

**PENGEMBANGAN FITUR WHAT'S ON UII PADA APLIKASI
MOBILE I'M UII MENGGUNAKAN FRAMEWORK
FLUTTER**



Disusun Oleh:

Nama : Fahrudin Nasikh Az Zuhdu

NIM : 19523005

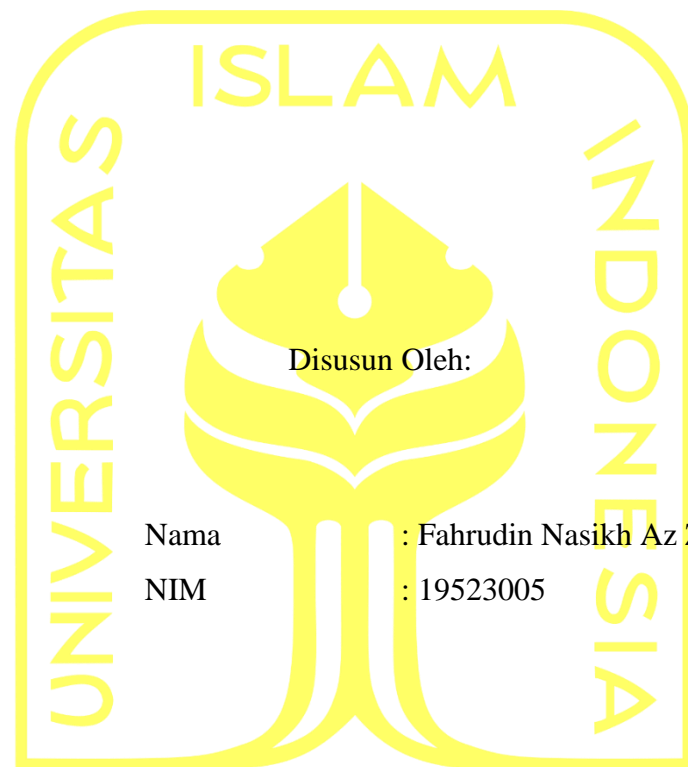
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2023

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN FITUR WHAT'S ON UII PADA APLIKASI
MOBILE I'M UII MENGGUNAKAN FRAMEWORK
FLUTTER**

TUGAS AKHIR



Nama : Fahrudin Nasikh Az Zuhdu

NIM : 19523005

الجامعة الإسلامية
الاندونيسية

Yogyakarta, 10 Juli 2023

Pembimbing,



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

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN FITUR WHAT'S ON UII PADA APLIKASI
MOBILE I'M UII MENGGUNAKAN FRAMEWORK
FLUTTER**

TUGAS AKHIR

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, 20 Juli, 2023

Tim Penguji

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.

Anggota 1

Hari Setiaji, S.Kom., M.Eng.

Anggota 2

Kholid Haryono, S.T., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



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

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Fahrudin Nasikh Az Zuhdu

NIM : 19523005

Tugas akhir dengan judul:

**PENGEMBANGAN FITUR WHAT'S ON UII PADA APLIKASI
MOBILE I'M UII MENGGUNAKAN FRAMEWORK
FLUTTER**

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 10 Juli 2023



(Fahrudin Nasikh Az Zuhdu)

HALAMAN PERSEMBAHAN

Laporan akhir ini dengan tulus dan penuh rasa terima kasih saya persembahkan kepada orangtua tercinta. Terima kasih atas dukungan, doa, dan cinta tanpa syarat yang telah mereka berikan sepanjang perjalanan ini. Kehadiran dan dorongan mereka menjadi sumber inspirasi yang tak tergantikan. Saya berterima kasih atas segala pengorbanan mereka yang telah membantu saya mencapai titik ini.

HALAMAN MOTO

"Tidak ada kesulitan yang bertahan selamanya. Sesungguhnya, bersama kesulitan, ada kemudahan."

-Al-Qur'an (Surat Al-Insyirah: 5-6)

"The only way to do great work is to love what you do."

- Steve Jobs

KATA PENGANTAR

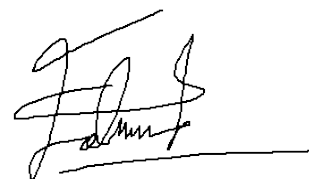
Alhamdulillah puji syukur ke hadirat Allah yang memberikan hidayah dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan akhir yang berjudul “Pengembangan Fitur What’s on UII pada Aplikasi *Mobile I’m UII* Menggunakan Framework Flutter”. Laporan akhir ini disusun sebagai syarat untuk memenuhi tugas akhir pada penjaluran magang, dan untuk memperoleh gelar sarjana pada Program Studi Informatika Universitas Islam Indonesia.

Dalam penyusunan laporan akhir ini, tentu tak lepas dari pengarahan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Orang tua serta keluarga tercinta yang selalu memberikan doa dan dukungan baik berupa materi maupun moral.
2. Bapak DThomas Hatta Fudholi, selaku Ketua Program Studi Informatika Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia, sekaligus dosen pembimbing yang telah banyak memberi arahan dan masukan selama penyusunan laporan akhir ini.
3. Bapak Raden Teduh Dirgahayu selaku Ketua Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia
4. Seluruh dosen di Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, atas arahan dan pengetahuan yang telah diberikan sepanjang masa perkuliahan ini.
5. Seluruh teman-teman penulis yang telah memberikan semangat, dukungan, serta menjadi tempat berkeluh kesah.

Karena kebaikan dari mereka semua, penulis dapat menyelesaikan laporan akhir ini dengan maksimal. Laporan akhir ini memang masih jauh dari kata sempurna, tapi penulis berharap laporan ini dapat bermanfaat bagi siapapun yang membacanya.

Yogyakarta, 10 Juli 2023



(Fahrudin Nasikh Az Zuhdu)

SARI

Dalam era modern ini, penggunaan aplikasi *mobile* telah mengubah cara kita menjalani kehidupan sehari-hari dengan memperkenalkan kecanggihan teknologi ke dalam genggaman tangan kita. Belakangan ini, salah satu tantangan yang dihadapi oleh mahasiswa di Universitas Islam Indonesia (UII) adalah kesulitan mendapatkan berita dan informasi yang diperlukan. Berita dan informasi yang relevan seringkali tersebar secara terpencar di berbagai media, menyebabkan mahasiswa harus mencari informasi secara manual dari sumber yang berbeda.

Untuk mengatasi masalah ini, Badan Sistem Informasi UII sedang mengembangkan sebuah fitur pada aplikasi *mobile* I'm UII yang diberi nama What's on UII. Fitur ini menyediakan akses terpusat ke berbagai informasi penting di UII. Dalam pengembangan aplikasi I'm UII, teknologi yang digunakan adalah Flutter. Flutter dipilih karena memiliki berbagai keunggulan dan kemudahan penggunaan dalam mengembangkan aplikasi lintas platform. Selama melaksanakan magang di BSI UII. Penulis telah terlibat dan ditugaskan untuk mengerjakan fitur What's on UII. Tugas utama penulis adalah merealisasikan aplikasi dari sisi *mobile*. Namun, penulis juga terlibat dalam pengembangan dari sisi *back-end*.

Dalam pengembangannya, metode yang diadopsi adalah metode Waterfall yang diawali dengan menganalisis kebutuhan, kemudian dilanjutkan desain arsitektur dan fisik aplikasi, dan dilanjutkan dengan pemograman aplikasi, hingga melakukan pengujian. Pengembangan fitur What's on UII telah mencapai tahap pengujian. Pada tahap ini, pengujian dilakukan menggunakan metode *black-box* dengan hasil sesuai dengan ekspektasi yang diinginkan. Hal ini menunjukkan bahwa fitur What's on UII telah berhasil dikembangkan sesuai dengan hasil yang diharapkan.

Kata kunci: aplikasi *mobile*, What's on UII, Waterfall.

GLOSARIUM

Aplikasi Mobile	Program yang dirancang khusus untuk digunakan pada perangkat mobile seperti smartphone atau tablet.
Android	Sistem operasi <i>mobile</i> yang dikembangkan oleh Google, umumnya digunakan pada perangkat-perangkat berbasis Linux.
Back-end	Bagian dari sistem aplikasi yang berfokus pada pengolahan data dan logika di sisi server.
Developer	Seseorang yang terlibat dalam proses pengembangan perangkat lunak, termasuk merancang, mengode, dan menguji aplikasi.
Framework	Kerangka kerja yang menyediakan struktur dan alat bantu untuk membangun aplikasi dengan lebih efisien.
Google	Perusahaan teknologi multinasional yang terkenal dengan mesin pencari dan berbagai produk dan layanannya.
iOS	Sistem operasi <i>mobile</i> yang dikembangkan oleh Apple untuk perangkat seperti iPhone dan iPad.
Multi-platform	Kemampuan sebuah aplikasi atau perangkat lunak untuk berjalan di berbagai platform atau sistem operasi yang berbeda.
Query parameter	Bagian dari URL yang digunakan untuk mengirim data tambahan kepada server saat melakukan permintaan.
Rendering engine	Komponen perangkat lunak yang bertanggung jawab untuk mengubah kode dan instruksi menjadi tampilan visual pada layar.
SDK	Kumpulan alat pengembangan perangkat lunak yang menyediakan berbagai komponen dan fungsi untuk membangun aplikasi.
Skia	Library grafis open-source yang digunakan oleh Google dalam pengembangan aplikasi, termasuk aplikasi Android dan Chrome.
Trigger	Kejadian atau perubahan yang memicu eksekusi suatu tindakan atau proses dalam aplikasi.
Usecase	Deskripsi singkat tentang bagaimana suatu aplikasi atau sistem digunakan untuk menyelesaikan masalah atau memenuhi kebutuhan pengguna.

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.....	x
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Ruang Lingkup.....	4
1.3 Tujuan	4
1.4 Manfaat	5
1.5 Sistematika Penulisan	5
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA	6
2.1 What’s on UII.....	6
2.2 Aplikasi <i>Mobile</i>	6
2.3 Flutter	7
2.4 Metode Waterfall	7
2.5 Black Box Testing.....	8
2.6 Tinjauan Pustaka	8
BAB III PELAKSANAAN MAGANG	10
3.1 Analisis Kebutuhan	11
3.1.1 Perancangan Proses Bisnis	12
3.2 Perancangan Desain Antarmuka Pengguna	18
3.3 Implementasi/Pengembangan	20
3.3.1 Menampilkan Berita	21
3.3.2 Berlangganan Penerbit	27
3.3.3 Tambah ke Favorit.....	32
3.3.4 Notifikasi Pengingat Membaca Berita	38
3.4 Pengujian.....	42
3.5 Perbandingan dengan Studi dan Aplikasi Terdahulu	43
BAB IV REFLEKSI PELAKSANAAN MAGANG	45
4.1 Relevansi Akademik	45
4.1.1 <i>Tools</i> Pengembangan Aplikasi <i>Mobile</i>	45
4.1.2 Metode Pengembangan	45
4.1.3 Koneksi <i>Back-end</i> ke <i>Front-end Mobile</i>	47
4.2 Pembelajaran Magang.....	47
BAB V PENUTUP	49
5.1 Kesimpulan	49
5.2 Saran.....	50
DAFTAR PUSTAKA	51
LAMPIRAN.....	53

DAFTAR TABEL

Tabel 3.1 Tabel analisis kebutuhan.....	12
Tabel 3.2 Tabel deskripsi <i>usecase diagram</i>	13
Tabel 3.3 Tabel hasil pengujian <i>Black Box</i>	43
Tabel 3.4 Perbedaan antar aplikasi	44

DAFTAR GAMBAR

Gambar 1.1 Struktur organisasi BSI UII.....	2
Gambar 3.1 Tahapan Waterfall pada pengembangan What's on UII.....	10
Gambar 3.2 <i>Usecase diagram</i> What's on UII.....	13
Gambar 3.3 <i>Activity diagram</i> melihat daftar berita	14
Gambar 3.4 <i>Activity diagram</i> melihat detail berita.....	15
Gambar 3.5 <i>Activity diagram</i> berlangganan penerbit berita	16
Gambar 3.6 <i>Activity diagram</i> menyalakan pengingat membaca berita	16
Gambar 3.7 <i>Activity diagram</i> membagikan tautan berita	17
Gambar 3.8 <i>Activity diagram</i> menambahkan berita ke favorit	18
Gambar 3.9 <i>Wireframe</i> halaman What's on UII.....	19
Gambar 3.10 Desain antarmuka What's on UII.....	20
Gambar 3.11 Tabel basis data berita.....	22
Gambar 3.12 <i>API list</i> berita	24
Gambar 3.13 <i>API detail</i> berita	24
Gambar 3.14 Implementasi <i>API list</i> berita pada Flutter	25
Gambar 3.15 Implementasi <i>API detail</i> berita pada Flutter	25
Gambar 3.16 Halaman What's on UII	26
Gambar 3.17 Halaman detail berita	27
Gambar 3.18 Basis data <i>subscribe</i>	28
Gambar 3.19 <i>API</i> untuk melakukan <i>subscribe</i> atau berlangganan	29
Gambar 3.20 <i>API</i> untuk menampilkan berita dari publiher langganan.....	29
Gambar 3.21 <i>API</i> untuk berhenti berlangganan	30
Gambar 3.22 Penerapan <i>dio.get</i> untuk fitur <i>subscribe</i>	31
Gambar 3.23 Penerapan <i>dio.post</i> untuk fitur <i>subscribe</i>	31
Gambar 3.24 Penerapan <i>dio.delete</i> untuk fitur <i>subscribe</i>	32
Gambar 3.25 Tombol langganan pada halaman detail berita	32
Gambar 3.26 Tabel basis data untuk fitur tambah favorit	33
Gambar 3.27 <i>API</i> untuk menampilkan berita favorit.....	34
Gambar 3.28 <i>API</i> untuk menambahkan berita ke favorit	35
Gambar 3.29 <i>API</i> untuk menghapus berita dari favorit.....	35
Gambar 3.30 penerapan <i>dio.get</i> untuk fitur tambah favorit.....	36
Gambar 3.31 Penerapan <i>dio.post</i> untuk fitur tambah favorit.....	37

Gambar 3.32 penerapan dio.delete untuk fitur tambah favorit	37
Gambar 3.33 Tombol tambah ke favorit pada halaman detail berita.....	38
Gambar 3.34 Penerapan package Flutter Local Notification.....	39
Gambar 3.35 Kode untuk memunculkan notifikasi	40
Gambar 3.36 Notifikasi mengambang dari berita di modul What's on UII	41
Gambar 3.37 Tampak notifikasi pada <i>notification shade</i> Android.....	41
Gambar 4.1 Dokumentasi tugas pada platform ClickUp	46
Gambar 4.2 Grafik penggunaan <i>framework mobile</i> multiplatform	48

BAB I

PENDAHULUAN

1.1 Latar Belakang

Badan Sistem Informasi Universitas Islam Indonesia (BSI UII) pertama kali didirikan pada tahun 1985 dengan nama Pusat Komputer dan Statistik. Namun, beberapa tahun kemudian namanya berubah menjadi Pusat Komputer, kemudian kembali menjadi Pusat Komputer dan Statistik, dan terakhir menjadi Pusat Sistem Informasi. Pada tahun 2006, nama terakhir yang dipilih adalah Badan Sistem Informasi dan masih digunakan hingga sekarang. Sejak tahun 2016, peran BSI diposisikan sebagai pelayanan bagi universitas, fakultas, dan program studi, pendamping bagi pimpinan universitas, fakultas, dan program studi, serta pengakselerasi seluruh proses bisnis UII.

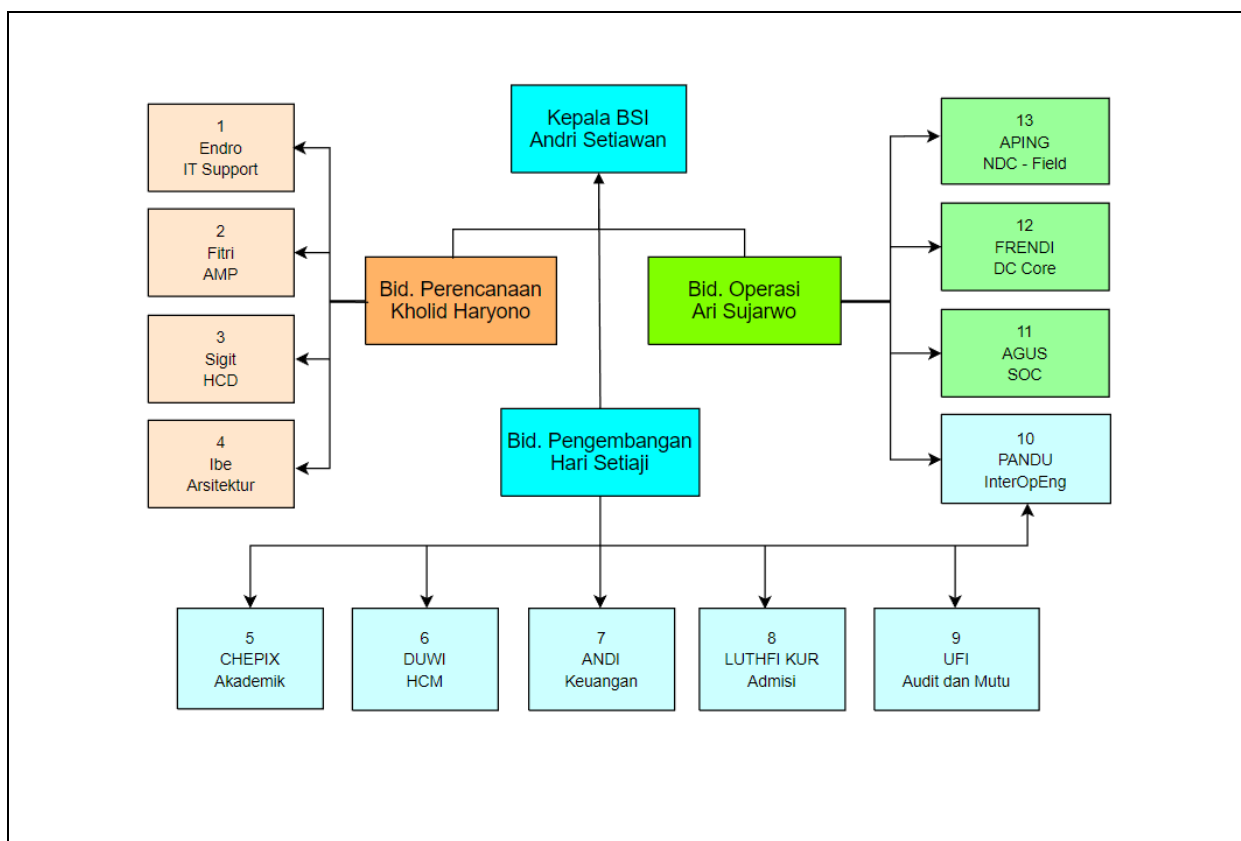
Sampai tahun akademik 2022/2023, Universitas Islam Indonesia telah memiliki delapan Fakultas, satu program Diploma Tiga, dua puluh enam program Sarjana, tiga program Sarjana Terapan, tiga program Profesi, empat belas program Magister, dan lima program Doktor. Untuk mendukung semua hasil yang dicapai UII, salah satu unit yang membantu adalah Badan Sistem Informasi (BSI). Organisasi ini dibuat pada abad ke-20 untuk memberikan layanan terbaik bagi UII dalam bidang teknologi informasi, dengan pengguna yang meliputi mahasiswa hingga para pimpinan UII.

Badan Sistem Informasi (BSI) adalah sebuah perusahaan non profit yang berada di bawah dan merupakan bagian penting dari Universitas Islam Indonesia (UII). BSI bergerak di bidang teknologi informasi yang memberikan pelayanan terkait sistem informasi dalam lingkungan Universitas Islam Indonesia. BSI memiliki peran besar dalam memenangkan persaingan UII di masa depan. Karena untuk mengembangkan kapasitasnya, UII juga harus meningkatkan kualitas layanan sistem dan teknologi informasi. BSI berperan dalam seluruh proses yang terkait dengan sistem dan teknologi informasi mulai dari perencanaan, pengembangan, operasi, hingga pelayanan dengan penggunanya yang merupakan seluruh Civitas Academica UII yang mencapai kurang lebih 30.000 pengguna (Badan Sistem Informasi Universitas Islam Indonesia, 2023).

BSI dilengkapi dengan tenaga-tenaga profesional di bidang teknologi informasi yang terdiri dari analis, *programmer*, *engineer*, dan konsultan yang berpengalaman. Dalam pelayanannya, BSI memiliki tanggung jawab dalam beberapa divisi, yaitu bidang perencanaan sistem informasi, divisi pengembangan sistem informasi, divisi operasi sistem informasi, divisi

layanan pengguna, divisi administrasi dan manajemen proyek. Dalam melaksanakan tugasnya, BSI merumuskan tiga peranan yaitu melayani, mendampingi, dan mengakselerasi. Dengan tiga peran tersebut, BSI siap mengawal kemajuan Universitas Islam Indonesia (Badan Sistem Informasi Universitas Islam Indonesia, 2023).

Gambar 1.1 menunjukkan struktur organisasi di BSI UII. BSI UII dikepalai oleh Bapak Andri Setiawan. Secara garis besar BSI memiliki tiga bidang utama, yaitu bidang Perencanaan, bidang pengembangan, dan bidang operasional. Tiap bidang tersebut dikepalai oleh seorang kepala bidang dan memiliki beberapa tim yang dapat dilihat pada bagan struktur organisasi BSI.



Gambar 1.1 Struktur organisasi BSI UII

Dalam era digital saat ini, penggunaan aplikasi *mobile* semakin menjadi kebutuhan sehari-hari. Hampir semua orang menggunakan *smartphone* untuk memudahkan berbagai aktivitas, seperti berkomunikasi, berbelanja, membaca berita, hingga bermain *game*. Berkat perkembangan teknologi, penggunaan aplikasi *mobile* semakin mudah dan praktis. Aplikasi *mobile* dapat diunduh dengan mudah dari toko aplikasi dan dapat diakses kapan saja dan di mana saja. Hal ini memudahkan pengguna untuk melakukan aktivitas apapun yang mereka inginkan dengan cepat dan mudah. Tidak mengherankan jika penggunaan aplikasi *mobile* terus

meningkat setiap tahunnya (Adisty, 2023). Aplikasi *mobile* dapat menjadi solusi dari beberapa masalah yang kita temui di sekitar kita.

Salah satu tantangan yang dihadapi oleh mahasiswa di Universitas Islam Indonesia (UII) saat ini adalah kesulitan dalam mendapatkan berita dan informasi yang diperlukan. Di UII, berita dan informasi yang relevan sering kali tersebar secara terpencar, membuat mahasiswa kesulitan untuk mengaksesnya. Sebagai contoh, pengumuman perkuliahan, informasi perpustakaan, beasiswa, dan acara kampus lainnya sering kali dipublikasikan melalui berbagai *platform* dan media yang berbeda. Hal ini menyebabkan mahasiswa harus mencari informasi secara manual dari berbagai sumber yang berbeda, yang menghabiskan waktu dan energi. Sebagai hasilnya, mahasiswa sering kali kehilangan akses cepat dan efisien ke informasi yang dibutuhkan. Untuk mengatasi masalah ini, perlu adanya upaya untuk menyediakan akses terpusat ke berita dan informasi penting di UII, sehingga mahasiswa dapat dengan mudah dan efisien mendapatkan informasi yang mereka butuhkan dalam satu tempat yang terintegrasi.

Badan Sistem Informasi Universitas Islam Indonesia (BSI UII) saat ini sedang mengembangkan sebuah aplikasi *mobile* bernama I'm UII. Aplikasi ini diharapkan dapat menjadi solusi bagi mahasiswa dalam mengatasi tantangan mendapatkan berita dan informasi di UII yang terpencar. Terdapat sebuah studi yang mengatasi permasalahan serupa (Suharto & Mardiana, 2020). Berdasarkan studi tersebut, aplikasi Eresha Mobile yang merupakan wadah untuk saling bertukar serta pendistribusian informasi, dapat mengatasi permasalahan mahasiswa STMIK Eresha dalam mengakses informasi. Permasalahan pada studi tersebut serupa dengan masalah yang ingin diselesaikan menggunakan aplikasi I'm UII, yaitu masalah terkait akses berita.

Oleh karena itu, dirancanglah sebuah fitur yang diberi nama What's on UII. Fitur ini dirancang untuk memberikan akses terpusat ke berbagai informasi terkini seputar kegiatan dan peristiwa di UII. Dengan menggunakan fitur What's on UII dalam aplikasi I'm UII, mahasiswa dapat dengan mudah mengakses berita-berita terbaru, pengumuman perkuliahan, informasi perpustakaan, beasiswa, dan acara kampus lainnya dalam satu *platform* yang terintegrasi.

Selama magang di BSI UII, penulis terlibat dalam pengembangan proyek I'm UII ini. Penulis ditempatkan sebagai seorang pengembang atau *developer*. Tim yang menangani proyek ini merupakan sebuah tim percepatan atau disebut juga dengan tim akselerasi. Tim ini merupakan tim di luar dari tim utama yang ada di BSI. Dalam tim ini, anggota tim dibagi menjadi dua divisi, yaitu divisi produk dan divisi pengembang.

Aplikasi I'm UII dikembangkan menggunakan *framework* Flutter. Dalam pengembangan proyek ini, Flutter dipilih karena Flutter dinilai dapat mempercepat proses pengembangan aplikasi (Swarna, Purnama, & Anthony, 2020). Hal ini dikarenakan Flutter adalah sebuah *framework* multi-platform yang dapat mengembangkan aplikasi *mobile* dengan dua sistem operasi sekaligus tanpa harus membuat kode yang terpisah (Rao, Pavan, Sarivastava, Amani, & Sharma, 2022).

1.2 Ruang Lingkup

Selama melaksanakan magang di BSI UII, penulis ditempatkan ke dalam tim percepatan atau biasa disebut tim akselerasi. Di dalam tim ini, penulis dimasukkan ke dalam kategori tim pengembang. Penulis diberi tugas untuk mengerjakan salah satu fitur utama aplikasi yang dinamakan What's on UII. Tugas utama penulis adalah mengerjakan dan merealisasikan fitur What's on UII ke dalam aplikasi *mobile* dengan menggunakan Flutter. Ruang lingkup penulis dalam mengerjakan proyek ini antara lain:

1. Mengimplementasikan desain UI/UX ke aplikasi *mobile*
2. Mengimplementasikan API ke aplikasi *mobile*
3. Merealisasikan fitur-fitur atau fungsionalitas pada fitur What's on UII
 - a. Menampilkan list berita: Berita terbaru, berita dari penerbit yang dilanggan, berita favorit
 - b. Menampilkan detail berita
 - c. Fitur *subscribe* atau berlangganan
 - d. Menambahkan berita ke favorit
 - e. Menambahkan pengingat membaca berita
4. Melakukan pengujian Black-box untuk fitur What's on UII

Walaupun tugas utama penulis adalah sebagai *mobile developer*, penulis juga ikut andil dalam melakukan pengembangan aplikasi *back-end*. Dalam pengembangan aplikasi *back-end* dari fitur What's on UII, penulis diberikan sebuah *template* kode yang hanya tinggal dimodifikasi sedikit untuk menghasilkan API yang siap digunakan.

1.3 Tujuan

Tujuan dari dikembangkannya fitur What's on UII pada aplikasi I'm UII adalah:

1. Agar semua berita dan juga informasi di UII baik akademik maupun non-akademik dapat diakses secara terpusat dengan satu platform dan tidak terpisah-pisah.

2. Mengetahui pengembangan aplikasi *mobile* menggunakan *framework* Flutter.

1.4 Manfaat

Manfaat yang didapatkan dari pengembangan fitur What's on UII atau berita pada aplikasi I'm UII adalah:

1. Karena berita dan informasi dapat diakses terpusat pada satu platform, pengguna dapat memperoleh informasi dan membaca berita dengan mudah.
2. Dengan Flutter, pengembangan aplikasi *mobile* dapat dilakukan untuk dua platform sekaligus yaitu Android dan iOS hanya dengan satu basis kode, sehingga dapat menghemat waktu dan anggaran dalam pengembangan aplikasi.

1.5 Sistematika Penulisan

Sistematika penulisan berisi susunan dari laporan akhir yang dapat memberikan gambaran umum dari laporan akhir. Sistematika penulisan adalah sebagai berikut:

1. BAB I: PENDAHULUAN

Bab ini berisi latar belakang, ruang lingkup magang, tujuan, manfaat, dan sistematika penulisan.

2. BAB II: LANDASAN TEORI DAN TINJAUAN PUSTAKA

Bab ini berisi definisi dan teori-teori yang menjadi landasan dalam menyusun laporan, yang memuat rangkuman dari berbagai sumber pustaka yang relevan, serta tinjauan pustaka makalah terdahulu.

3. BAB III: PELAKSANAAN MAGANG

Bab ini berisi tentang aktifitas penulis selama melakukan magang, mulai dari pekerjaan hingga metode pengembangan yang penulis gunakan.

4. BAB IV: REFLEKSI PELAKSANAAN MAGANG

Bab ini akan menjelaskan tentang refleksi dari pelaksanaan magang yang telah dilakukan. Bab ini berisi relevansi akademik dan pembelajaran yang didapat selama magang.

5. BAB V: PENUTUP

Bab ini berisi tentang kesimpulan dari keseluruhan laporan akhir ini, dan juga saran.

BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 What's on UII

Fitur What's on UII dalam aplikasi I'm UII adalah komponen yang memberikan pengguna akses terhadap berita terkini secara terpusat. Fitur ini dirancang dengan berbagai kemampuan yang memungkinkan pengguna untuk mendapatkan informasi terkini dengan mudah dan nyaman. Salah satu fitur yang disediakan adalah kemampuan untuk berlangganan penerbit berita tertentu, sehingga pengguna dapat mengikuti dan mendapatkan update terbaru dari sumber berita yang mereka percayai.

Selain itu, fitur ini juga menyediakan opsi penandaan favorit, yang memungkinkan pengguna untuk menyimpan berita yang menarik minat mereka dan dapat diakses kembali dengan cepat di kemudian hari. Pengguna juga dapat berbagi berita melalui berbagai platform media sosial atau melalui pesan langsung kepada teman-teman mereka.

Fitur What's on UII juga menawarkan notifikasi pengingat membaca, yang memberikan pengguna pengingat untuk membaca berita yang mereka inginkan. Hal ini memastikan bahwa pengguna tidak melewatkan berita penting atau menarik yang mungkin mereka lewatkan.

2.2 Aplikasi *Mobile*

Aplikasi *mobile* merupakan sebuah perangkat lunak yang dirancang khusus untuk perangkat bergerak atau *mobile* seperti telepon seluler dan tablet (Jela & Suratno, 2022). Aplikasi *mobile* biasanya dapat digunakan dengan cara dipasang ke dalam perangkat, namun ada juga beberapa yang dapat dibuka menggunakan peramban web. Aplikasi *mobile* menawarkan berbagai macam fungsi dan tujuan, mulai dari media sosial yang memungkinkan pengguna berinteraksi dengan teman dan keluarga, hingga aplikasi komunikasi seperti pesan teks dan panggilan suara melalui koneksi internet. Selain itu, ada juga aplikasi hiburan seperti *streaming* musik dan video, serta permainan yang dapat dimainkan di perangkat seluler. Aplikasi produktivitas membantu pengguna mengatur waktu, mengelola tugas, dan meningkatkan efisiensi. Aplikasi *mobile* yang baik dapat menyajikan informasi secara efisien dan praktis (Karim & Ardiansyah, 2022).

2.3 Flutter

Flutter adalah sebuah *SDK (Software Development Kit) framework* aplikasi *mobile* yang dikembangkan oleh Google (Suhendro, Sudarma, & Khrisne, 2021). Flutter dapat digunakan untuk mengembangkan aplikasi *mobile* lintas platform, memungkinkan aplikasi yang dibangun menggunakan Flutter dapat beroperasi dengan baik di kedua sistem operasi utama, yaitu Android dan iOS. Dengan menggunakan Flutter, pengembang dapat mengembangkan aplikasi berbasis Android maupun iOS hanya dengan menggunakan satu basis kode saja (Katti & K, 2022). Flutter menggunakan bahasa pemrograman Dart yang mirip gabungan antara bahasa Java dan Javascript (Arif, Kusumo, & Puspitasari, 2021). Dengan menggunakan Flutter, pengembang dapat menciptakan aplikasi dengan desain yang seragam dan mengikuti panduan desain Material Design untuk platform Android atau panduan desain Cupertino untuk platform iOS.

Flutter menawarkan berbagai *widget* yang lengkap dan fleksibel, memungkinkan pengembang untuk dengan mudah membangun antarmuka pengguna yang interaktif dan menarik bagi para pengguna. Flutter juga memiliki performa yang baik karena memiliki *rendering engine* sendiri (Jorgi, Asmawati, & Zulkarnaim, 2022). *Rendering engine* pada Flutter diberi nama Skia. Dengan memanfaatkan Skia, Flutter memiliki kemampuan untuk menggambar antarmuka pengguna secara langsung pada canvas, yang menghasilkan performa yang cepat dan responsif.

2.4 Metode Waterfall

Metode Waterfall adalah salah satu pendekatan yang digunakan dalam pengembangan perangkat lunak yang mengikuti alur tahapan yang terstruktur dan berurutan. Metode ini terdiri dari lima tahapan yang dimulai dari analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan (Zakaria, Kasih, Halim, & Gunadi, 2023). Dalam metode ini, setiap tahap pengembangan dilakukan secara berurutan dan secara linear. Setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya, dan perubahan yang terjadi pada satu tahap akan sulit untuk diterapkan pada tahap sebelumnya. Metode Waterfall menekankan perencanaan yang matang dan dokumentasi yang jelas, dengan hasil yang dihasilkan dari satu tahap menjadi masukan untuk tahap berikutnya. Metode Waterfall cocok untuk digunakan pada proyek dengan persyaratan yang stabil dan jelas, di mana perubahan yang signifikan tidak diharapkan terjadi selama proses pengembangan.

2.5 Black Box Testing

Black Box Testing, juga dikenal sebagai pengujian kotak hitam, adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem tanpa memperhatikan struktur atau logika internalnya. Pendekatan ini melihat sistem sebagai sebuah "kotak hitam" di mana hanya input dan *output* yang menjadi fokus pengujian. Tujuan utama dari metode pengujian ini adalah untuk mengetahui apakah semua fungsi dapat berjalan sesuai dengan yang telah didefinisikan (Haryono & Budi Negara, 2020).

Dalam Black Box Testing, pengujian dilakukan berdasarkan spesifikasi fungsional sistem. Penguji hanya memperhatikan input yang diberikan ke sistem dan memeriksa luaran yang dihasilkan untuk memastikan bahwa sistem berperilaku sesuai dengan harapan. Penguji tidak perlu mengetahui detail implementasi internal sistem, melainkan fokus pada bagaimana sistem berinteraksi dengan input dan menghasilkan output yang diinginkan.

2.6 Tinjauan Pustaka

Studi terdahulu yang terkait dengan skripsi ini, menyatakan bahwa dengan adanya aplikasi berita Labuhanbatu berbasis *mobile*, memudahkan masyarakat Labuhanbatu untuk mengakses informasi secara cepat, tepat, dan akurat (Duha, Sarkum, Munthe, & Purnama, 2018). Pada studi tersebut aplikasi dikembangkan menggunakan *native* Android yang berarti aplikasi ini hanya dapat berjalan pada *platform* Android. Pada aplikasi ini, pengguna dapat mengakses daftar berita dan juga detail dari setiap berita.

Studi lainnya (Suharto & Mardiana, 2020), juga menjelaskan bahwa dengan adanya aplikasi Eresha Mobile, mahasiswa dapat mengakses berita serta informasi dan tidak akan ketinggalan informasi terbaru. Pada aplikasi Eresha Mobile, notifikasi akan muncul setiap ada pembaruan berita. Hal ini akan membuat semua berita yang baru terbit akan muncul di notifikasi pengguna.

Studi terdahulu telah membahas perbedaan antara Flutter dan *native* Android maupun iOS dalam pengembangan aplikasi *mobile* (Rao, Pavan, Sarivastava, Amani, & Sharma, 2022). Studi ini membuat aplikasi yang dikembangkan menggunakan Flutter, *native* Android, dan *native* iOS. Aplikasi yang dibuat menampilkan tampilan dan juga fitur yang sama. Beberapa parameter yang dijadikan perbandingan diantaranya adalah performa CPU, efisiensi kode, dan juga tampilan. Setelah melakukan perbandingan, telah ditarik kesimpulan bahwa performa CPU Flutter (Android dan iOS), *native* Android, dan *native* iOS memiliki perbedaan yang sangat kecil. Namun jika dibandingkan dengan kinerja Flutter di iOS dan Android, terlihat

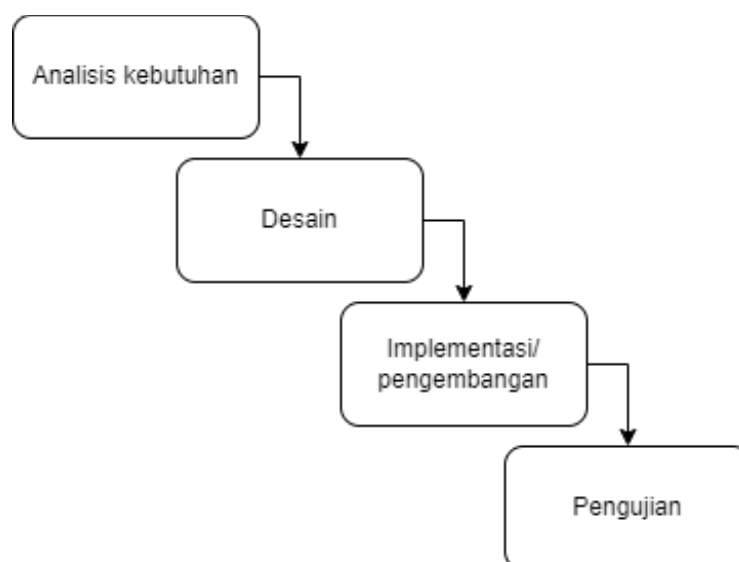
bahwa Flutter berperforma lebih baik daripada aplikasi *native*. Selain itu, efisiensi kode pada Flutter juga lebih baik. Aplikasi Flutter memiliki jumlah baris yang lebih sedikit daripada baris kode yang ditulis pada *native* iOS dan android. Bagian terbaiknya adalah Flutter tidak perlu menulis kode yang terpisah untuk Android dan iOS.

BAB III

PELAKSANAAN MAGANG

Pengembangan aplikasi *mobile* I'm UII dikerjakan di dalam sebuah tim. Dalam pengembangannya, metode yang diterapkan adalah metode Waterfall. Seperti yang sudah dijelaskan sebelumnya, penulis ditempatkan ke dalam sebuah tim percepatan yang bukan merupakan tim utama di BSI UII. Sehingga anggota tim yang merupakan karyawan hanya dapat mengerjakan proyek di luar jam kerja. Hal ini membuat komunikasi antara tim magang dan karyawan sangat terbatas. Oleh karena itu, tidak memungkinkan jika menggunakan metode *agile* yang notabene merupakan metode yang menerapkan iterasi. Metode Waterfall dinilai tepat karena pengerjaan proyek dilakukan secara terstruktur dan beruntun sesuai dengan instruksi di awal pengerjaan proyek. Pada pengerjaan proyek ini, ruang lingkup pekerjaan penulis adalah pengembangan fitur What's on UII.

Tahapan yang dilakukan dalam pengembangan What's on UII dimulai dari analisis kebutuhan, desain, implementasi, dan pengujian seperti yang diilustrasikan pada Gambar 3.1. Saat ini pengerjaan What's on UII pada aplikasi I'm UII telah mencapai tahap pengembangan dan pengujian. Oleh karena itu, batas pembahasan pada skripsi ini hanyalah sampai tahap pengujian.



Gambar 3.1 Tahapan Waterfall pada pengembangan What's on UII

3.1 Analisis Kebutuhan

Pada tahap ini, kebutuhan potensial dari aplikasi dianalisis secara terstruktur dan diungkapkan dalam dokumen spesifik yang menjadi dasar bagi semua langkah pengembangan yang akan datang. Hasil dari analisis ini adalah dokumen persyaratan yang menggambarkan apa yang harus dicapai oleh aplikasi.

Untuk menganalisis kebutuhan pengguna, dilakukan survei dengan menggunakan metode wawancara. Metode wawancara digunakan untuk mendapatkan pemahaman yang mendalam tentang kebutuhan, preferensi, dan harapan pengguna terhadap aplikasi. Dalam survei ini, sejumlah mahasiswa yang masih aktif kuliah merupakan responden yang dipilih untuk diwawancarai secara langsung. Wawancara dilakukan dengan mengajukan serangkaian pertanyaan terstruktur untuk menggali informasi tentang tujuan pengguna, tugas-tugas yang ingin mereka lakukan, kendala yang mereka hadapi, serta fitur dan fungsi yang diharapkan. Survei dengan metode wawancara ini menjadi landasan penting dalam pengembangan aplikasi I'm UII, sehingga aplikasi dapat memberikan solusi yang efektif dan memenuhi kebutuhan pengguna secara optimal.

Dari wawancara tersebut, didapati bahwa mahasiswa UII saat ini memiliki beberapa kendala dalam masa perkuliahan. Salah satu kendala yang dialami oleh banyak mahasiswa adalah terkait akses berita dan informasi. Dari permasalahan tersebut, dapat dianalisis beberapa kebutuhan pengguna seperti pada Tabel 3.1.

Tabel 3.1 merupakan tabel analisis kebutuhan pengguna yang berisi kebutuhan pengguna serta skenario sistem untuk memenuhi kebutuhan pengguna tersebut. Pada Tabel 3.1 terdapat tiga kolom. Kolom pertama merupakan kolom untuk pengguna, dalam hal ini adalah mahasiswa. Setelah itu, terdapat kolom kebutuhan yang berisi kebutuhan-kebutuhan pengguna berdasarkan analisis dari hasil survei. Terdapat juga kolom skenario sistem. Kolom skenario sistem memuat hal apa saja yang dapat sistem lakukan untuk memenuhi kebutuhan pengguna.

Setelah melakukan analisis kebutuhan pengguna, langkah selanjutnya adalah perancangan proses bisnis. Perancangan proses bisnis bertujuan untuk menggambarkan alur kerja atau rangkaian aktivitas yang harus dilakukan dalam menjalankan fungsi-fungsi utama aplikasi.

Tabel 3.1 Tabel analisis kebutuhan

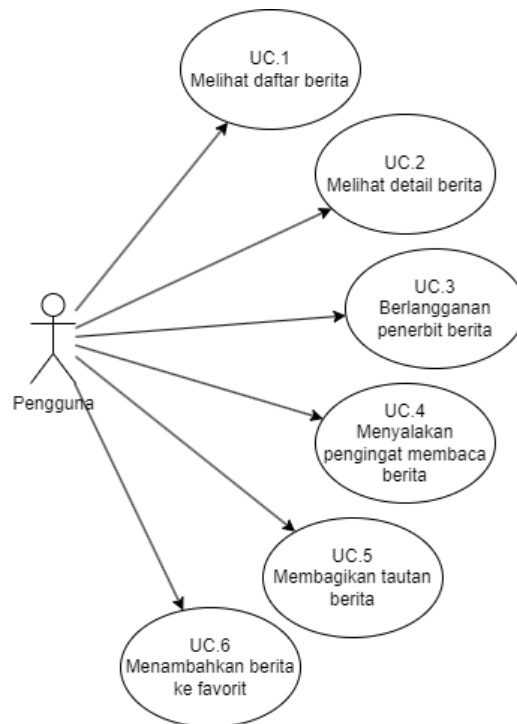
Pengguna	Kebutuhan	Skenario Sistem
Mahasiswa	Pengguna dapat mengakses berita dan informasi secara terpusat	Sistem mendapatkan akses dari sumber berita dan informasi yang ada di UII dan dapat menampilkannya.
	Pengguna dapat melakukan personalisasi untuk mendapatkan berita	Sistem menyediakan fitur agar pengguna dapat berlangganan penerbit berita yang dikehendaki. Sistem dapat menampilkan berita hanya dari penerbit yang telah dilanggan pengguna.
	Pengguna dapat menyimpan berita	Sistem menyediakan fitur agar pengguna dapat menyimpan berita, seperti menambahkan berita ke daftar favorit.
	Pengguna bisa mendapatkan notifikasi agar tidak melewatkan informasi	Sistem menyediakan fitur agar mahasiswa dapat mengatur dan mendapatkan notifikasi pengingat
	Pengguna dapat berbagi informasi sesama mahasiswa	Sistem menyediakan fitur untuk membagikan berita melalui tautan

3.1.1 Perancangan Proses Bisnis

Usecase Diagram

Usecase diagram merupakan diagram yang memodelkan fungsionalitas sistem. Di dalam *usecase diagram*, terdapat dua komponen utama yaitu aktor dan juga *use case*. *Usecase diagram* untuk fitur What's on UII pada aplikasi I'm UII dapat dilihat pada Gambar 3.2. Adapun penjelasan atau deskripsi dari tiap *usecase*, dapat dilihat pada Tabel 3.2. Pada sistem yang dirancang untuk fitur What's on UII hanya terdapat satu aktor yang terlibat, yaitu pengguna. Fitur What's on UII pada aplikasi I'm UII tidak memerlukan aktor yang berperan sebagai admin karena berita pada aplikasi I'm UII terintegrasi dengan aplikasi lain yang menjadi suplai berita.

Pada Tabel 3.2, terdapat beberapa kolom yaitu kode, nama *usecase*, dan deskripsi. Kolom kode berisi kode yang digunakan untuk merepresentasikan *usecase*. Kode yang penulis gunakan adalah UC yang ditambah dengan nomor urut. Kolom berikutnya memuat nama dari setiap *usecase* dan kolom terakhir adalah deskripsi dari setiap *usecase*.



Gambar 3.2 Usecase diagram What's on UII

Tabel 3.2 Tabel deskripsi usecase diagram

Kode	Nama Usecase	Deskripsi
UC.1	Melihat daftar berita	Proses melihat daftar berita yang ditampilkan oleh sistem. Pengguna membuka halaman What's on UII atau berita, kemudian pengguna akan dapat melihat daftar berita yang tersedia.
UC.2	Melihat detail berita	Proses di mana pengguna membaca keseluruhan isi berita. Pengguna dapat melihat detail berita dengan menekan atau memilih salah satu berita dari daftar berita.
UC.3	Berlangganan penerbit berita	Proses di mana pengguna melakukan <i>subscribe</i> atau berlangganan terhadap penerbit berita. Pengguna menekan tombol berlangganan yang berada di halaman detail berita.
UC.4	Menyalakan pengingat membaca berita	Proses di mana pengguna mengaktifkan notifikasi pengingat untuk membaca berita. Pengguna dapat memilih waktu munculnya notifikasi pengingat.
UC.5	Membagikan tautan berita	Proses di mana pengguna membagikan tautan yang menuju ke suatu berita.
UC.6	Menambahkan berita ke favorit	Proses menambahkan berita ke dalam daftar berita favorit yang dapat diakses di halaman daftar berita

Activity diagram

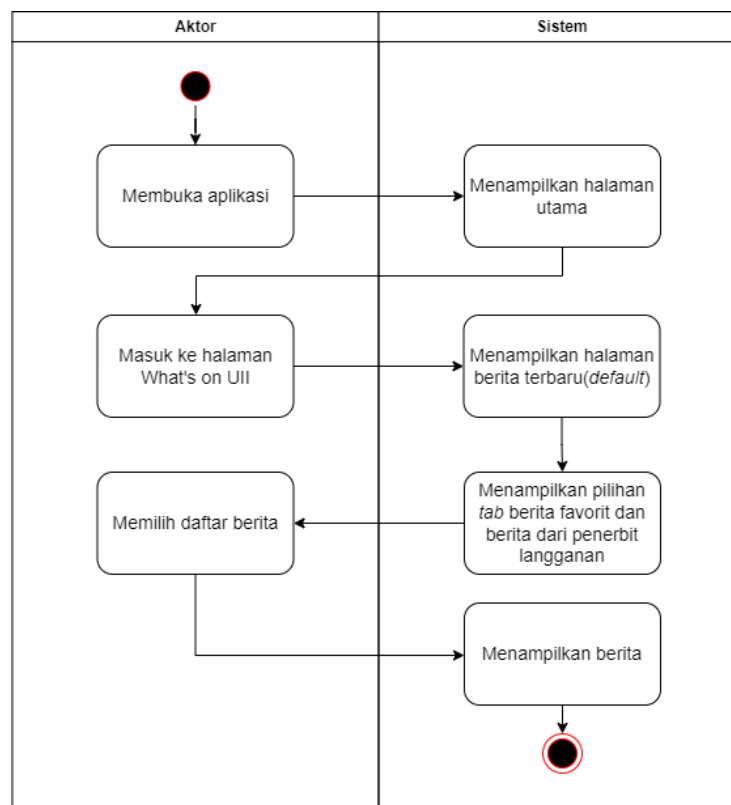
Alur pada fitur What's on UII di aplikasi I'm UII direpresentasikan menggunakan *activity diagram*. Berikut merupakan *activity diagram* dari fitur What's on UII:

a. Melihat daftar berita

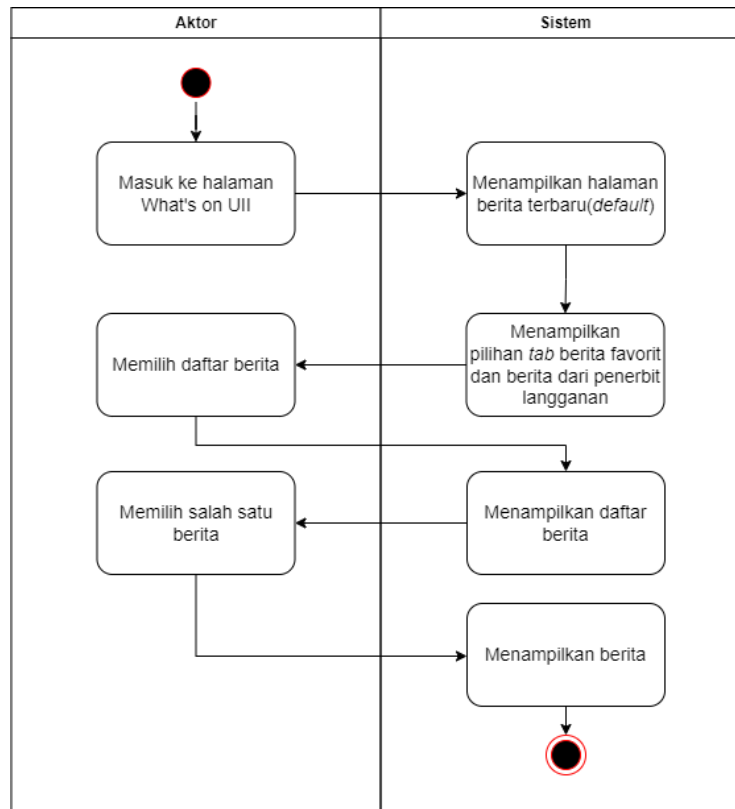
Pada alur ini, diawali dengan pengguna berada di halaman utama atau *home page* dari aplikasi. Setelah itu pengguna dapat menuju ke halaman berita dengan menekan tombol berita atau What's on UII. Alur aktivitas diperjelas pada Gambar 3.3.

b. Melihat detail berita

Alur ini menunjukkan proses bagaimana pengguna melihat detail isi dari berita. Alur ini dimulai dari pengguna yang sudah berada di halaman berita, kemudian pengguna memilih salah satu berita dari daftar berita untuk dilihat detailnya. Detail dari alur aktivitas ini dapat dilihat pada *activity diagram* yang ditunjukkan Gambar 3.4.



Gambar 3.3 Activity diagram melihat daftar berita



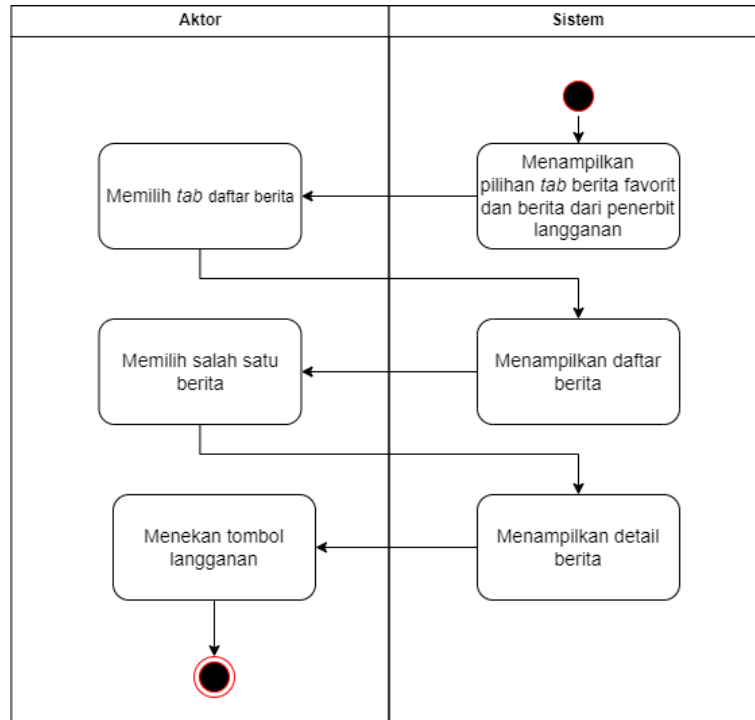
Gambar 3.4 *Activity diagram* melihat detail berita

c. Berlangganan penerbit berita

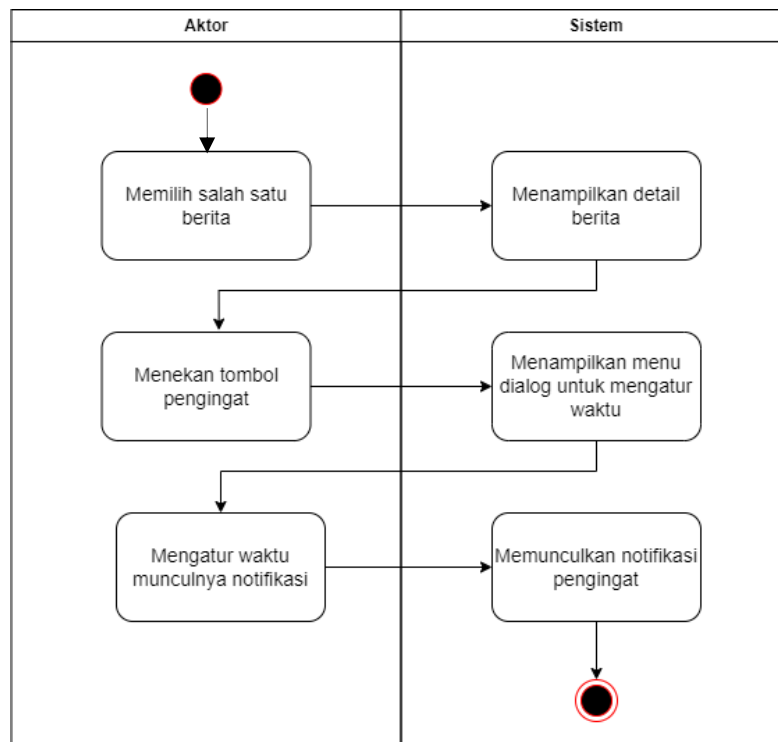
Alur ini menunjukkan proses bagaimana pengguna melakukan langganan terhadap penerbit berita. Hal ini dimulai dari halaman detail berita. Detail dari alur aktivitas ini, ditunjukkan pada *activity diagram* yang ditampilkan pada Gambar 3.5.

d. Menyalakan pengingat membaca berita

Ketika pengguna menemukan berita atau informasi yang ingin dibaca, dan ia belum berkesempatan membaca, pengguna bisa menegaktifkan pengingat dengan waktu yang dapat ia tentukan kedepannya. Alur dari menyalakan pengingat membaca berita pada aplikasi ini dapat dilihat pada *activity diagram* di Gambar 3.6.



Gambar 3.5 Activity diagram berlangganan penerbit berita



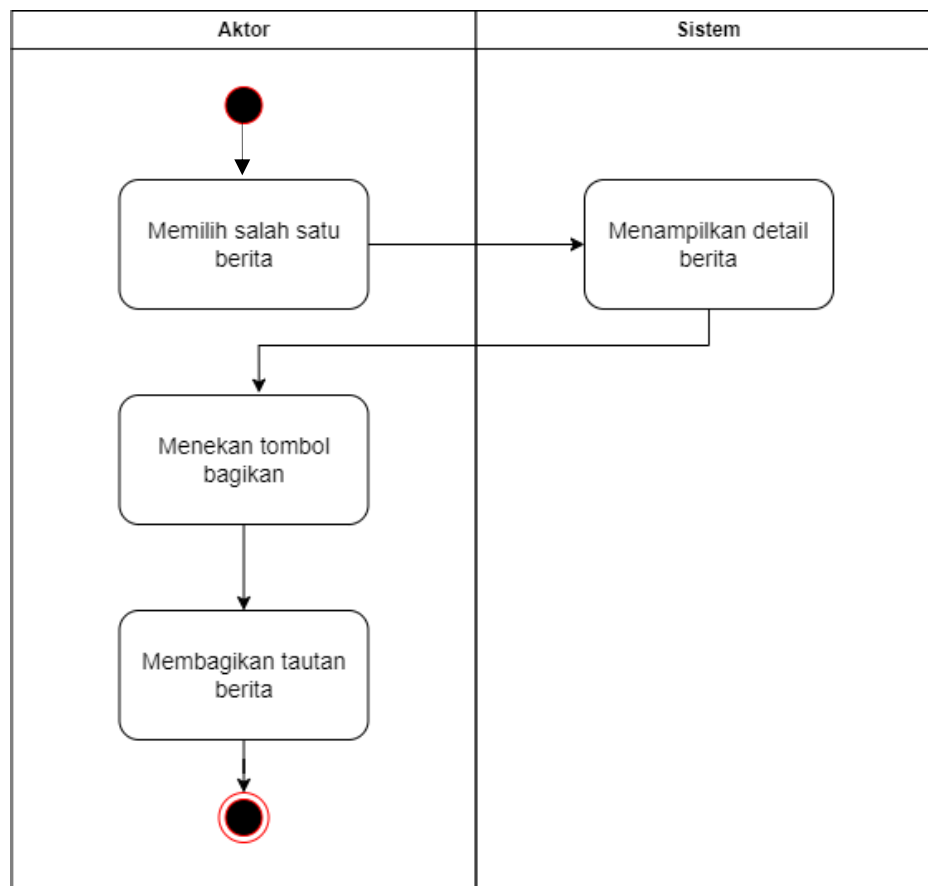
Gambar 3.6 Activity diagram menyalakan pengingat membaca berita

e. Membagikan tautan berita

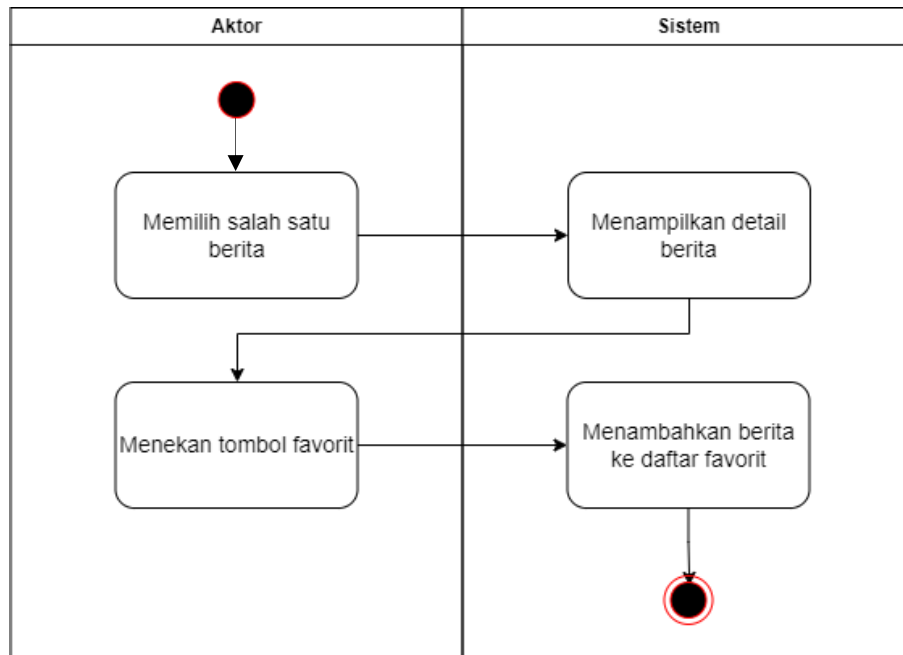
Pengguna dapat berbagi berita ke sesama civitas academica UII dengan alur yang ditunjukkan pada Gambar 3.7. Alur dimulai dari pengguna membuka detail halaman berita, kemudian menekan tombol bagikan. Untuk lebih jelasnya dapat dilihat pada *activity diagram* di Gambar 3.7.

f. Menambahkan berita ke favorit

Pengguna juga dapat menambahkan berita ke favorit agar berita atau informasi tersebut tidak tertumpuk dengan berita lain. Alur untuk aktivitas menambahkan berita ke favorit dimulai dari membuka berita yang diinginkan, kemudian pengguna dapat menekan tombol untuk menambahkan berita tersebut ke favorit. Alur dari aktivitas ini dapat dilihat pada *activity diagram* di Gambar 3.8.



Gambar 3.7 *Activity diagram* membagikan tautan berita



Gambar 3.8 *Activity diagram* menambahkan berita ke favorit

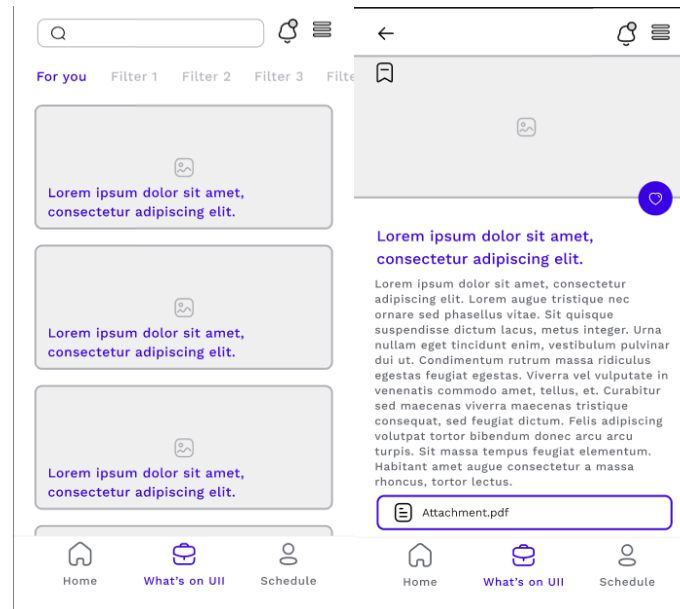
3.2 Perancangan Desain Antarmuka Pengguna

Perancangan desain antarmuka pengguna adalah proses menciptakan tampilan visual dan interaksi yang menarik, intuitif, dan efektif untuk pengguna dalam mengoperasikan suatu aplikasi atau sistem. Pada tahap ini, elemen-elemen seperti tata letak, warna, tipografi, ikon, dan kontrol interaktif dirancang dengan mempertimbangkan prinsip-prinsip desain yang baik dan kebutuhan pengguna. Tujuan dari perancangan desain antarmuka pengguna adalah menciptakan pengalaman pengguna yang menyenangkan, mudah digunakan, dan memfasilitasi pencapaian tujuan pengguna dengan efisiensi. Hasilnya adalah desain visual yang menggambarkan struktur dan tata letak antarmuka, termasuk halaman layar, navigasi, elemen-elemen interaktif, dan kejelasan komunikasi visual.

Perancangan desain antarmuka pengguna dilakukan oleh tim produk dengan menggunakan *platform* Figma. Figma merupakan sebuah aplikasi desain yang berbasis *cloud* dan juga berfungsi sebagai alat *prototyping* untuk proyek digital. Figma dirancang dengan tujuan untuk mempermudah kolaborasi antara pengguna dalam sebuah proyek dan memungkinkan mereka bekerja secara tim secara fleksibel di mana pun mereka berada (Pramudita, Arifin, Alfian, Safitri, & Anwariya, 2021).

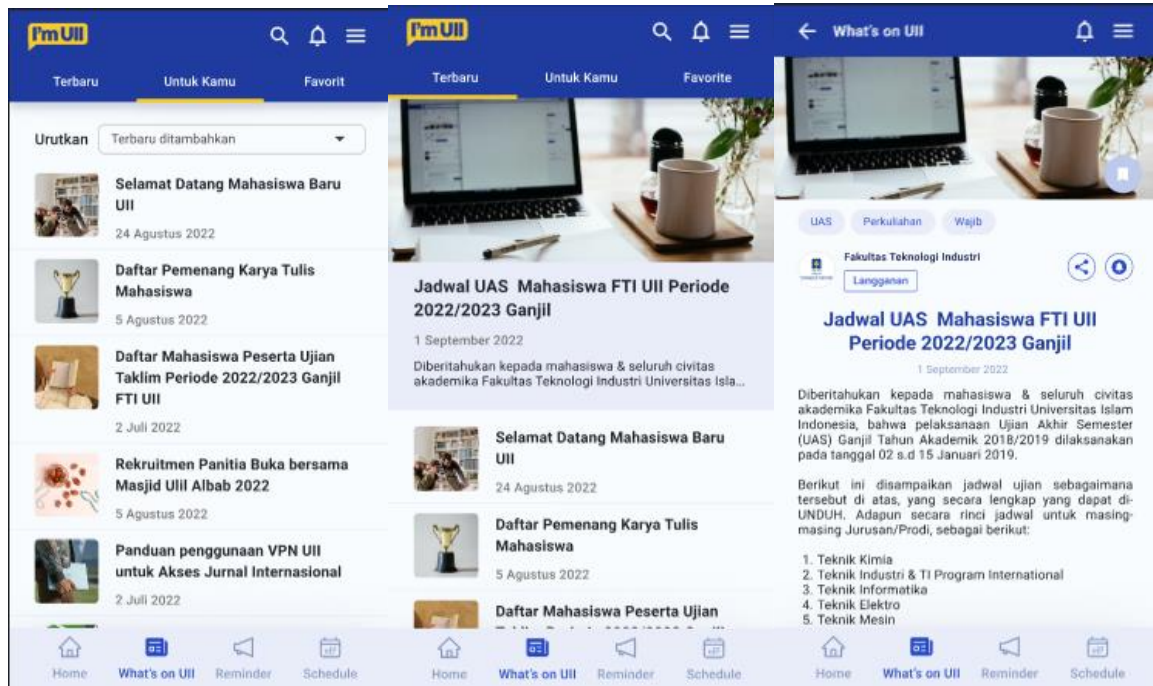
Beberapa contoh desain tersebut dapat dilihat pada Gambar 3.9 yang merupakan desain antarmuka halaman What's on UII yang masih berupa *wireframe*. *Wireframe* merupakan sketsa

awal atau kerangka dasar untuk mengatur dan menyusun elemen-elemen dalam sebuah halaman website sebelum proses desain sebenarnya dimulai (R & Saputra, 2023).



Gambar 3.9 Wireframe halaman What's on UII

Gambar 3.10 merupakan desain halaman What's on UII yang sudah diberi pewarnaan dan disempurnakan menjadi desain antarmuka yang siap diimplementasikan ke aplikasi *mobile*. Seperti yang dapat kita lihat pada Gambar 3.10, warna dasar dari aplikasi I'm UII adalah warna biru dengan aksen kuning, dan juga dibarengi dengan beberapa warna turunannya. Warna tersebut dipilih berdasarkan tema warna UII yang telah ditetapkan pada pedoman penggunaan merek Universitas Islam Indonesia (Bidang Hubungan Masyarakat Universitas Islam Indonesia, 2023). Hal ini dilakukan agar aplikasi I'm UII memiliki ciri khas UII sekaligus menunjukkan bahwa aplikasi ini adalah buatan UII.



Gambar 3.10 Desain antarmuka What's on UII

Pada halaman What's on UII, terdapat tiga *tab* utama yang berisi berita terbaru, berita “untuk kamu” yang merupakan berita dari penerbit yang sudah dilanggan pengguna, dan juga berita favorit. Selain itu pada *tab* berita terbaru, terdapat berita yang ditampilkan sebagai *headline*. Berita yang tampil di *headline* merupakan berita dengan tingkat urgensi yang lebih tinggi dari berita lainnya, dan bisa juga berita yang paling hangat atau paling baru.

Aplikasi I'm UII juga dilengkapi dengan *bottom navigation bar* yaitu tombol navigasi yang berada di bagian bawah tampilan aplikasi. Tombol navigasi ini dibuat agar pengguna dapat berpindah halaman fitur utama aplikasi dengan mudah.

3.3 Implementasi/Pengembangan

Seperti yang sudah dijelaskan sebelumnya, What's on UII merupakan salah satu fitur utama dari aplikasi I'm UII. What's on UII merupakan fitur yang dibuat berdasarkan kebutuhan pengguna yang mana merupakan sebuah fitur yang menjadi wadah dari berita-berita dan informasi-informasi yang ada di UII. Dengan adanya fitur ini, berita dan informasi yang tadinya tersebar dapat dikumpulkan menjadi satu dan mudah untuk diakses oleh pengguna. Fungsionalitas utama dari What's on UII adalah menampilkan berita, namun What's on UII juga dilengkapi dengan beberapa fungsionalitas ataupun fitur pendukung. Beberapa diantaranya adalah menambahkan berita atau informasi ke favorit, berlangganan ke penerbit berita, membuat pengingat untuk membaca berita ataupun informasi, dan membagikan berita.

Pada tahapan ini, penulis bertugas untuk merealisasikan semua fitur-fitur tersebut. Khusus untuk fitur bagikan berita, dikerjakan oleh rekan penulis. Dalam merealisasikan What's on UII, penulis melakukan beberapa tahapan, mulai dari mengimplementasi desain antarmuka, merancang basis data, membuat aplikasi *back-end* dan API, hingga menerapkan atau mengimplementasikan API di Flutter.

Dalam pengembangan fitur What's on UII, pengembangan aplikasi *back-end* dilakukan menggunakan sebuah *framework* dari Laravel yang bernama Lumen. Lumen adalah sebuah *framework* PHP yang dirancang khusus untuk membuat aplikasi *microservices* dan API. Dengan Lumen, penulis membuat beberapa *endpoint* API yang dibutuhkan dalam fitur What's on UII. Beberapa API yang dibutuhkan adalah API untuk mendapatkan berita, mendapatkan detail berita, menambahkan berita ke daftar favorit, menghapus berita dari daftar favorit, berlangganan dan berhenti berlangganan kepada penerbit berita.

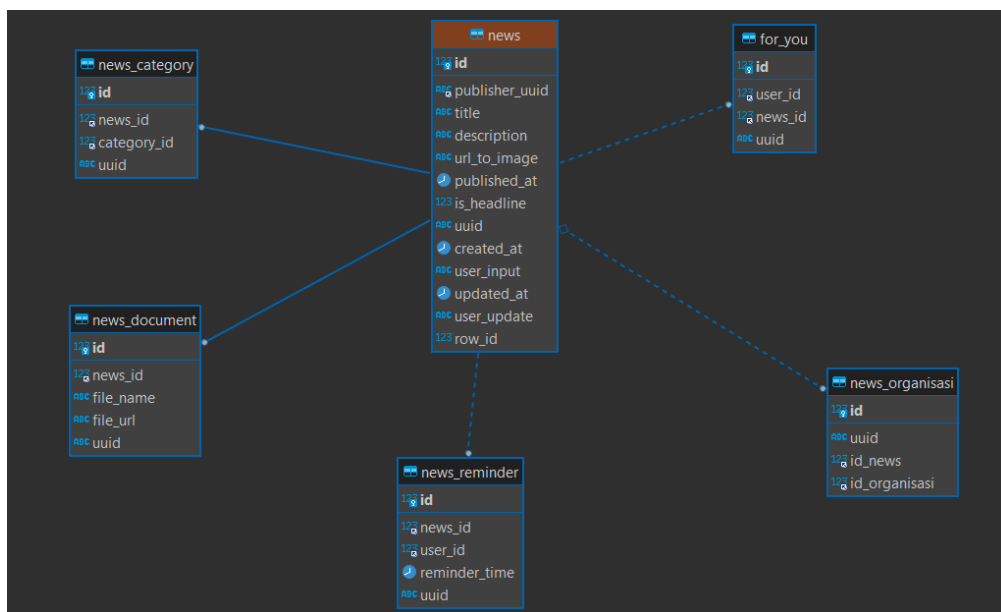
3.3.1 Menampilkan Berita

What's on UII merupakan salah satu fitur utama pada aplikasi I'm UII untuk menampilkan berita. Untuk merealisasikan agar fitur dapat menampilkan berita, penulis memulai pengerjaan dengan mengimplementasikan desain tampilan antarmuka dari *prototype* yang sudah dibuat oleh tim produk, menjadi sebuah aplikasi *mobile*. Terkait penampilan berita ini sendiri dikelompokkan menjadi tiga, yaitu berita terbaru, berita yang sudah dipersonalisasi berdasarkan penerbit yang dilanggan, dan berita yang ditambahkan ke favorit.

Sebelum ditampilkan di aplikasi, tentu saja berita-berita tersebut disimpan dan dimuat dalam basis data. Oleh karena itu, penulis merancang sebuah basis data yang diperlukan untuk fungsionalitas menampilkan berita. Rancangan utama dari basis data berita adalah seperti yang ditampilkan pada Gambar 3.11. Dapat dilihat bahwa di situ terdapat beberapa tabel yaitu *news*, *for_you*, *news_category*, *news_document*, *news_organisasi*, dan *news_reminder*. Tabel utama dalam fitur menampilkan adalah tabel *news*. Tabel *news* berisi data-data yang diperlukan untuk menampilkan *list* berita dan juga detail berita seperti judul, nama penerbit, deskripsi, dan lainnya. Selain tabel *news*, dalam menampilkan berita juga diperlukan beberapa tabel lain seperti *news_document* untuk menampilkan dokumen berita, *news_category* untuk menampilkan kategori berita, kemudian ada juga tabel *for_you* yang mana digunakan untuk menampilkan berita sesuai dengan personalisasi pengguna. Saat ini tabel *for_you* masih belum digunakan karena untuk personalisasi saat ini masih menampilkan berita dari penerbit yang

dilanggan. Pada awalnya personalisasi direncanakan menggunakan *machine learning*. Namun karena kurangnya sumber daya, saat ini masih belum terealisasi.

Pada Gambar 3.11 juga terdapat tabel *news_organisasi*, tabel ini digunakan untuk menampilkan berita sesuai dengan pengelompokkan organisasi di UII. Tabel ini digunakan untuk fitur filter. Pengguna dapat melakukan filter berita sesuai dengan organisasi yang diinginkan. Terdapat juga tabel *news_reminder*, tabel ini awalnya digunakan untuk keperluan notifikasi pengingat membaca berita, namun karena penulis menggunakan *library* Flutter Local Notification, tabel ini hingga saat ini tidak digunakan dan kemungkinan akan dihapus.



Gambar 3.11 Tabel basis data berita

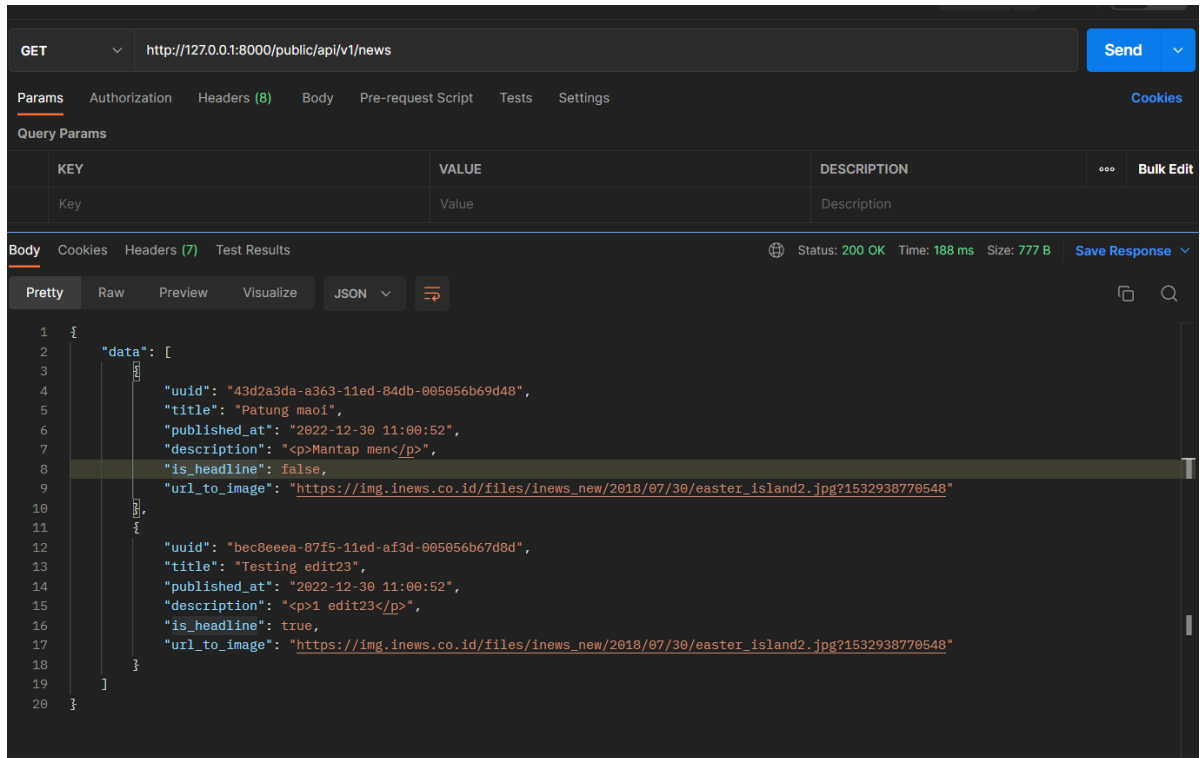
Terkait input data berita ke basis data What's on UII, dilakukan dengan mengintegrasikan basis data news pada What's on UII ke basis data UII Berita. UII Berita merupakan sebuah aplikasi berita berbasis web yang juga sedang dikembangkan BSI. Pengintegrasian basis data ini dilakukan menggunakan sebuah *tools* yang bernama Kafka. Dalam membuat skrip Kafka, penulis dibantu oleh senior karyawan BSI dengan tabel basis data news pada I'm UII sebagai destinasi, dan tabel basis data pada UII berita sebagai sumber. Sehingga setiap dilakukan input pada basis data sumber yaitu UII Berita, data tersebut akan otomatis terinput ke dalam basis data destinasi yaitu tabel basis data news pada I'm UII.

Untuk menampilkan berita ke aplikasi *mobile*, penulis mengembangkan aplikasi *back-end* yang akan menghasilkan sebuah API. Dalam menampilkan berita *request* yang dibutuhkan adalah GET. Untuk menampilkan berita sendiri membutuhkan dua API, yaitu API untuk

menampilkan *list* berita yang dapat dilihat pada Gambar 3.12, dan API untuk menampilkan detail berita yang dapat dilihat pada Gambar 3.13. Data yang ditampilkan pada *list* berita hanyalah gambar, judul, tanggal publikasi, dan deskripsi yang nantinya akan dibatasi pada sisi *mobile*. Terdapat juga data lain yang tidak ditampilkan di *mobile* tapi digunakan untuk keperluan logika bisnis seperti data UUID, dan juga data *boolean* untuk menentukan berita tersebut masuk ke berita *headline* atau tidak. Sedangkan data yang ditampilkan pada detail berita bukan berupa *list* dan menampilkan data yang lebih banyak dan detail.

Setelah API tersedia, yang penulis lakukan adalah mengimplementasi API tersebut pada aplikasi *mobile*. Seperti yang sudah dijelaskan sebelumnya, untuk mengimplementasi API pada Flutter, penulis menggunakan sebuah *package* bernama Dio. Untuk menampilkan list berita, penulis menggunakan *method* `dio.get` seperti pada Gambar 3.14, *method* tersebut dimasukkan ke dalam fungsi yang penulis beri nama `getListNews`.

Untuk menampilkan detail berita, masih menggunakan *method* yang sama yaitu `dio.get`. Namun, seperti yang ditampilkan pada Gambar 3.15, untuk menampilkan detail berita membutuhkan UUID pengguna sebagai *query parameter*, dan juga membutuhkan UUID berita pada URL path. UUID berita pada *path* menspesifikkan berita yang ingin ditampilkan detailnya, dan UUID pengguna pada *query parameter* digunakan untuk mendapatkan informasi seperti status langganan dan status favorit dari berita.



GET <http://127.0.0.1:8000/public/api/v1/news> Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

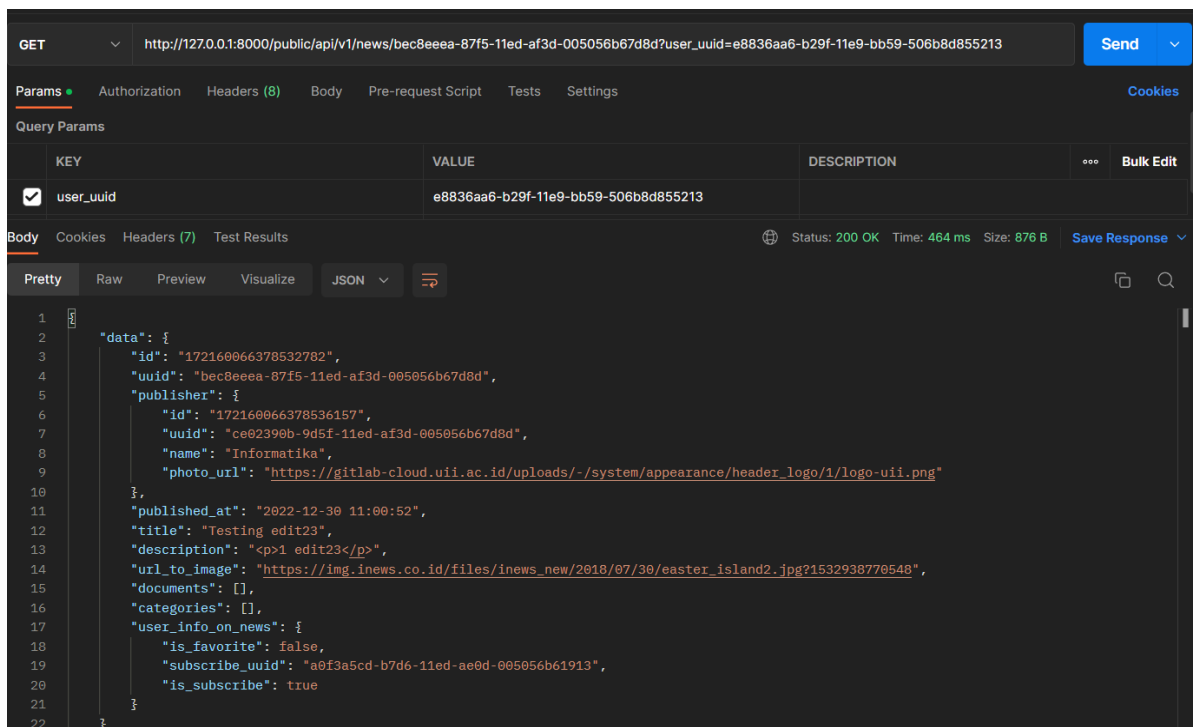
Body Cookies Headers (7) Test Results Status: 200 OK Time: 188 ms Size: 777 B Save Response

Pretty Raw Preview Visualize JSON Copy Search

```

1  {
2    "data": [
3      {
4        "uuid": "43d2a3da-a363-11ed-84db-005056b69d48",
5        "title": "Patung maol",
6        "published_at": "2022-12-30 11:00:52",
7        "description": "<p>Mantap men</p>",
8        "is_headline": false,
9        "url_to_image": "https://img.inews.co.id/files/inews_new/2018/07/30/easter_island2.jpg?1532938770548"
10       },
11     ],
12     "data": [
13       {
14         "uuid": "bec8eaea-87f5-11ed-af3d-005056b67d8d",
15         "title": "Testing edit23",
16         "published_at": "2022-12-30 11:00:52",
17         "description": "<p>1 edit23</p>",
18         "is_headline": true,
19         "url_to_image": "https://img.inews.co.id/files/inews_new/2018/07/30/easter_island2.jpg?1532938770548"
20       }
21     ]
22   }

```

Gambar 3.12 API *list* berita


GET http://127.0.0.1:8000/public/api/v1/news/bec8eaea-87f5-11ed-af3d-005056b67d8d?user_uuid=e8836aa6-b29f-11e9-bb59-506b8d855213 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> user_uuid	e8836aa6-b29f-11e9-bb59-506b8d855213			

Body Cookies Headers (7) Test Results Status: 200 OK Time: 464 ms Size: 876 B Save Response

Pretty Raw Preview Visualize JSON Copy Search

```

1  {
2    "data": {
3      "id": "172160066378532782",
4      "uuid": "bec8eaea-87f5-11ed-af3d-005056b67d8d",
5      "publisher": {
6        "id": "172160066378536157",
7        "uuid": "ce02390b-9d5f-11ed-af3d-005056b67d8d",
8        "name": "Informatika",
9        "photo_url": "https://gitlab-cloud.uui.ac.id/uploads/-/system/appearance/header_logo/1/logo-uu.png"
10       },
11      "published_at": "2022-12-30 11:00:52",
12      "title": "Testing edit23",
13      "description": "<p>1 edit23</p>",
14      "url_to_image": "https://img.inews.co.id/files/inews_new/2018/07/30/easter_island2.jpg?1532938770548",
15      "documents": [],
16      "categories": [],
17      "user_info_on_news": {
18        "is_favorite": false,
19        "subscribe_uuid": "a0f3a5cd-b7d6-11ed-ae0d-005056b61913",
20        "is_subscribe": true
21       }
22     }

```

Gambar 3.13 API detail berita

```

Future<News> getListNews(BuildContext context) async {
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news";
      //Membuat request
      final response = await dio.get(
        path,
        options: Options(
          headers: {
            "x-app": "tes",
            "x-menu": "tes"
          }
        ),
      );
      if(response.statusCode == 200) {
        // Mengelola data yang diterima dari server
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        return News.fromJson(resData);
      }
    }
    .....
  }
}

```

Gambar 3.14 Implementasi API list berita pada Flutter

```

Future<NewsDetail> getNewsDetail(BuildContext context, String newsUuid) async {
  userUuid = await Utils.getString('user_uuid');
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news/$newsUuid";
      final response = await dio.get(
        path,
        options: Options(
          headers: {
            "x-app": "tes",
            "x-menu": "tes"
          }
        ),
        queryParameters: {
          "user_uuid" : userUuid
        }
      );
      if(response.statusCode == 200) {
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        // return NewsDetail.fromJson(json.decode(resData));
        return NewsDetail.fromJson(resData);
      }
    }
    .....
  }
}

```

Gambar 3.15 Implementasi API detail berita pada Flutter

Hasil dari implementasi fungsionalitas menampilkan berita dapat di lihat pada Gambar 3.16 yang merupakan halaman What's on UII yang menunjukkan *list* berita terbaru. Pada halaman tersebut, terdapat tiga *tab* yang mana pengguna dapat melihat berita berdasarkan berita terbaru, berita dari penerbit yang sudah dilanggan, dan berita yang ditambahkan ke favorit. Hal

ini dapat memudahkan pengguna dalam mencari dan membaca berita yang relevan bagi dirinya. Pada *tab* berita terbaru, sesuai dengan panduan desain dari Tim Produk, terdapat berita *headline* yang menampilkan berita dengan urgensi lebih tinggi atau berita paling baru. Terdapat juga berita-berita lainnya yang dikemas dalam *list* berita.

Gambar 3.17 menunjukkan halaman detail berita yang berisi informasi berita yang lebih lengkap seperti keseluruhan deskripsi berita, informasi penerbit, dan juga informasi lainnya seperti lampiran dokumen dan juga kategori berita. Pada halaman detail juga terdapat tombol-tombol aksi untuk menjalankan fitur-fitur tambahan seperti menambahkan berita ke favorit, berlangganan penerbit, mengatur pengingat, dan membagikan berita.



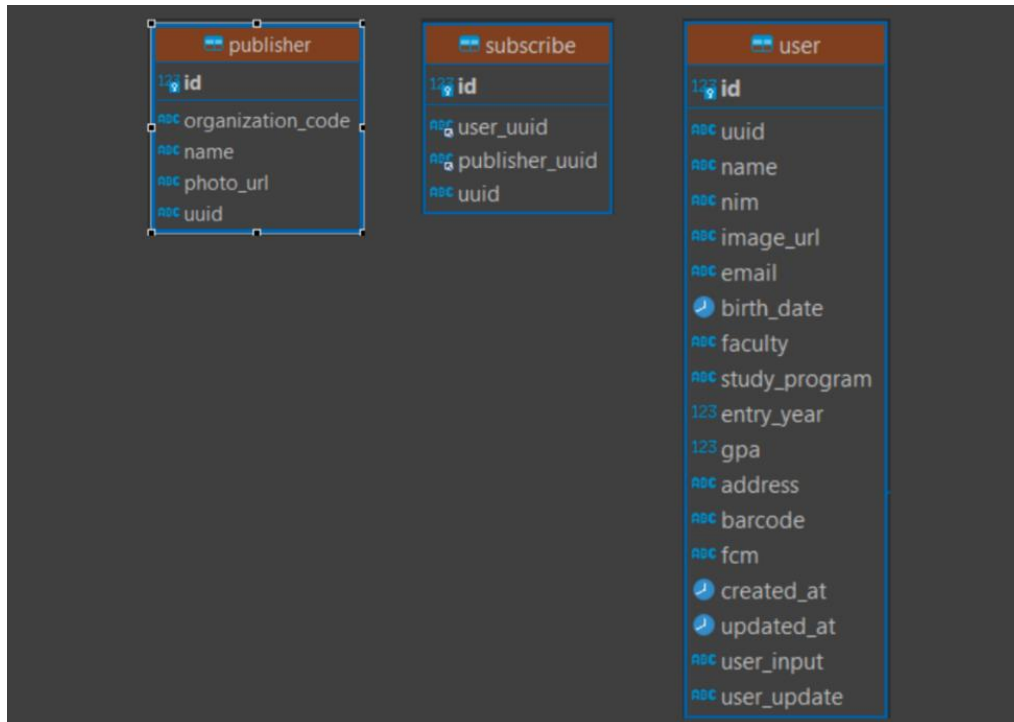
Gambar 3.16 Halaman What's on UII



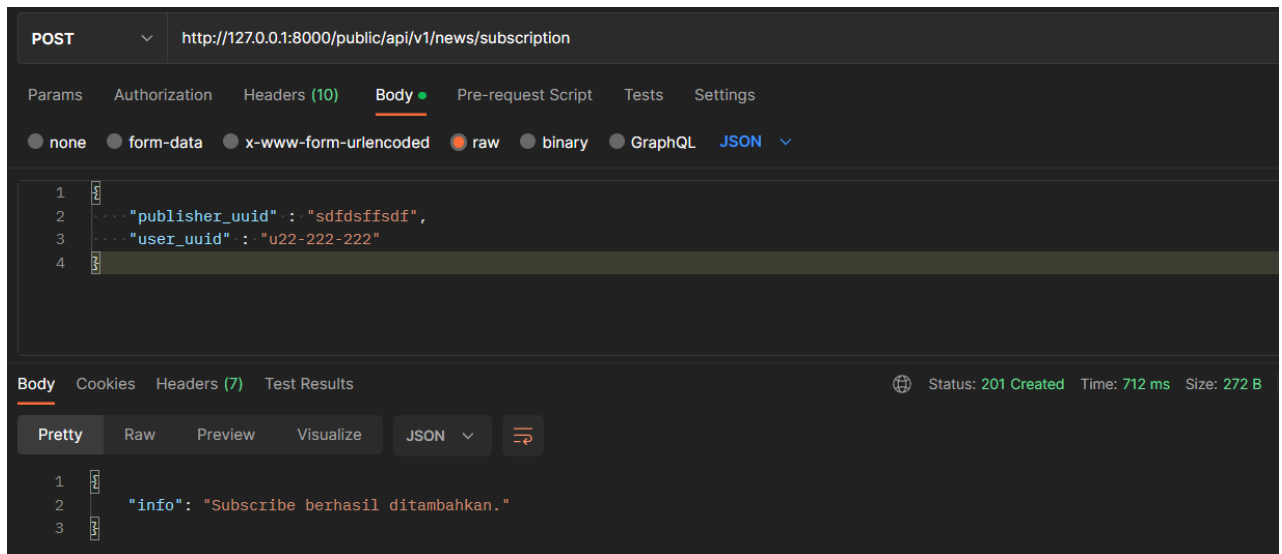
Gambar 3.17 Halaman detail berita

3.3.2 Berlangganan Penerbit

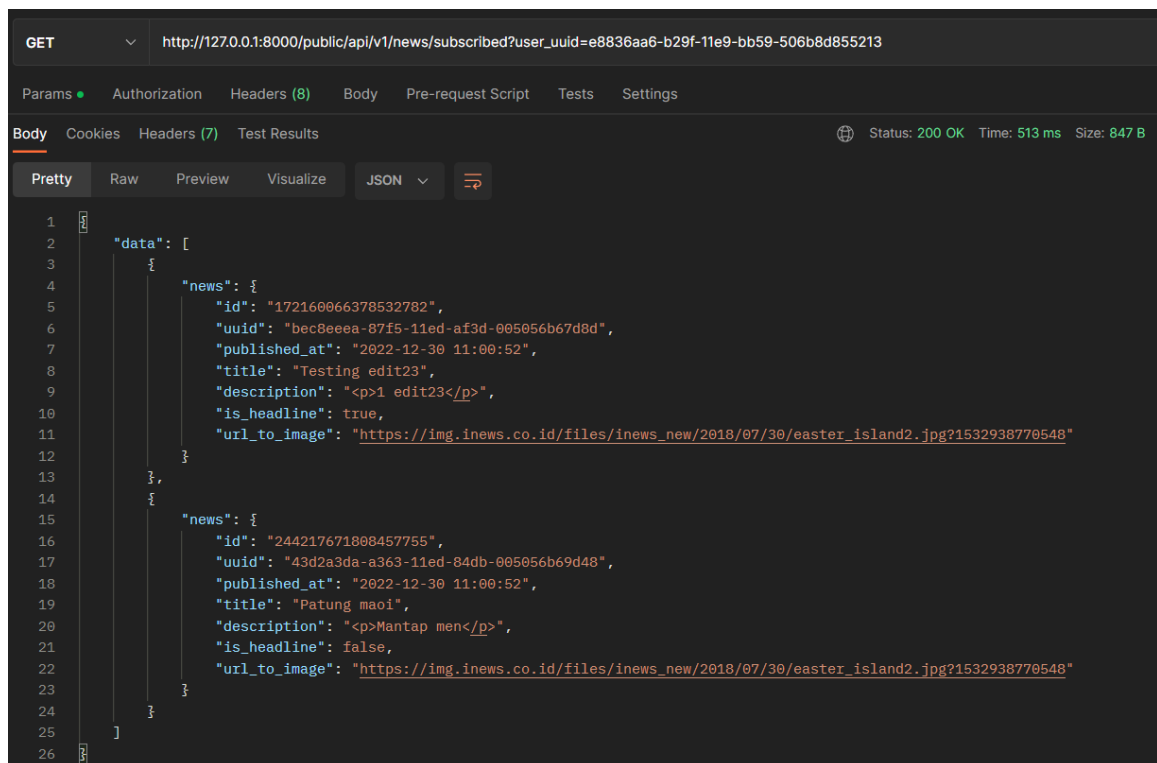
Selain menampilkan berita, pengguna juga dapat berlangganan terhadap penerbit yang diinginkan. Setelah itu, pengguna dapat mengakses postingan berita dari tiap penerbit yang dilanggan pada *tab* Untuk Anda. Untuk merealisasikan fitur ini juga memerlukan basis data. Dalam melakukan proses berlangganan, pengguna akan mengirimkan data ke dalam basis data *subscribe*. Tabel basis data *subscribe* memiliki tiga properti yang diantaranya merupakan UUID dari pengguna dan penerbit seperti yang terlihat pada Gambar 3.18. Tabel basis data *subscribe* ini nantinya akan digunakan untuk menampilkan berita berdasarkan status langganan pada API berita, dan berdasarkan penerbit berita. Adapun API yang digunakan pada fitur ini terdapat tiga macam *request*, yaitu GET untuk menampilkan berita, POST untuk melakukan input data langganan, dan DELETE yang digunakan ketika berhenti berlangganan.

Gambar 3.18 Basis data *subscribe*

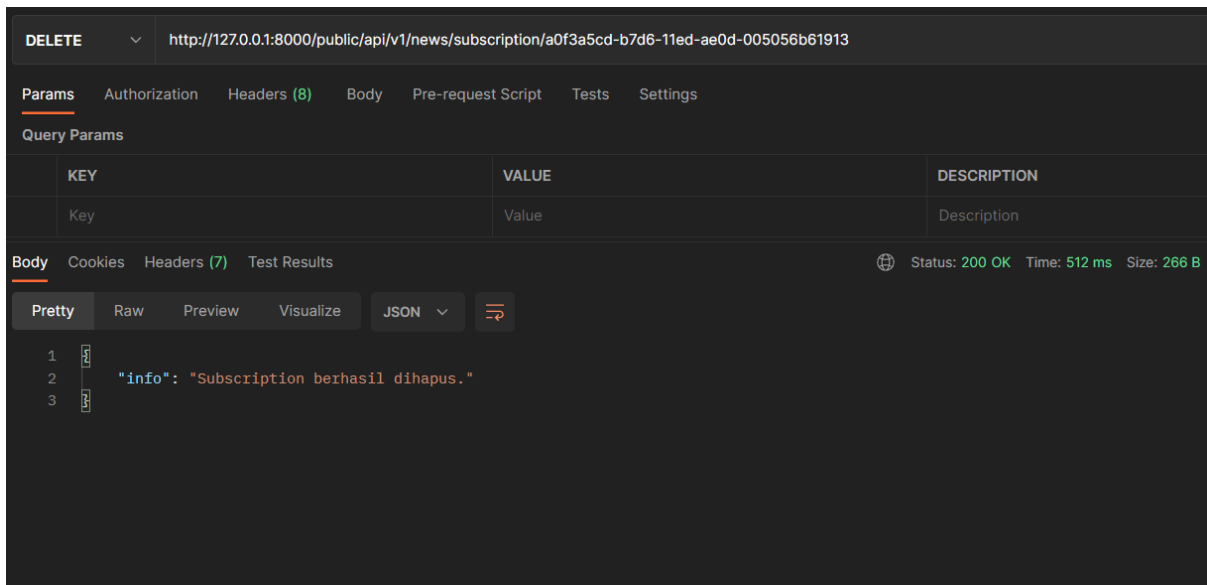
Pada Gambar 3.19, kita dapat melihat API untuk melakukan *request* POST yang mana melakukan input atau memasukkan data UUID dari penerbit dan juga pengguna. API ini diimplementasikan ketika menekan tombol berlangganan pada halaman detail berita. Pada Gambar 3.20, kita dapat melihat API dengan *request* GET yang mana menampilkan daftar berita sesuai dengan penerbit yang dilanggan oleh pengguna. API GET memerlukan sebuah *query parameter* yang berupa UUID pengguna. Pada Gambar 3.21, kita dapat melihat sebuah API dengan jenis *request* DELETE. API ini diimplementasikan ketika pengguna membatalkan langganan terhadap penerbit. Jika kita melihat kembali pada Gambar 3.13, kita dapat melihat sebuah objek dengan *key* bernama *user_info_on_news*. Di dalam objek ini terdapat informasi tentang pengguna terhadap berita, apakah pengguna sudah berlangganan terhadap penerbit berita atau belum, jika sudah maka properti *is_subscribe* akan bernilai *true*, dan akan ada properti yang memuat UUID *subscribe*. UUID inilah yang akan digunakan pada URL *path* API dengan *method* DELETE untuk berhenti berlangganan pada penerbit berita.



Gambar 3.19 API untuk melakukan subscribe atau berlangganan



Gambar 3.20 API untuk menampilkan berita dari publsiher langganan



Gambar 3.21 API untuk berhenti berlangganan

Untuk penerapannya pada Flutter, penulis masih menggunakan *package* yang sama yaitu Dio. Kali ini *method* yang digunakan adalah `dio.get`, `dio.post`, dan `dio.delete`. Pada Gambar 3.22 kita bisa melihat kode Dart pada Flutter untuk penerapan `dio.get` dalam melakukan *request* untuk mendapatkan berita dari penerbit yang dilanggan oleh pengguna. Sama seperti sebelumnya, untuk menampilkan berita dari penerbit yang telah dilanggan oleh pengguna juga membutuhkan UUID pengguna sebagai *query parameter*. Namun, model yang digunakan kali ini berbeda dengan yang digunakan ketika menampilkan seluruh berita terbaru.

Gambar 3.23 menampilkan kode untuk mengimplementasi API dengan *method* POST menggunakan `dio.post`. Penggunaannya hampir sama dengan `dio.get`, hanya saja menambahkan properti `data`. Properti `data` berisi nilai yang akan dimasukkan ke basis data, dalam hal ini adalah UUID dari penerbit dan UUID dari pengguna. Hal ini digunakan ketika pengguna mulai berlangganan terhadap penerbit. Dan Gambar 3.24 merupakan kode dari pengimplementasian API dengan *method* DELETE menggunakan `dio.delete`. Ini digunakan untuk menghapus data dari basis data ketika pengguna berhenti berlangganan terhadap penerbit. Fitur berlangganan penerbit dapat dijalankan ketika pengguna menekan tombol langganan yang dapat ditemukan di halaman beranda seperti pada Gambar 3.25.

```

Future<NewsSubscribe> getSubscribeNews(BuildContext context) async {
  userUuid = await Utils.getString('user_uuid');
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news/subscribed";
      final response = await dio.get(
        path,
        options: Options(
          headers: {
            "x-app": "isi",
            "x-menu": "isi"
          }
        ),
        queryParameters: {
          "user_uuid" : userUuid
        }
      );

      if(response.statusCode == 200) {
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        // return News.fromJson(json.decode(resData));
        return NewsSubscribe.fromJson(resData);
      }
      else {
        print('${response.statusCode} : ${response.data.toString()}');
        throw response.statusCode!;
      }
    }
    .....
  }
}

```

Gambar 3.22 Penerapan dio.get untuk fitur *subscribe*

```

Future<NewsDetail> postSubscribeNews(BuildContext context, String penerbitUuid,
String userUuid) async {
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news/subscription/";
      final response = await dio.post(
        path,
        options: Options(
          headers: {
            "x-app": "tes",
            "x-menu": "tes"
          }
        ),
        data: {
          'penerbit_uuid' : penerbitUuid,
          'user_uuid' : userUuid
        }
      );
      if(response.statusCode == 200) {
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        // return NewsDetail.fromJson(json.decode(resData));
        return NewsDetail.fromJson(resData);
      }
      else {
        print('${response.statusCode} : ${response.data.toString()}');
        throw response.statusCode!;
      }
    }
    .....
  }
}

```

Gambar 3.23 Penerapan dio.post untuk fitur *subscribe*

```

Future<NewsDetail> deleteSubscribeNews(BuildContext context, String
subscribeUuid) async {
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news/subscription/$subscribeUuid";
      final response = await dio.delete(
        path,
        options: Options(
          headers: {
            "x-app": "tes",
            "x-menu": "tes"
          }
        )
      );

      if(response.statusCode == 200) {
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        // return NewsDetail.fromJson(json.decode(resData));
        return NewsDetail.fromJson(resData);
        .....
      }
    }
  }
}

```

Gambar 3.24 Penerapan dio.delete untuk fitur subscribe



Gambar 3.25 Tombol langganan pada halaman detail berita

3.3.3 Tambah ke Favorit

What's on UII juga memiliki fitur untuk menambahkan berita ke favorit. Dengan adanya fitur ini, pengguna bisa menyimpan berita ataupun informasi penting sesuai yang dikehendaki. Fitur ini juga menyimpan data ke dalam basis data. Ketika pengguna menambahkan berita ke daftar favorit, pengguna akan mengirimkan atau menginputkan data ke dalam basis data. Beberapa tabel basis data yang dibutuhkan untuk merealisasikan fitur ini dapat dilihat pada Gambar 3.26.

Dapat kita lihat tabel utama untuk menyimpan data berita favorit adalah tabel dengan nama `favorit_news`. Tabel `favorit_news` membutuhkan properti dari kedua tabel lain yaitu tabel `user` dan tabel `news`. Properti yang dibutuhkan adalah UUID dari pengguna dan juga UUID dari berita. Data tersebut akan dimasukkan ke dalam tabel `favorit_news` ketika pengguna menambahkan berita ke daftar favorit.

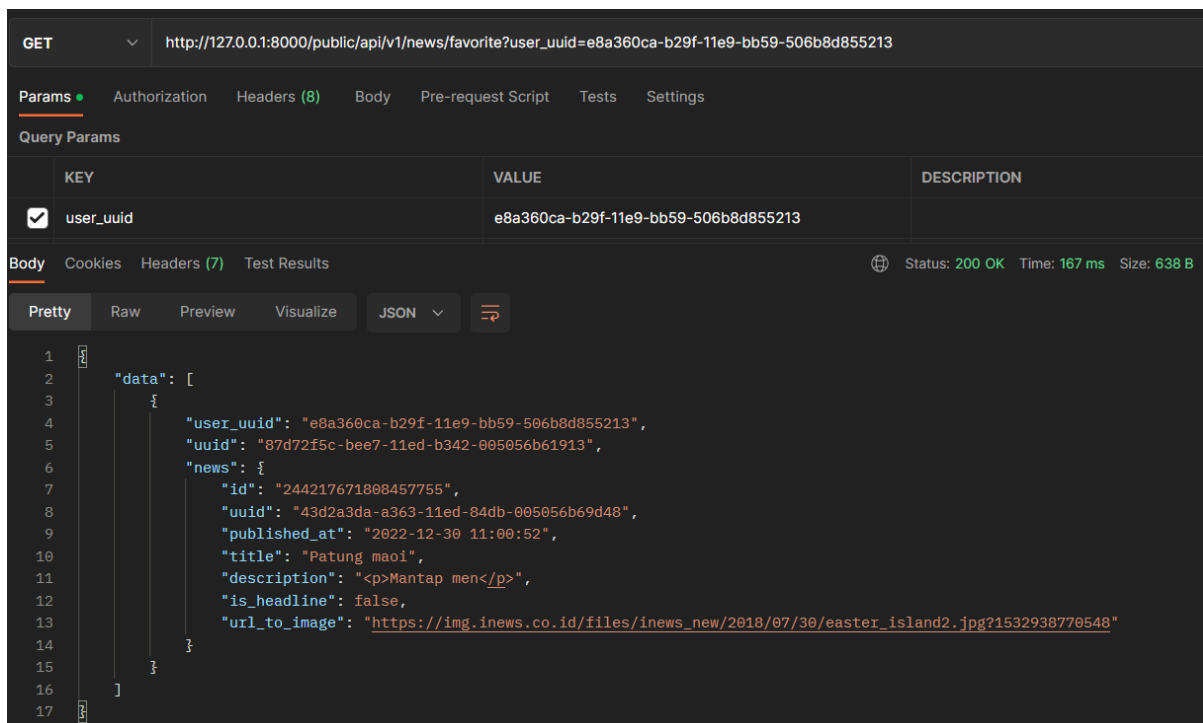
API yang digunakan pada fitur ini juga memiliki tiga macam *request*, sama seperti fitur untuk berlangganan. *Request* yang digunakan adalah GET untuk menampilkan berita favorit, POST untuk menambahkan berita ke favorit, dan DELETE untuk menghapus berita dari daftar favorit.

news	favorite_news	user
123 id	123 id	123 id
ABC publisher_uuid	ABC user_uuid	ABC uuid
ABC title	ABC news_uuid	ABC name
ABC description	ABC uuid	ABC nim
ABC url_to_image		ABC image_url
ABC published_at		ABC email
123 is_headline		ABC birth_date
ABC uuid		ABC faculty
ABC created_at		ABC study_program
ABC user_input		123 entry_year
ABC updated_at		123 gpa
ABC user_update		ABC address
123 row_id		ABC barcode
		ABC fcm
		ABC created_at
		ABC updated_at
		ABC user_input
		ABC user_update

Gambar 3.26 Tabel basis data untuk fitur tambah favorit

Gambar 3.27 menunjukkan API dengan jenis *request* GET yang digunakan untuk menampilkan berita favorit. API GET memerlukan sebuah *query parameter* yang berupa UUID pengguna. Gambar 3.28 menunjukkan API yang digunakan untuk menambahkan berita ke favorit dengan jenis *request* POST. Data yang dimasukkan ke API ini adalah UUID dari pengguna dan UUID dari berita. Ketika *request* berhasil, terdapat respon yang menunjukkan

bahwa berita berhasil ditambahkan ke favorit. Pada Gambar 3.29, kita dapat melihat sebuah API dengan jenis *request* DELETE. API ini diimplementasikan ketika pengguna menghapus berita dari daftar favorit. Sama seperti pada fitur subscribe, jika kita melihat kembali pada Gambar 3.13, kita dapat melihat sebuah objek dengan *key* bernama *user_info_on_news*. Di dalam objek ini terdapat informasi tentang pengguna terhadap berita, apakah pengguna sudah menambahkan berita tersebut ke daftar favorit, jika sudah maka properti *is_favorite* akan bernilai *true*, dan akan ada properti yang memuat UUID favorit. UUID inilah yang akan digunakan pada URL *path* API dengan *method* DELETE untuk menghapus berita dari daftar favorit.

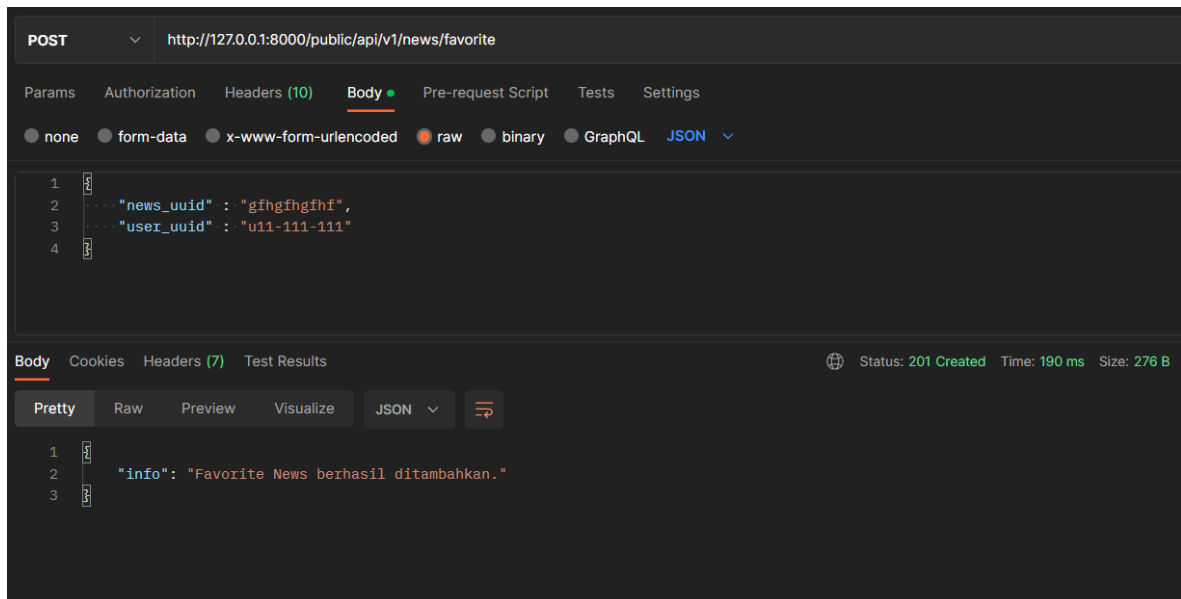


```
GET http://127.0.0.1:8000/public/api/v1/news/favorite?user_uuid=e8a360ca-b29f-11e9-bb59-506b8d855213

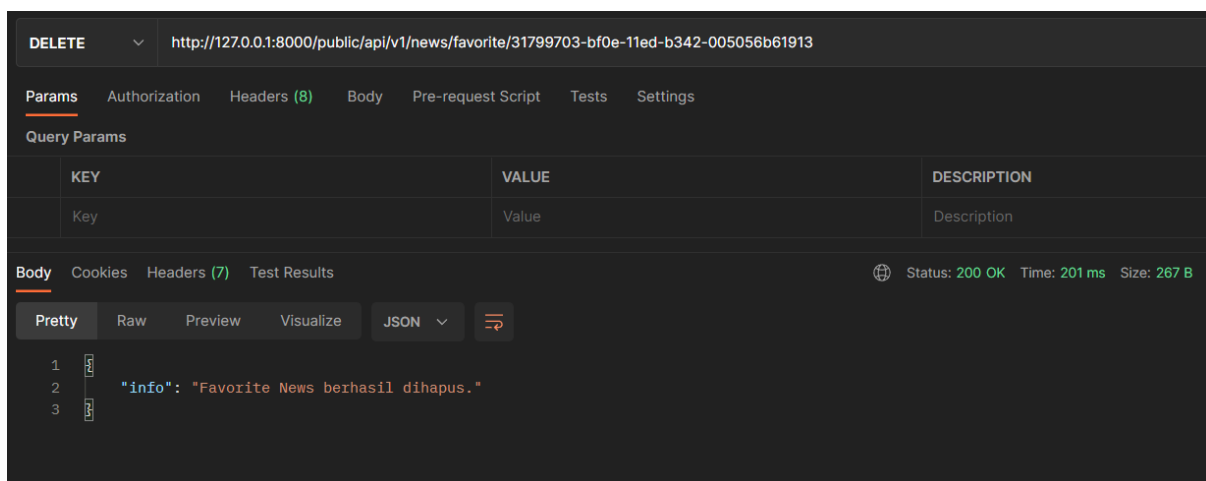
Params • Authorization Headers (8) Body Pre-request Script Tests Settings
Query Params
KEY VALUE DESCRIPTION
[x] user_uuid e8a360ca-b29f-11e9-bb59-506b8d855213

Body Cookies Headers (7) Test Results Status: 200 OK Time: 167 ms Size: 638 B
Pretty Raw Preview Visualize JSON ↕
1 2
2 "data": [
3   {
4     "user_uuid": "e8a360ca-b29f-11e9-bb59-506b8d855213",
5     "uuid": "87d72f5c-bee7-11ed-b342-005056b61913",
6     "news": {
7       "id": "244217671808457755",
8       "uuid": "43d2a3da-a363-11ed-84db-005056b69d48",
9       "published_at": "2022-12-30 11:00:52",
10      "title": "Patung maoi",
11      "description": "<p>Mantap men</p>",
12      "is_headline": false,
13      "url_to_image": "https://img.inews.co.id/files/inews_new/2018/07/30/easter_island2.jpg?1532938770548"
14    }
15  }
16 ]
17
```

Gambar 3.27 API untuk menampilkan berita favorit



Gambar 3.28 API untuk menambahkan berita ke favorit



Gambar 3.29 API untuk menghapus berita dari favorit

Untuk penerapannya pada Flutter, penulis masih menggunakan *package* yang sama yaitu Dio. Kali ini *method* yang digunakan adalah `dio.get`, `dio.post`, dan `dio.delete`. Pada Gambar 3.30 kita bisa melihat kode Dart pada Flutter untuk penerapan `dio.get` dalam melakukan *request* untuk mendapatkan daftar berita favorit. Sama seperti sebelumnya, untuk menampilkan berita favorit membutuhkan UUID pengguna sebagai *query parameter*. Model yang digunakan juga kali ini berbeda dengan model untuk berita terbaru dan berita dari penerbit langganan.

Gambar 3.31 menampilkan kode untuk melakukan implementasi API dengan *method* POST menggunakan `dio.post`. Penggunaannya hampir sama dengan `dio.get`, hanya saja menambahkan properti data. Properti data berisi nilai yang akan dimasukkan ke basis data,

dalam hal ini adalah UUID dari berita dan UUID dari pengguna. Hal ini digunakan ketika pengguna menambahkan berita ke daftar favorit. Dan Gambar 3.32 merupakan kode dari pengimplementasian API dengan *method* DELETE menggunakan `dio.delete`. Ini digunakan untuk menghapus data dari basis data ketika pengguna menghapus berita dari daftar berita favorit.

Untuk menambahkan berita ke favorit, pengguna dapat menekan tombol yang ada pada banner gambar dari berita di halaman detail berita seperti yang ditampilkan pada Gambar 3.33. Untuk mengakses daftar berita favorit, pengguna dapat menggunakan *tab* berita favorit pada halaman What's on UII.

```
Future<NewsFavorite> getFavoriteNews(BuildContext context) async {
  userUuid = await Utils.getString('user_uuid');
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news/favorite";
      final response = await dio.get(
        path,
        options: Options(
          headers: {
            "x-app": "isi",
            "x-menu": "isi"
          }
        ),
        queryParameters: {
          "user_uuid" : userUuid
        }
      );

      if(response.statusCode == 200) {
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        // return News.fromJson(json.decode(resData));
        return NewsFavorite.fromJson(resData);
      }
      .....
    }
  }
}
```

Gambar 3.30 penerapan `dio.get` untuk fitur tambah favorit

```

Future<News> postFavoriteNews(BuildContext context, String newsUuid, String
userUuid) async {
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news/favorite";
      final response = await dio.post(
        path,
        options: Options(
          headers: {
            "x-app": "tes",
            "x-menu": "tes"
          }
        ),
        data: {
          'news_uuid' : newsUuid,
          'user_uuid' : userUuid
        }
      );

      if(response.statusCode == 200) {
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        // return NewsDetail.fromJson(json.decode(resData));
        return News.fromJson(resData);
      }
    }
    .....
  }
}

```

Gambar 3.31 Penerapan dio.post untuk fitur tambah favorit

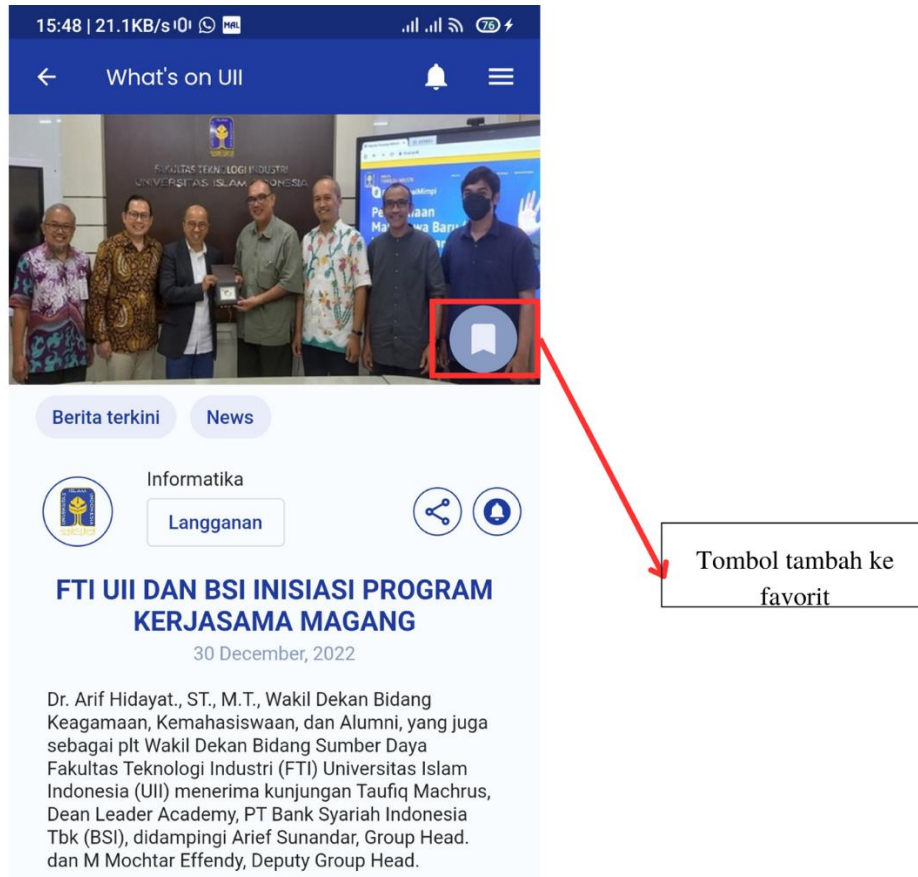
```

Future<NewsDetail> deleteFavoriteNews(BuildContext context, String
favoriteUuid) async {
  if (await Utils.checkConnection()) {
    try {
      String path = "$url/public/api/v1/news/favorite/$favoriteUuid";
      final response = await dio.delete(
        path,
        options: Options(
          headers: {
            "x-app": "tes",
            "x-menu": "tes"
          }
        )
      );

      if(response.statusCode == 200) {
        var resData = response.data;
        onApiSuccess(resData, response.statusCode);
        // return NewsDetail.fromJson(json.decode(resData));
        return NewsDetail.fromJson(resData);
      }
    }
    .....
  }
}

```

Gambar 3.32 penerapan dio.delete untuk fitur tambah favorit



Gambar 3.33 Tombol tambah ke favorit pada halaman detail berita

3.3.4 Notifikasi Pengingat Membaca Berita

Pada fitur ini, pengguna dapat mengatur pengingat untuk membaca berita. Pengguna juga dapat menyalakan notifikasi pengingat sebelum waktu yang ditentukan. Untuk merealisasikan fitur ini, tentu saja penulis harus membuat sebuah mekanisme notifikasi yang terjadwal. Dalam hal ini, penulis menggunakan sebuah *package* dari Flutter yang bernama Flutter Local Notification. Flutter Local Notification adalah fitur yang tersedia dalam *framework* Flutter yang memungkinkan aplikasi untuk menampilkan notifikasi di perangkat. Notifikasi dapat ditampilkan di layar perangkat atau di area notifikasi perangkat, tergantung pada pengaturan yang telah diberikan oleh pengguna.

Untuk menggunakan Flutter Local Notification, pertama-tama kita perlu menambahkan *package* flutter_local_notifications ke dalam *file* pubspec.yaml proyek Flutter. Setelah itu, kita dapat membuat notifikasi dengan menggunakan fungsi-fungsi yang tersedia di *package* tersebut.

Flutter Local Notification juga menyediakan fitur-fitur lain seperti penjadwalan notifikasi, pengulangan notifikasi, dan menampilkan notifikasi dengan format yang lebih kompleks

seperti notifikasi dengan gambar atau tautan. Dengan demikian, Flutter Local Notification dapat membantu penulis untuk menampilkan notifikasi di aplikasi Flutter dengan mudah.

Dalam penerapannya ke fitur pengingat untuk membaca berita, hal pertama yang dilakukan penulis adalah membuat sebuah kelas untuk melakukan konfigurasi atau pengkodean untuk menerapkan *package* Flutter Local Notification. Pada kelas ini, kita dapat menginisialisasi pengaturan notifikasi untuk Android dan iOS. Kita juga dapat mengatur bagaimana detail notifikasi yang nanti akan muncul. Kita juga dapat menambahkan fungsi untuk opsi apakah aplikasi akan muncul secara instan dengan menggunakan *trigger* atau notifikasi akan muncul secara terjadwal. Penerapan *package* Flutter Local Notification pada modul What's on UII dapat dilihat pada Gambar 3.34.

Gambar 3.34 menunjukkan kelas `LocalNotificationService` yang dibuat oleh penulis. Kelas ini dibuat pada *file* `local_notification.dart`. Kelas ini nantinya akan diinstansiasi pada kelas `ReminderDialog` yang merupakan lokasi dimana fitur reminder ini diatur oleh pengguna.

```
class LocalNotificationService {
  // LocalNotificationService();

  final _localNotificationService = FlutterLocalNotificationsPlugin();
  final BehaviorSubject<String?> onNotificationClick = BehaviorSubject();

  Future<void> initialize() async {
    tz.initializeTimeZones();
    const AndroidInitializationSettings androidInitializationSettings =
      AndroidInitializationSettings('@drawable/uii_icon_white');

    DarwinInitializationSettings iosInitializationSettings =
    const DarwinInitializationSettings(
      requestAlertPermission: true,
      requestBadgePermission: true,
      requestSoundPermission: true,
    );

    final InitializationSettings settings = InitializationSettings(
      android: androidInitializationSettings,
      iOS: iosInitializationSettings,
    );

    //when app is closed
    final details = await
    _localNotificationService.getNotificationAppLaunchDetails();
    if (details != null && details.didNotificationLaunchApp){
      onNotificationClick.add(details.notificationResponse?.payload);
    }
    .....
  }
}
```

Gambar 3.34 Penerapan *package* Flutter Local Notification

Seperti yang sudah dijelaskan di atas, kelas ini kemudian akan diinstansiasi ke dalam kelas `ReminderDialog` menjadi suatu objek. Objek ini nantinya dapat penulis gunakan untuk memanggil fungsi yang ada pada kelas `LocalNotificationService` untuk memunculkan

notifikasi. Kode yang penulis buat untuk memunculkan notifikasi pada fitur ini dapat dilihat pada Gambar 3.35.

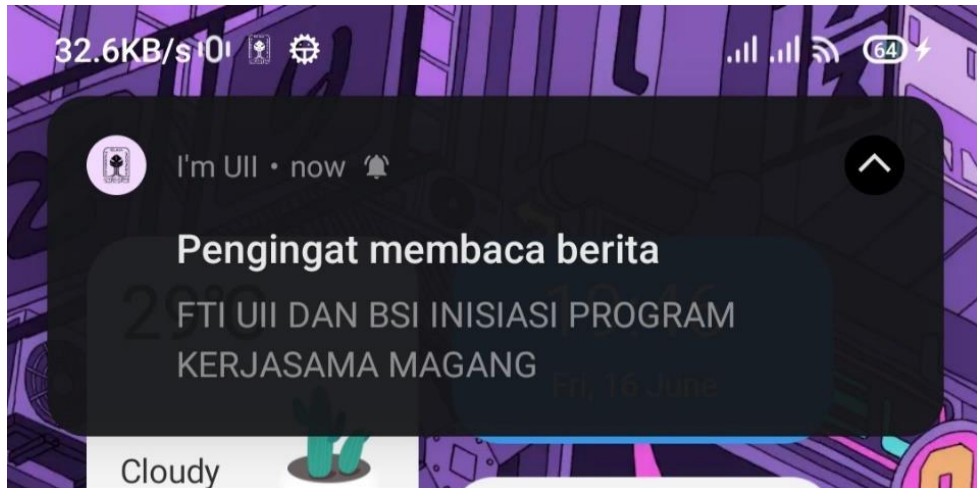
Seperti yang terlihat pada Gambar 3.35, terdapat beberapa kondisi untuk memunculkan notifikasi. Hal ini sesuai dengan fitur yang diharapkan. Beberapa kondisi tersebut dibedakan sesuai dengan waktu pengingat yang diinginkan pengguna. Seperti tepat pada waktu yang ditentukan, 10 menit sebelum waktu yang ditentukan, 20 menit, dan seterusnya. Adapun hasil dari penerapan fitur ini dapat dilihat pada Gambar 3.36 yang menunjukkan notifikasi mengambang dari aplikasi. Gambar 3.37 menunjukkan notifikasi pada area notifikasi (*notification shade*) di Android.

```

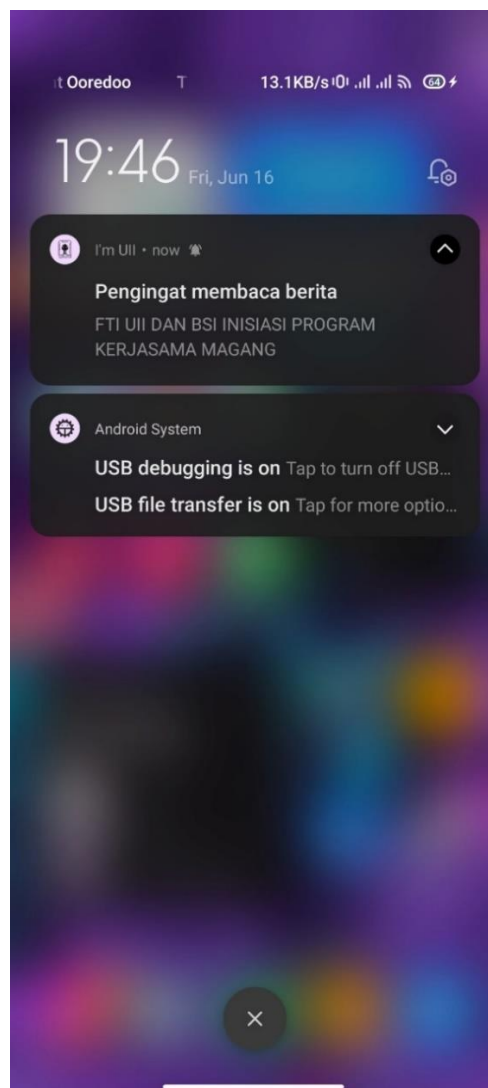
if(_selectedTime == "10 menit sebelumnya"){
    DateTime _reminderDate =
        _nowDate.subtract(const Duration(minutes: 10));
    await service.showScheduledNotification(
        id: widget.newsId, title:
        'Pengingat membaca berita', body: widget.newsTitle, dateTime:
        _reminderDate);
}
else if(_selectedTime == "30 menit sebelumnya"){
    DateTime _reminderDate = _nowDate.subtract(const Duration(minutes: 30));
    await service.showScheduledNotification(
        id: widget.newsId, title: 'Pengingat membaca berita', body:
        widget.newsTitle, dateTime: _reminderDate);
}
else if(_selectedTime == "1 jam sebelumnya"){
    DateTime _reminderDate = _nowDate.subtract(const Duration(hours: 1));
    await service.showScheduledNotification(
        id: widget.newsId, title: 'Pengingat membaca berita', body:
        widget.newsTitle, dateTime: _reminderDate);
}
else if(_selectedTime == "2 jam sebelumnya"){
    DateTime _reminderDate = _nowDate.subtract(const Duration(hours: 2));
    await service.showScheduledNotification(
        id: widget.newsId, title: 'Pengingat membaca berita', body:
        widget.newsTitle, dateTime: _reminderDate);
}
else {
    await service.showScheduledNotification(id: widget.newsId, title:
    'Pengingat membaca berita', body: widget.newsTitle, dateTime: _nowDate);
}

```

Gambar 3.35 Kode untuk memunculkan notifikasi



Gambar 3.36 Notifikasi mengambang dari berita di modul What's on UII



Gambar 3.37 Tampak notifikasi pada *notification shade* Android

3.4 Pengujian

Tahap pengujian atau *testing* pada adalah tahap di mana sistem yang telah dikembangkan diperiksa untuk memastikan bahwa semua persyaratan dan fungsionalitas telah terpenuhi dengan baik. Pengujian meliputi identifikasi kasus uji, pelaksanaan uji coba, dan verifikasi apakah sistem berperilaku sesuai dengan yang diharapkan. Pada tahapan ini metode pengujian yang digunakan adalah Black Box Testing, yang bertujuan untuk menguji fungsionalitas perangkat lunak yang telah dikembangkan, dilakukan pemeriksaan apakah sistem berperilaku sesuai dengan spesifikasi kebutuhan atau tidak.

Pengujian ini dilakukan secara internal di dalam tim. Dengan responden adalah seluruh anggota tim pengembang yang berjumlah lima orang. Pengujian dilakukan dengan mencoba langsung aplikasi, kemudian melakukan aksi sesuai dengan skenario yang telah dibuat. Setelah itu, penguji melihat respon dari aplikasi apakah aplikasi dapat berjalan sesuai dengan skenario atau belum. Skenario beserta hasil uji disajikan ke dalam sebuah tabel. Tabel tersebut dapat dilihat pada Tabel 3.3. Pada Tabel 3.3, terdapat kolom *usecase* yang menggambarkan fungsional aplikasi. Seperti UC.1 yang merupakan fungsional melihat daftar berita. UC.1 memiliki dua skenario, yang pertama adalah pengguna dapat membuka halaman What's on UII, kemudian yang kedua pengguna dapat melihat daftar berita. Terdapat juga kolom hasil yang diharapkan ketika pengguna atau responden menjalankan skenario. Setelah itu, hasil pengujian dipaparkan ke dalam kolom hasil pengujian.

Berdasarkan data yang tercantum dalam Tabel 3.3, dapat disimpulkan bahwa seluruh fungsionalitas pada fitur What's on UII yang terdapat dalam aplikasi I'm UII berfungsi dengan baik dan sesuai dengan harapan. Hasil ini menunjukkan bahwa pengembangan fitur What's on UII dalam aplikasi I'm UII telah mencapai tujuan yang telah ditetapkan dan dapat memberikan pengalaman positif bagi pengguna. Temuan ini juga memberikan keyakinan bahwa fitur What's on UII dalam aplikasi I'm UII memberikan manfaat yang signifikan bagi pengguna dalam mendapatkan akses mudah dan informasi yang relevan dari berbagai sumber tepercaya di UII.

Tabel 3.3 Tabel hasil pengujian Black Box

<i>Usecase</i>	Skenario	Hasil yang diharapkan	Hasil pengujian
UC.1 Melihat daftar berita	Membuka halaman What's on UII	Pengguna masuk dan dapat melihat halaman What's on UII.	Berhasil
	Melihat daftar berita	Di halaman What's on UII, pengguna dapat melihat daftar berita yang tersedia	Berhasil
UC.2 Melihat detail berita	Membuka halaman detail berita	Pengguna dapat masuk ke halaman detail berita dan melihat isi keseluruhan atau detail dari berita	Berhasil
UC.3 Berlangganan penerbit	Berlangganan penerbit berita	Setelah menekan tombol berlangganan, pengguna dapat melihat berita-berita dari penerbit langganan di daftar berita untuk anda	Berhasil
	Berhenti berlangganan	Setelah menekan tombol berhenti berlangganan, berita dari penerbit tersebut tidak lagi muncul di daftar berita untuk anda	Berhasil
UC.4 Menyalakan pengingat	Mengaktifkan pengingat membaca berita	Pengguna dapat menerima notifikasi pengingat membaca berita sesuai dengan waktu yang ditentukan	Berhasil
UC.5 Bagikan berita	Membagikan berita dengan tautan	Berita dapat diakses melalui tautan yang didapatkan ketika menekan tombol bagikan	Berhasil
UC.6 Tambah berita ke favorit	Menambahkan berita ke favorit	Setelah menekan tombol tambah ke favorit, pengguna dapat melihat berita tersebut di daftar berita favorit.	Berhasil
	Menghapus berita dari favorit	Setelah menekan tombol untuk menghapus berita dari favorit, berita tidak lagi ditampilkan di daftar berita favorit	Berhasil

3.5 Perbandingan dengan Studi dan Aplikasi Terdahulu

Berdasarkan hasil dari pengujian, seluruh fungsionalitas pada What's on UII telah berhasil dikembangkan. Merujuk pada studi yang telah dibahas pada kajian pustaka, penulis

membandingkan fitur antara What's on UII dan aplikasi yang dikembangkan pada studi terdahulu, yaitu Berita Labuhanbatu dan Eresha Mobile. Selain itu, dikarenakan saat ini banyak aplikasi berita yang tersedia dan dapat diunduh melalui berbagai platform seperti Google Play Store dan Apple App Store. Penulis juga menambahkan beberapa aplikasi terdahulu sebagai perbandingan. Beberapa aplikasi tersebut diantaranya adalah kompas.com dan IDN News. Aplikasi-aplikasi yang digunakan sebagai pembanding juga menawarkan fitur untuk mengakses berita dan informasi. Namun, fitur What's on UII pada aplikasi I'm UII memiliki keunggulan pada beberapa fitur.

Tabel 3.4 menunjukkan perbandingan antara What's on UII dan beberapa aplikasi yang telah disebutkan. Dari Tabel 3.4, dapat terlihat bahwa fitur What's on UII pada aplikasi I'm UII memiliki beberapa keunggulan yang membedakannya dari aplikasi berita lainnya. Salah satu keunggulannya adalah adanya fitur pengingat membaca berita, yang memungkinkan pengguna untuk menandai berita atau informasi yang ingin dibaca namun tidak sempat saat itu, dan mengatur pengingat untuk membacanya di waktu yang lebih sesuai. Fitur ini membantu pengguna agar tidak lupa dan dapat mengatur waktu mereka dengan lebih efisien. Selain itu, What's on UII juga menyediakan fitur berlangganan penerbit berita, di mana terdapat beberapa penerbit berita seperti fakultas, jurusan, dan organisasi lain yang ada di UII. Dengan fitur berlangganan ini, pengguna dapat memilih untuk mendapatkan pembaruan berita terkini dari penerbit-penerbit yang mereka minati, sehingga mereka dapat memperoleh informasi yang lebih relevan dan sesuai dengan minat mereka di lingkungan UII.

Tabel 3.4 Perbedaan antar aplikasi

Fitur	What's on UII	Berita Labuhanbatu	Eresha Mobile	Kompas.com	IDN News
Menampilkan berita dalam <i>list</i>	Ya	Ya	Ya	Ya	Ya
Notifikasi	Ya	Tidak	Ya	Ya	Ya
Menyimpan berita (<i>favorit/bookmark</i>)	Ya	Tidak	Tidak	Ya	Tidak
Berlangganan penerbit	Ya	Tidak	Tidak	Tidak	Tidak
Membagikan berita	Ya	Tidak	Tidak	Ya	Ya
Mengatur pengingat membaca berita	Ya	Tidak	Tidak	Tidak	Tidak

BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Selama magang, penulis menemukan beberapa perbedaan antara pelaksanaan di lapangan dengan teori yang telah dipelajari di perkuliahan. Beberapa perbedaan yang ditemukan antara lain adalah penggunaan *tools* untuk mengembangkan aplikasi *mobile*, interaksi antara aplikasi *back-end* dan *front-end mobile*, dan perbedaan dalam metode pengembangan. Bab ini akan membahas relevansi akademik melalui perbandingan antara teori yang telah dipelajari dan implementasi di lapangan selama melaksanakan magang.

4.1.1 *Tools Pengembangan Aplikasi Mobile*

Selama melaksanakan magang, penulis mengembangkan What's on UII dengan menggunakan sebuah *framework* multiplatform yaitu Flutter. Penggunaan Flutter sebagai *framework* multiplatform memberikan keleluasaan dalam mengembangkan aplikasi *mobile* yang dapat berjalan di berbagai sistem operasi. Hal ini dikarenakan dengan menggunakan Flutter, pengembang dapat mengembangkan aplikasi yang dapat berjalan di platform Android dan iOS sekaligus hanya dengan satu basis kode saja (Katti & K, 2022).

Penggunaan *framework* multiplatform selama magang, berbeda dengan teori yang diajarkan di perkuliahan. Teori yang dipelajari untuk pengembangan aplikasi *mobile* selama perkuliahan adalah pengembangan aplikasi dengan menggunakan teknologi *native*. Teknologi *native* yang digunakan pun sebatas *native* Android saja dengan bahasa pemrograman Kotlin.

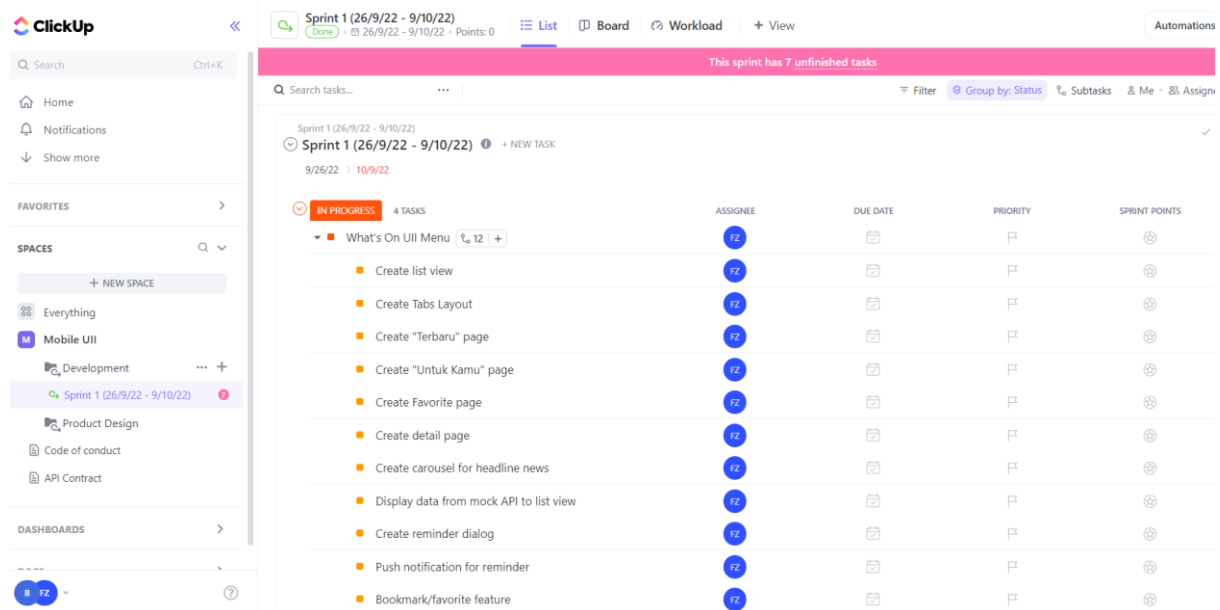
BSI UII memilih Flutter karena Flutter dinilai dapat mempercepat proses pengembangan aplikasi (Swarna, Purnama, & Anthony, 2020). Saat ini, Flutter juga mulai banyak diminati oleh beberapa perusahaan besar seperti Gojek, Nubank, hingga Alibaba Group. Hal ini dikarenakan Flutter menawarkan beberapa keunggulan seperti pengembangan aplikasi lebih cepat, dokumentasi yang lengkap, pembuatan UI yang menarik, tidak memiliki ketergantungan antar *library*, hingga kapabilitas testing yang mumpuni (Ilham, 2023).

4.1.2 *Metode Pengembangan*

Seperti yang sudah dijelaskan pada bab pertama, penulis ditempatkan ke dalam sebuah tim percepatan yang berada di luar tim utama BSI UII. Metode pengembangan yang umum digunakan di BSI UII adalah Agile dengan menggunakan *framework* Scrum. Pada awalnya,

tim percepatan yang ditempati penulis memulai pengerjaan proyek dengan Scrum. Pengerjaan proyek dimulai dengan *sprint planning*, kemudian penulis diberikan tugas yang terdokumentasi pada platform ClickUp seperti yang ditampilkan pada Gambar 4.1.

Gambar 4.1 menunjukkan tugas-tugas atau *task* yang harus penulis kerjakan dalam satu *sprint* dengan rentang waktu dua minggu. Dokumentasi ini dibuat sesuai dengan tugas yang diberikan ketika melakukan *sprint planning*. Penulis harus memberikan tanda ketika telah menyelesaikan setiap tugas.



Gambar 4.1 Dokumentasi tugas pada platform ClickUp

Selain *sprint planning*, terdapat juga beberapa *event* yang harus dilakukan di dalam Scrum seperti *daily stand up meeting*, *sprint review*, dan *retrospective*. Dikarenakan terdapat anggota tim yang merupakan karyawan dan memiliki kesibukan di tim inti masing-masing, beberapa *event* tersebut sulit untuk dilaksanakan. Hal ini yang menyebabkan Scrum tidak terlaksana secara maksimal di dalam tim percepatan. Setelah berdiskusi langsung dengan *product owner*, metode pengembangan yang tadinya Scrum berubah haluan menjadi Waterfall.

Metode pengembangan Waterfall dipilih untuk menyesuaikan keadaan tim. Waterfall menjadi pilihan karena tidak memiliki *event* harian seperti pada Scrum. Selain itu, Waterfall memiliki tahapan yang terstruktur, berurutan secara linier, dan tidak menerapkan iterasi secara responsif. Hal ini sesuai dengan kondisi tim yang jarang memiliki waktu untuk melakukan *meeting*.

Terdapat perbedaan yang ditemukan antara teori metode pengembangan yang telah dipelajari di perkuliahan dengan yang dilaksanakan di lapangan ketika magang. Pada dasarnya, setiap tahapan pada Waterfall membutuhkan dokumentasi yang diperlukan untuk tahapan lainnya. Namun, selama menggunakan metode Waterfall di lapangan saat magang, dokumen-dokumen tersebut tidak ada. Contohnya adalah tidak adanya dokumen pada tahap perancangan seperti SDD (*Software Design Document*) dan SRS (*Software Requirement Specification*). Desain dan juga *requirement* yang harus penulis implementasikan pada tahap implementasi hanya berdasarkan desain prototipe antarmuka saja.

4.1.3 Koneksi *Back-end* ke *Front-end Mobile*

Selain itu, walaupun fokus penulis adalah pengembangan aplikasi *mobile*, penulis juga terlibat dalam pengerjaan aplikasi secara *fullstack* selama magang. Meskipun di perkuliahan sudah mempelajari konsep *fullstack* dengan penggunaan PHP sebagai bahasa pemrograman *backend* dalam pengembangan *website*, namun untuk menampilkan atau menghubungkan data dari basis data, tidak menggunakan API. Alih-alih menggunakan API, yang dilakukan adalah dengan menggunakan metode pemrograman *server-side* dengan PHP. Dalam metode ini, yang dilakukan adalah menulis kode PHP untuk melakukan koneksi ke basis data, mengambil data yang diperlukan dari tabel, dan kemudian memproses data tersebut untuk membangun tampilan HTML. Sedangkan yang dilakukan di lapangan, untuk menghubungkan antara *back-end* dan *front-end*(dalam hal ini *mobile*), penulis harus mengembangkan API dan kemudian melakukan pemanggilan API(*API call*) pada *front-end* untuk menampilkan data.

Penggunaan API untuk komunikasi antara aplikasi *back-end* ke *front-end* memiliki beberapa keuntungan mulai dari pemisahan tanggung jawab antara logika bisnis *back-end* dan antarmuka *front-end*, dapat digunakan untuk menghubungkan *back-end* ke banyak *front-end* (reusabilitas kode), dan memungkinkan tim untuk mengembangkan *back-end* dan *front-end* secara paralel, karena kode *back-end* dan *front-end* dikerjakan secara terpisah.

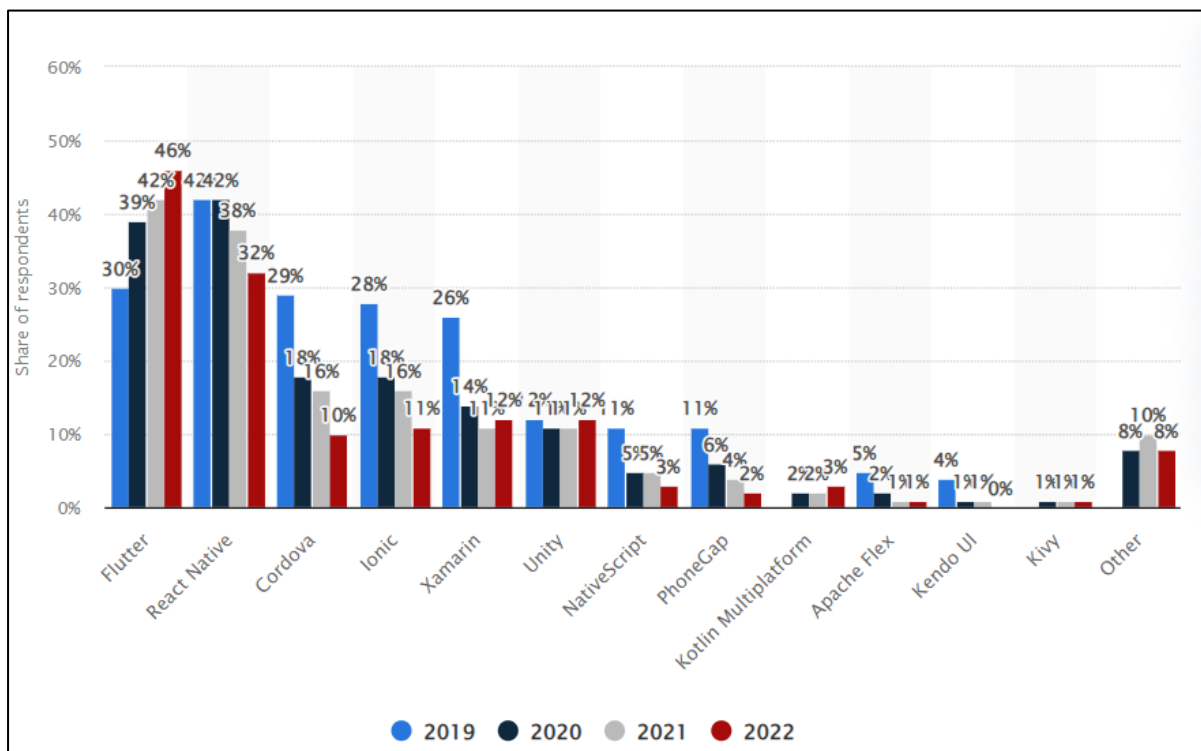
4.2 Pembelajaran Magang

Selama menjalani magang di BSI UII dan bekerja pada proyek I'm UII, penulis berkesempatan untuk mempelajari sebuah teknologi yang belum pernah penulis pelajari sebelumnya dalam pengembangan aplikasi *mobile*, yaitu Flutter. Penulis jadi mengetahui bahwa pengembangan aplikasi *mobile* dapat dilakukan secara multiplatform sekaligus dengan menggunakan *framework* Flutter. Penulis juga jadi mengetahui keuntungan yang diberikan

Flutter, yaitu pengerjaan proyek yang lebih cepat karena hanya membutuhkan satu basis kode saja untuk mengembangkan aplikasi multiplatform.

Selain itu, penulis juga terlibat dalam pengembangan API untuk fitur What's on UII. Sebelumnya, penulis belum memiliki pengalaman langsung dalam mengembangkan maupun menggunakan API. Namun, selama magang, penulis mendapatkan kesempatan untuk mempelajari konsep-konsep dasar dalam pengembangan API seperti *endpoints*, *HTTP methods*, *request API*, *response API*, dan format standar API yaitu JSON. Penulis juga menjadi lebih paham terkait penerapan API pada aplikasi *front-end mobile* menggunakan Flutter.

Secara keseluruhan, pengalaman magang di BSI UII dan mengerjakan fitur What's on UII pada proyek aplikasi I'm UII telah menjadi sumbangan pengetahuan yang bermanfaat dalam karir penulis di masa depan, karena saat ini Flutter merupakan *framework* multiplatform yang paling banyak digunakan. Buktinya dapat dilihat dalam grafik pada Gambar 4.2, yang merupakan grafik penggunaan Flutter oleh *software developer* di seluruh dunia. Berdasarkan survei terbaru yang telah dilakukan (Statista, 2023), pada tahun 2022, Flutter merupakan *framework mobile* multiplatform yang paling banyak digunakan oleh perusahaan pengembang perangkat lunak di dunia dengan persentase 46%. Oleh karena itu, dengan mempelajari Flutter selama magang dapat membuka peluang penulis untuk meniti karir di masa depan.



Gambar 4.2 Grafik penggunaan *framework mobile* multiplatform

Sumber: (Statista, 2023)

BAB V

PENUTUP

5.1 Kesimpulan

Penulis telah melaksanakan magang di Badan Sistem Informasi Universitas Islam Indonesia (BSI UII). Dalam magang ini, penulis diberi tugas untuk mengerjakan salah satu fitur pada aplikasi *mobile* yang diberi nama I'm UII, yakni fitur berita yang diberi nama What's on UII. Berdasarkan hasil dari magang dan pengembangan fitur What's on UII pada aplikasi *mobile* I'm UII, yang dikembangkan menggunakan metode Waterfall, dapat disimpulkan sebagai berikut:

- a. Fitur What's on UII pada aplikasi *mobile* I'm UII berhasil dikembangkan dengan menerapkan metode Waterfall yang dimulai dari tahapan analisis kebutuhan, desain, implementasi, hingga pengujian. Saat ini aplikasi masih dalam tahap pengembangan, sehingga belum mencapai tahap pemeliharaan.
- b. Fitur What's on UII pada aplikasi *mobile* I'm UII berhasil dikembangkan. Fitur ini dikembangkan menggunakan Flutter dengan bahasa pemrograman Dart. Penulis juga diminta mengembangkan aplikasi *back-end* pada fitur ini. Untuk pengembangan aplikasi *back-end* dan API, bahasa yang digunakan adalah PHP dengan *framework* Lumen.
- c. Fitur What's on UII pada aplikasi I'm UII berhasil dikembangkan sesuai dengan beberapa fungsionalitas yang telah ditetapkan tim produk, seperti menampilkan daftar berita, menampilkan detail berita, berlangganan penerbit berita, menyimpan berita dengan menambahkannya ke favorit, menyalakan notifikasi pengingat pembaca berita, hingga membagikan berita.
- d. Metode pengujian yang digunakan pada pengembangan fitur What's on UII adalah metode Black Box Testing. Berdasarkan hasil dari pengujian, seluruh fungsionalitas berhasil dijalankan.
- e. Selama melaksanakan magang, penulis menerapkan beberapa ilmu yang telah didapatkan dari perkuliahan seperti penerapan metode Waterfall dalam mengembangkan aplikasi. Namun terdapat juga beberapa hal baru yang juga menjadi perbedaan antara ilmu yang didapatkan di perkuliahan dan juga di lapangan. Seperti pengembangan aplikasi *multiplatform*, dan pengembangan API.

5.2 Saran

Beberapa saran terkait pelaksanaan magang serta pengembangan fitur What's on UII pada aplikasi *mobile* I'm UII adalah sebagai berikut:

- a. Terkait notifikasi, saat ini pengguna hanya mendapatkan notifikasi dari pengingat membaca berita yang telah ia atur. Akan lebih baik jika pengguna juga bisa mendapatkan notifikasi ketika penerbit yang dilanggan mengunggah berita baru.
- b. Saat ini, fitur pencarian pada aplikasi I'm UII belum berjalan. Untuk memudahkan pengguna mendapatkan berita atau informasi, akan lebih baik jika fitur pencarian dikembangkan. Selain pencarian, bisa juga dengan menggunakan filter semisal filter berupa tombol *dropdown*.
- c. Terdapat beberapa hal baru di lapangan yang umum digunakan namun belum dipelajari di perkuliahan. Seperti pengembangan API serta pengaplikasian API dalam aplikasi. Untuk kedepannya, alangkah lebih baik jika pelajaran di perkuliahan disesuaikan dengan hal-hal yang umum digunakan di industri saat ini.

DAFTAR PUSTAKA

- Adisty, N. (2023, 7 5). *Mengulik Perkembangan Penggunaan Smartphone di Indonesia*. Retrieved from GoodStats: <https://goodstats.id/>
- Arif, M. A., Kusumo, D. S., & Puspitasari, S. Y. (2021). Optimasi Pengembangan Aplikasi Cross-platform Berbasis Flutter Menggunakan Pendekatan Arsitektur Model MVI (Model-View-Intent). *e-Proceeding of Engineering*, 8(5, Oktober).
- Badan Sistem Informasi Universitas Islam Indonesia. (2023, Juli 2). *Sekilas Tentang BSI*. Retrieved from Badan Sistem Informasi: <https://bsi.uui.ac.id>
- Bidang Hubungan Masyarakat Universitas Islam Indonesia. (2023, 7 9). *Branding Guidelines*. Retrieved from Bidang Hubungan Masyarakat: <https://humas.uui.ac.id>
- Duha, D. L., Sarkum, S., Munthe, I. R., & Purnama, I. (2018). Aplikasi Berita Online Berbasis Android: Studi pada Pemerintah Kabupaten Labuhanbatu. *Jurnal Informatika: Jurnal Pengembangan IT*, 380-386.
- Haryono, K., & Budi Negara, R. M. (2020). Sistem Informasi Kegiatan dan Infak Masjid: Studi Kasus Masjid Ulil Albab UII Yogyakarta. *AUTOMATA*, 1(2).
- Ilham, A. (2023, 8 16). *Masifnya Peran Flutter di Perusahaan Teknologi*. Retrieved from Dicoding: <https://www.dicoding.com/blog/masifnya-peran-flutter-di-perusahaan-teknologi/>
- Jela, R., & Suratno, B. (2022). Pengembangan Aplikasi Bergerak(Mobile App) Penerimaan Mahasiswa Baru. *AUTOMATA*.
- Jorgi, S., Asmawati, & Zulkarnaim, N. (2022). PENGEMBANGAN APLIKASI MOBILE LAYANAN PARIWISATA LOKALSULAWESI BARAT MENGGUNAKAN FRAMEWORK FLUTTER. *J-Icon : Jurnal Komputer dan Informatika*, 10(2, Oktober), 114-121.
- Karim, M. A., & Ardiansyah, A. R. (2022). ANALISIS DAN PERANCANGAN APLIKASI MOBILE UNTUK DONASIMENGGUNAKAN METODE HYBRID BERBASIS REACT NATIVE. *Jurnal Informatika terpadu*, 26-34.
- Katti, V. S., & K, S. (2022). Flutter –Cross Platform Ide for Mobile Applications. *Grenze International Journal of Engineering and Technology*, 8(1, January).
- Pramudita, R., Arifin, R. W., Alfian, A. N., Safitri, N., & Anwariya, S. D. (2021). PENGGUNAAN APLIKASI FIGMA DALAM MEMBANGUN UI/UX YANG

- INTERAKTIF PADA PROGRAM STUDI TEKNIK INFORMATIKA STMIK TASIKMALAYA. *Jurnal Buana Pengabdian*, 3(1, Februari), 149-154.
- R, S. R., & Saputra, S. A. (2023). PERANCANGAN UI/UX DESIGN PADA APLIKASI JASA FREELANCER BERBASIS ANDROID MENGGUNAKAN METODE USER CENTERED DESIGN. *Jurnal Ilmiah MATRIK*, 25(1, April), 7-14.
- Rao, P. S., Pavan, B., Sarivastava, A., Amani, K. V., & Sharma, A. (2022). DISTINCTION OF MOBILE FRAMEWORKS: FLUTTER VS NATIVE APPS. *International Research Journal of Modernization in Engineering Technology and Science*, 2422-2427.
- Statista. (2023, 8 16). *Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2022*. Retrieved from statista: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>
- Suharto, A., & Mardiana. (2020). APLIKASI ERESHA MOBILE BERBASIS ANDROID DENGAN METODE PERSONAL EXTREME PROGRAMMING DI ERA INDUSTRI 4.0. *JurTI: Jurnal Teknologi Informasi*, 335-344.
- Suhendro, J. M., Sudarma, M., & Khrisne, D. C. (2021). Rancang Bangun Aplikasi Seluler Penyedia Jasa Perawatan dan Kecantikan Menggunakan Framework Flutter. *Jurnal SPEKTRUM*, 8(2, Juni), 68-82.
- Swarna, I., Purnama, J., & Anthony, R. (2020). Cross-Platform Analysis and Development of Online Catering Platform (Kunyahku). *Journal of Applied of Information, Communication and Technology*, 79-89.
- Zakaria, T. M., Kasih, J., Halim, N., & Gunadi, M. S. (2023). Rancang Bangun Sistem Pembayaran Kursus Mandarin Dengan Metode Waterfall. *Jurnal Teknik Informatika dan Sistem Informasi*, 9(1, April), 127-140.

LAMPIRAN