengubah <i>body</i> dalam format JSON yang berisi data <i>reminder</i> yang ingin diubah lalu mengirimkannya dan mendapatkan respon <i>reminder</i> berhasil diubah | Berhasil terkirim dan mendapat responnya namun dengan catatan terdapat bug pada bagian <i>guest list reminder</i> yang terkadang tidak berubah daftarnya |
| <i>UC.5</i> <i>Endpoint PUT</i> <i>complete reminder</i> | Berhasil mengubah status selesai <i>reminder</i> | Mengubah <i>body</i> dalam format JSON yang berisi <i>field</i> berupa status selesai <i>reminder</i> menjadi <i>true</i> atau <i>false</i> lalu mengirimkannya dan mendapatkan respon status selesai berhasil diperbarui | Berhasil |
| <i>UC.6</i> <i>Endpoint PUT pin</i> <i>reminder</i> | Berhasil mengubah status pin <i>reminder</i> | Mengubah <i>body</i> dalam format JSON yang berisi <i>field</i> berupa status pin <i>reminder</i> menjadi <i>true</i> atau <i>false</i> lalu mengirimkannya dan mendapatkan respon status pin berhasil diperbarui | Berhasil |
| <i>UC.7</i> <i>Endpoint DELETE</i> <i>user reminder</i> | Berhasil menghapus <i>reminder</i> | Memasukkan parameter yang dibutuhkan oleh <i>endpoint</i> untuk menghapus sebuah <i>reminder</i> dan | Berhasil |

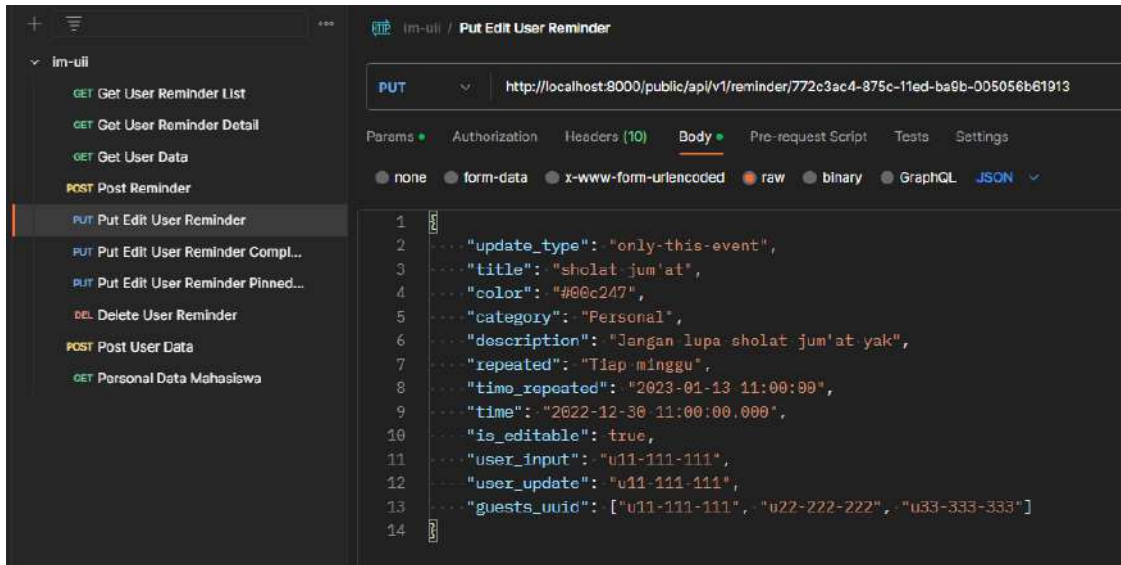
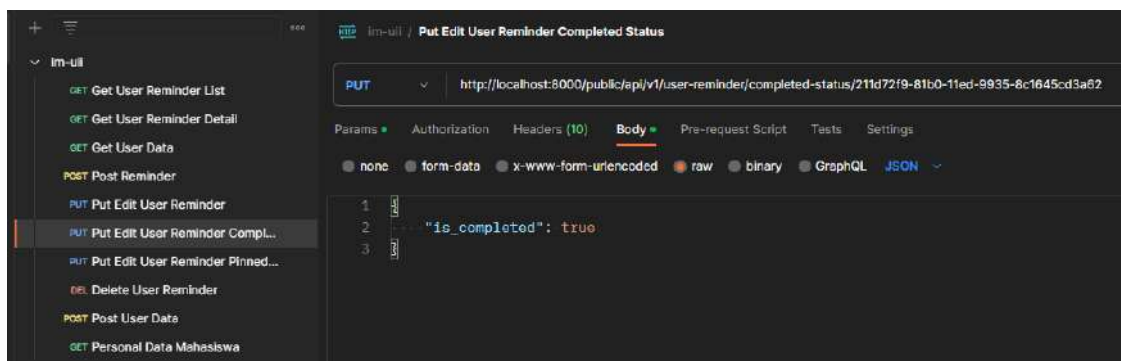
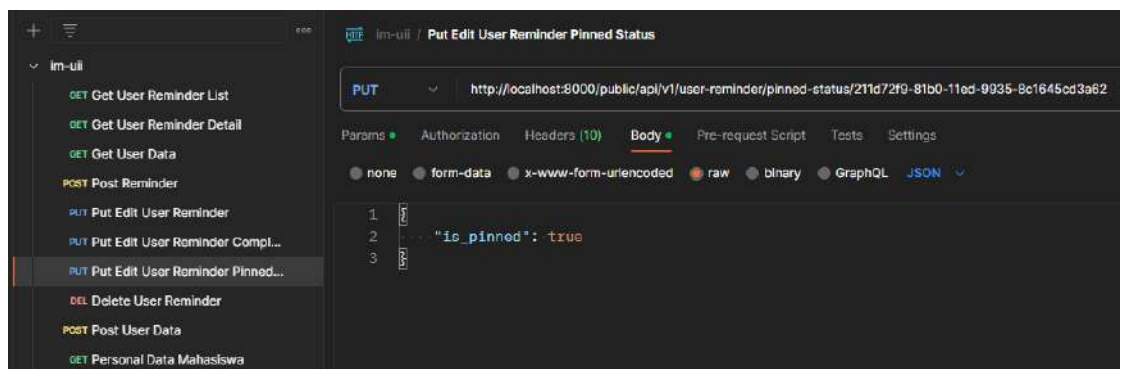
| | | | |
|---------------------------------------|---|---|----------|
| | | mendapatkan respon <i>reminder</i> berhasil dihapus | |
| UC.8 <i>Endpoint GET user data</i> | Mendapatkan respon berupa data pengguna | Memasukkan parameter yang dibutuhkan oleh <i>endpoint</i> , yaitu UUID milik pengguna lalu mengirimkannya dan mendapatkan respon berupa data milik pengguna | Berhasil |

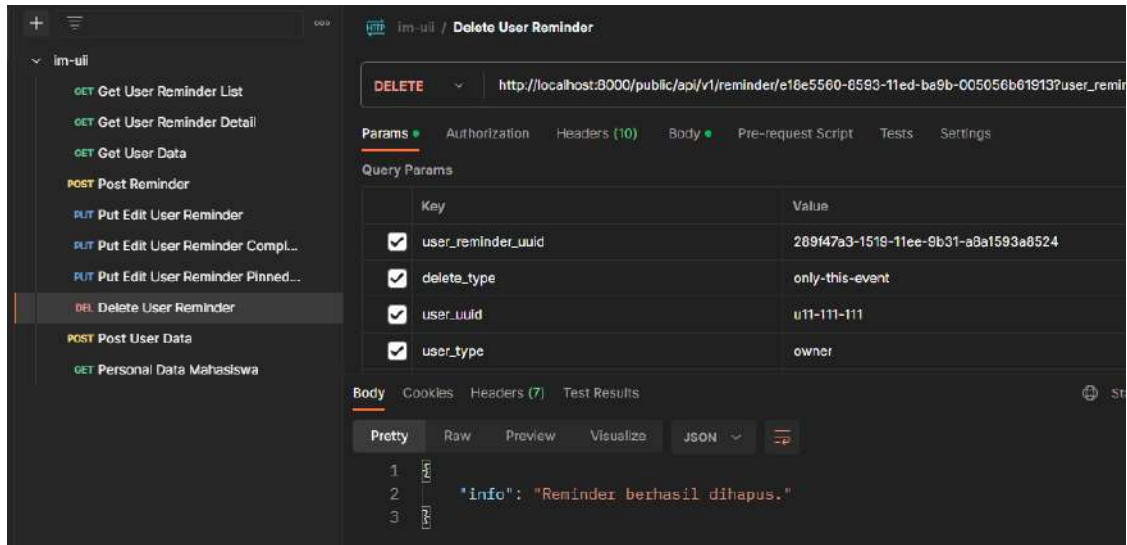
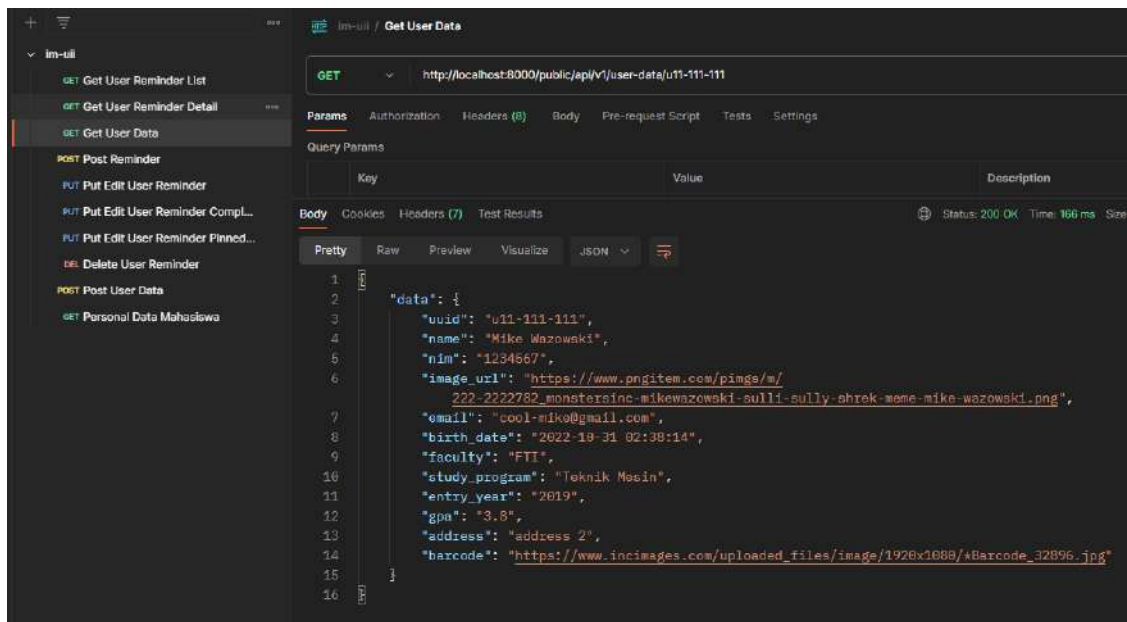
Gambar 3.29 GET *reminder list* hingga Gambar 3.36 GET *user data* menunjukkan tangkapan layar dari API yang telah dibuat. Untuk hasil gambar tangkapan layar tersebut akan ditampilkan berupa hasil uji coba *endpoint* dari aplikasi Postman, yaitu sebuah *software* yang digunakan untuk melakukan uji coba API dan juga tangkapan layar dari hasil akhir basis data yang telah dibuat.



Gambar 3.29 GET *reminder list*

Gambar 3.30 GET *reminder* detailGambar 3.31 POST *reminder*

Gambar 3.32 PUT edit *user reminder*Gambar 3.33 PUT edit *reminder completed status*Gambar 3.34 PUT edit *reminder pinned status*

Gambar 3.35 DELETE *user reminder*Gambar 3.36 GET *user data*

BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Berdasarkan teori-teori yang telah dikaji sebelumnya, ditemukan bahwa terdapat perbedaan antara teori yang telah dikemukakan dengan fakta sebenarnya yang terjadi di lapangan. Menurut (Holla & Katti, 2012) SDK Android menyediakan kumpulan antarmuka pemrograman aplikasi (API) yang luas, modern, dan tangguh, dan layanan inti dari *device* Android dapat diakses langsung oleh semua aplikasi untuk berbagi data jika diberikan izin yang sesuai. Nyatanya tidak semua tipe *smartphone* Android memberikan akses izin dengan bebas, beberapa merek tertentu membatasi izin yang diberikan sehingga developer perlu memikirkan cara ekstra agar bisa mendapatkan izin tersebut.

Perbedaan ini sangat dirasakan saat proses pengembangan aplikasi I'm UII, aplikasi ini dikembangkan agar bisa berjalan secara *background* dan mendapatkan notifikasi baru walaupun aplikasi sedang tidak dijalankan, agar bisa berjalan seperti ini dibutuhkan izin khusus secara *background* pada sistem Android di setiap *smartphone* yang terinstal aplikasi I'm UII. Namun ditemukan kekurangan pada aplikasi ini saat melakukan uji coba, aplikasi tidak berjalan secara *background* pada beberapa tipe *smartphone*, padahal secara struktur kode sudah diberikan perintah kode untuk meminta izin ke Android untuk menggunakan *foreground* dan *background service* akan tetapi izin tersebut tidak bisa berjalan. Setelah ditelusuri, ditemukan bahwa pada sebagian tipe *smartphone* perlu memberikan izin untuk kontrol optimisasi baterai yang dilakukan Android terhadap aplikasi-aplikasi yang ada di dalamnya, jika optimisasi baterai tersebut dimatikan untuk aplikasi yang diinginkan *foreground* dan *background service* bisa berjalan, akan tetapi ada juga yang memerlukan izin untuk otomatis memulai aplikasi atau *auto-start* saat *smartphone* dinyalakan sehingga aplikasi pun bisa langsung berjalan tanpa perlu membuka aplikasi tersebut. Akan tetapi, terkadang sulit untuk menemukan cara mengakses izin dari auto-start dan optimisasi baterai pada tiap *smartphone* sehingga pengguna perlu mengaturnya sendiri di pengaturan *smartphone* masing-masing dan dapat mengurangi pengalaman pengguna dalam menggunakan aplikasi.

Untuk proses pengembangan *backend* berjalan dengan sesuai seperti teori yang sudah dijelaskan, pembuatan API berjalan lancar dan setiap *endpoint* berfungsi dengan baik untuk melakukan proses CRUD di dalam basis data. Pengembangan aplikasi *mobile* I'm UII juga berjalan lancar menggunakan *framework* Flutter, dengan fitur *hot reload* yang sangat

membantu dalam proses pengembangan karena jika dibandingkan dengan *native* Android yang menggunakan bahasa pemrograman seperti Kotlin, fitur *hot reload* hanya perlu untuk *refresh widget* yang sedang berjalan dan tidak perlu untuk *build* ulang seluruh aplikasi yang sedang dijalankan seperti ketika menggunakan Kotlin.

4.2 Pembelajaran Magang

Banyak manfaat yang didapatkan selama mengikuti pelaksanaan magang di BSI UII selama delapan bulan. Manfaat tersebut yakni mengetahui tentang pengembangan aplikasi *mobile* dengan menggunakan *framework* Flutter, pengembangan *backend* dengan membuat API menggunakan *framework* Lumen dan menghubungkannya dengan Flutter, dan cara *version control* sebuah proyek menggunakan Gitlab.

Manfaat pertama adalah mengetahui bagaimana cara pengembangan sebuah proyek aplikasi *mobile* dengan menggunakan *framework* Flutter. Jika dibandingkan dengan bahasa *native* Android seperti Kotlin, Flutter memiliki keunggulan di beberapa hal salah satunya adalah *hot reload* yang memungkinkan developer untuk tidak perlu *build* ulang aplikasi sehingga membutuhkan waktu lama selama proses pengembangan. Kemudian sistem *single codebase*-nya menjadi andalan Flutter agar bisa mengembangkan aplikasi untuk banyak *device* hanya dengan satu bahasa pemrograman, yaitu Dart.

Manfaat kedua, yaitu mengetahui bagaimana cara membuat API dengan menggunakan *framework* Lumen. Dari pembuatan API ini, pemagang bisa mempelajari mengenai kegunaannya sebagai jembatan penghubung antara satu aplikasi dengan aplikasi lainnya. Selain dari membuat API, pemagang jadi bisa mengetahui bagaimana implementasinya untuk mengonsumsi API tersebut di aplikasi lain sehingga bisa berkomunikasi dengan lancar terutama dalam Flutter.

Manfaat ketiga adalah mengetahui bagaimana cara *version control* sebuah proyek dengan menggunakan Gitlab. Melalui Gitlab jadi mengetahui bagaimana caranya berkolaborasi dengan tim untuk mengerjakan sebuah proyek, mulai dari membuat *branch* baru, menyelesaikan *conflict* saat melakukan penggabungan *branch*, dan aksi seperti *push*, *pull*, *commit*, dan *reverse*.

4.3 Kendala, Hambatan, dan Tantangan

Selama proses pelaksanaan magang adapun kendala yang dialami, yaitu saat proses pengembangan aplikasi I'm UII ditemukan bahwa *foreground* dan *background service* ada yang tidak berjalan pada beberapa tipe *smartphone* seperti yang telah dijelaskan pada Subbab

4.1 sehingga notifikasi tidak akan berjalan dan aplikasi tidak bisa berjalan di latar belakang atau secara *background*. Hal ini sempat menjadi kendala karena sulit untuk menemukan solusinya sebab setiap tipe *smartphone* juga berbeda cara perizinan yang diberikan untuk mengakses sistem Android-nya, seperti ada yang harus mematikan optimisasi baterainya, lalu ada juga yang perlu menyalakan *auto-start* dan setiap izin tersebut tidak bisa dilakukan lewat aplikasi pihak ketiga dan perlu untuk mengaturnya langsung melalui pengaturan pada *smartphone*. Hal ini tentunya bisa mengurangi pengalaman pengguna ketika menggunakan aplikasi sehingga solusi sementara yang ditemukan adalah untuk mengarahkan pengguna jika ditemukan bahwa *foreground* dan *background service* tidak berjalan maka akan diarahkan ke dalam pengaturan *smartphone* pengguna untuk melakukan perizinan secara manual dan tidak melalui aplikasi I'm UII.

Selanjutnya untuk hambatan yang dialami adalah kurangnya pengetahuan dan pengalaman pemegang mengenai *backend*. Karena pemegang sangat minim pengetahuan mengenai *backend* sehingga pemegang perlu untuk mempelajarinya terlebih dahulu dan membutuhkan waktu tidak sebentar. Selain itu perlu waktu juga untuk mengimplementasikan hasil yang telah dipelajari ke dalam pembuatan API menggunakan Lumen. Walaupun sudah dibantu dengan dokumentasi Lumen yang lengkap terkadang masih ada banyak *bug* yang ditemukan selama proses pembuatan API. Solusi yang dilakukan untuk hambatan ini adalah dengan belajar secara cepat dan efektif, walaupun di luar dari waktu magang di BSI pemegang melanjutkan pembelajarannya mengenai Lumen dengan harapan agar secepatnya untuk bisa terjun dan membuat API.

Terakhir untuk tantangan yang dihadapi adalah mengenai komunikasi antar anggota Tim Akselerasi, karena tim ini terdiri dari anggota yang merupakan karyawan BSI dan memiliki jam kerja tetap, yaitu dari jam 08.00-16.00 WIB sedangkan waktu kerja untuk Tim Akselerasi adalah di luar dari jam kerja tetap BSI. Dari perbedaan waktu tersebut, pemegang mengalami kesulitan jika ingin berkomunikasi dengan anggota lainnya karena anggota lain sibuk dengan pekerjaan yang lainnya dan hanya bisa berkomunikasi di waktu istirahat saja dan di luar jam kerja BSI, ditambah jam kerja pemegang yang hanya dari jam 08.00-12.00 WIB. Solusi dari tantangan ini adalah hanya bisa mengandalkan media komunikasi seperti WhatsApp untuk berkomunikasi dengan rekan tim.

Adapun tantangan lainnya adalah mengenai tidak adanya penerapan *android architecture* seperti *Model View ViewModel* (MVVM) pada proyek aplikasi I'm UII. Penerapan *android architecture* pada sebuah aplikasi *mobile* merupakan hal yang penting untuk dilakukan karena

kode yang dibuat akan lebih kokoh, mudah untuk diuji, serta *maintanable* untuk kedepannya (Google, 2023). Selain itu hal ini juga penting untuk diterapkan pada aplikasi berbasis *mobile* karena sifatnya yang dinamis dan potensi untuk menghadirkan fitur-fitur baru sesuai kebutuhan penggunanya ke dalam aplikasi, jika kode yang dibuat sulit untuk dipahami dan tidak terstruktur akan menyusahkan developer jika ingin membuat dan mengintegrasikan fitur baru tersebut ke dalam kodenya. Karena proyek ini juga baru pertama kali diinisiasi oleh BSI UII dan baru mulai dikerjakan saat periode magang BSI UII dibuka menyebabkan tidak adanya penerapan *android architecture*. Solusi yang dilakukan mengenai tantangan ini adalah menuliskan kodenya secara langsung saat pengerjaan tanpa ada petunjuk mengenai struktur penulisan kode. Cara ini berhasil dilakukan namun sangat beresiko nanti untuk kedepannya karena kode yang dibuat kemungkinan akan sulit dipahami apalagi pengerjaan proyek ini terdiri dari banyak orang.

BAB V PENUTUP

5.1 Kesimpulan

Pengembangan aplikasi *mobile* I'm UII untuk fitur Reminder dan halaman profil serta e-KTM telah berhasil dilakukan dengan menggunakan *framework* Flutter dan juga bagian *backend*-nya dengan menggunakan *framework* Lumen. Pada fitur Reminder terdapat halaman utama untuk melihat daftar *reminder* milik pengguna, menambah *reminder* baru, mengubah *reminder* yang sudah ada, menambah atau menghapus pengguna lain pada *reminder* baik saat menambah ataupun mengubah *reminder*, pin *reminder* yang berfungsi membuat *reminder* menjadi ada pada daftar teratas bagian pin di halaman utama, mengubah status *reminder* apakah sudah selesai atau belum selesai, dan menghapus *reminder*. Lalu pada halaman profil pengguna yang terdapat menu untuk mengakses halaman personal data pengguna dan e-KTM pengguna.

Untuk bagian *backend* menggunakan *framework* Lumen dalam pembuatan API telah berhasil dilakukan dan diimplementasikan ke dalam aplikasi *mobile* I'm UII untuk digunakan sesuai yang diharapkan, namun dengan satu catatan pada bagian *endpoint* untuk mengubah *reminder* pengguna. Pada *endpoint* ubah *reminder* terdapat *bug* yang terkadang muncul dan menyebabkan daftar pengguna lain yang ada pada *reminder* tersebut tidak berubah walaupun pengguna lain tersebut sudah dihapus, namun berfungsi normal jika menambah pengguna baru ke dalam *reminder*. *Endpoint* lainnya seperti untuk melihat daftar *reminder* milik pengguna, detail *reminder* dan data pengguna berfungsi dengan baik, *endpoint* untuk mengubah status selesai dan pin *reminder* berfungsi dengan baik, lalu *endpoint* untuk menambah dan menghapus *reminder* juga berfungsi dengan baik.

5.2 Saran

Aplikasi *mobile* I'm UII masih memerlukan pengembangan untuk kedepannya. Salah satunya adalah untuk masalah perizinan di Android yang telah dijelaskan pada bab sebelumnya, karena perizinan ini juga akan berpengaruh jika aplikasi ingin di-*deploy* ke *app store* seperti Google Play Store yang memiliki peraturannya sendiri untuk masalah izin yang digunakan dan data yang diambil pada aplikasi. Sebagai saran diperlukan riset lebih mendalam lagi mengenai perizinan untuk tiap *smartphone* Android yang berbeda dan tahun rilisnya yang berbeda. Saran selanjutnya mengenai pengembangan aplikasi menggunakan *android architecture pattern*

seperti *Model View Controller* (MVC) atau *Model View ViewModel* (MVVM), karena rekomendasi dari Google saat ini adalah menggunakan *android architecture* berupa MVVM (Google, 2023). Dengan menerapkan *android architecture* harapannya kode untuk aplikasi I'm UII menjadi lebih *maintanable* dan lebih mudah dalam pengembangannya.

DAFTAR PUSTAKA

- Adenowo, A. A. A., & Adenowo, A. B. (2013). Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach. *International Journal of Scientific and Engineering Research*, 4(7).
- Adservio. (2022, July 20). *What is Flutter and Its Advantages?*
<https://www.adservio.fr/post/what-is-flutter-and-what-are-its-advantages>
- Dagne, L. (2019). *Flutter for cross-platform App and SDK development*.
- Dicoding Indonesia. (n.d.-a). *Belajar Fundamental Aplikasi Android*. Retrieved June 30, 2023, from <https://www.dicoding.com/academies/14/tutorials/6509>
- Dicoding Indonesia. (n.d.-b). *UX Design*. Retrieved June 16, 2023, from <https://www.dicoding.com/academies/335/tutorials/18739>
- Flower, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley Professional.
- Google. (2023, March 28). *Guide to app architecture*.
<https://developer.android.com/topic/architecture/>
- Hakim, A. R., Harefa, K., & Widodo, B. (2019). *PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS ANDROID MENGGUNAKAN FLUTTER DI POLITEKNIK*.
- Holla, S., & Katti, M. M. (2012). ANDROID BASED MOBILE APPLICATION DEVELOPMENT and its SECURITY. *International Journal of Computer Trends and Technology*, 3(3). <http://www.internationaljournalsrg.org>
- Lukyanichikova, E., Askarbekuly, N., Aslam, H., & Mazzara, M. (2023). A Case Study on Applications of the Hook Model in Software Products. *Software*, 2(2), 292–309.
<https://doi.org/10.3390/software2020014>
- Muktasim, I., & Buyens, J. (2001). *WEB database development*. Elex Media Komputindo.
- Pratama, A. P., & Kamisutara, M. (2021). PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS MOBILE MENGGUNAKAN FLUTTER DI UNIVERSITAS NAROTAMA SURABAYA. In *Jurnal Ilmiah NERO* (Vol. 6, Issue 2).
- Pressman, S. R. (2005). *Software Engineering: A Practitioner's Approach* (6th ed.). McGraw Hill.
- Sparklin. (2019, October 1). *What is a User Persona and how is it relevant to UX?*
<https://medium.muz.li/what-is-a-user-persona-and-how-is-it-relevant-to-ux-56e8c7c12887>

- Tantular Tezar, Grahita Banung, & Tresnadi Chandra. (2017). *Mobile Apps "MAIWAK" as Media to Increase Fish Consumption in Yogyakarta*. https://www.researchgate.net/publication/320297829_Mobile_Apps_MAIWAK_as_Media_to_Increase_Fish_Consumption_in_Yogyakarta
- Thomas, G. (2019, December 12). *What is Flutter and Why You Should Learn it in 2020*. FreeCodeCamp. <https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>
- Widarno, W., & Putra, S. I. (2023). Penyediaan Layanan Web Service Menggunakan Framework Lumen Untuk Berbagi Layanan Data (Studi Kasus: SITU Akademik Universitas Pasundan). *Pasinformatik*, 2(1).
- Wijayanto, A., Rohmadi, A., & Permana, U. (2018). *Membangun Web API dengan Lumen 5.5*.

LAMPIRAN

