

SKRIPSI

PENERAPAN ARSITEKTUR *MVVM* PADA APLIKASI REMINDER JADWAL PERKULIAHAN MENGGUNAKAN METODE *EXTREME PROGRAMMING*

*Sebagai salah satu syarat menyelesaikan
Program Studi Diploma IV/Sarjana Terapan Jurusan Teknik Informatika*



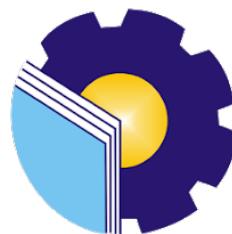
Oleh:

DARU NUGROHO
6304181154

**PROGRAM STUDI D-IV REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BENGKALIS
TAHUN 2021**

LEMBAR PENGESAHAN

PENERAPAN ARSITEKTUR MVVM PADA APLIKASI *REMINDER JADWAL PERKULIAHAN* MENGGUNAKAN METODE *EXTREME PROGRAMMING*



**DARU NUGROHO
NIM. 6304181154**

PROGRAM STUDI D-IV REKAYASA PERANGKAT LUNAK

Telah di seminarkan didepan Dosen Pembimbing dan Dosen Pengaji
tanggal 5 November 2021
dinyatakan telah memenuhi syarat untuk dilanjutkan menjadi TA/Skripsi
Bengkalis, 21 Januari 2022

Menyetujui, Dosen Pembimbing	Mengetahui Ketua Jurusan Teknik Informatika Politeknik Negeri Bengkalis
---	--

<u>Depandi Enda, M.Kom</u> NIP. 199005222019031010	<u>Danuri, M.Cs</u> NIP. 198508122014041001
---	--

PERNYATAAN

LEMBAR PERSEMBAHAN

KATA PENGANTAR

Assalamu'alaikum Wr.Wb

Bismillahirrahmanirrahim, Puji syukur kehadiran Allah swt, yang telah memberikan rahmat, karunia, dan hidayah-Nya kepada penulis. Sholawat dan salam senantiasa tercurah kepada Nabi Muhammad berserta keluarga dan para sahabatnya. Pada akhirnya penulis dapat menyelesaikan skripsi dengan judul “Penerapan Arsitektur *MVVM* Pada Aplikasi *Reminder Jadwal Perkuliahan Menggunakan Metode Extreme Programming*” untuk melengkapi salah satu syarat akademik untuk mendapatkan gelar Sarjana Terapan dengan Program Studi Rekayasa Perangkat Lunak Politeknik Negeri Bengkalis.

Selama menempuh pembelajaran dan dalam proses penyusunan skripsi ini penulis telah banyak mendapat bantuan, bimbingan, dan arahan dari berbagai pihak. Oleh karna itu, dengan kerendahan hati, penulis ingin mengucapkan terimakasih kepada:

1. Bapak Johny Custer, ST.,MT sebagai Direktur Politeknik Negeri Bengkalis.
2. Bapak Armada, S.T.,MT sebagai Wakil Direktur I Politeknik Negeri Bengkalis.
3. Bapak Danuri M.Cs sebagai Ketua Jurusan Teknik Informatika.
4. Ibu Rezki Kurniati, M.Kom sebagai Ketua Program Studi Sarjana Terapan Rekayasa Perangkat Lunak.
5. Bapak Depandi Enda M.Kom sebagai Dosen Pembimbing dan Seluruh Dosen Khususnya Program studi Rekayasa Perangkat Lunak.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih jauh dari sempurna, baik dari segi susunan, bahasa dan penulisan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk menjadi referensi bagi penulis dimasa yang akan datang. Semoga skripsi ini dapat bermanfaat bagi penulis dan pembaca, lebih khususnya yang berkenaan tentang ilmu teknologi dan informasi, sikap dan kepuasan kerja.

Bengkalis, Agustus 2022

Penulis

Daru Nugroho

6304181154

PENERAPAN ARSITEKTUR MVVM PADA APLIKASI REMINDER JADWAL PERKULIAHAN MENGGUNAKAN METODE EXTREME PROGRAMMING

Nama Mahasiswa	:	Daru Nugroho
NIM	:	6304181154
Dosen Pembimbing	:	Depandi Enda, M.Kom

Abstrak

Keterbacaan kode dan performa aplikasi pada pengembangan perangkat lunak terutama pada pengembangan aplikasi *Android* menjadi hal yang penting karena dengan kualitas kode yang baik dapat memudahkan dalam *debugging error* dan memperbaiki *bug* yang ada dalam kode sehingga dapat meningkatkan performa aplikasi. Seiring dengan berkembangnya pengetahuan dalam pengembangan perangkat lunak maka muncullah cara untuk membangun perangkat lunak yang memiliki performa handal, mudah dalam pemeliharaan, mudah dalam proses *debugging*, dapat dimodifikasi dan mudah saat melakukan pengujian, yaitu dengan menerapkan *design pattern* atau pola perancangan. salah satunya adalah dengan menerapkan *design pattern* MVVM (*Model View ViewMode*). Pada arsitektur ini bertujuan memisahkan antara komponen aplikasi dengan komponen lain sehingga pada komponen memiliki peran masing-masing. Metode *Extreme Programming* adalah metode pengembangan perangkat lunak yang memiliki empat tahapan dalam prosesnya, yaitu perencanaan, desain, pengkodean dan pengujian. Dengan melalui keempat tahap tersebut diharapkan dapat menghasilkan perangkat lunak yang sesuai dengan kebutuhan pengguna. Maka dari itu penelitian ini bertujuan menerapkan arsitektur perangkat lunak MVVM dan metode *Extreme Programming*. Pada aplikasi *reminder* jadwal perkuliahan. Dalam penelitian ini menggunakan bahasa pemrograman *kotlin* untuk membangun aplikasi *Android* dan *framework laravel* untuk membangun *web service*.

Kata Kunci : MVVM, Extreme Programming, Reminder, Web Service

***IMPLEMENTATION OF MVVM ARCHITECTURE IN
THE APPLICATION OF LECTURE SCHEDULE
REMINDER USING EXTREME PROGRAMMING
METHOD***

Name Of Student : Daru Nugroho
Student ID Number : 6304181154
Supervisor : Depandi Enda, M.Kom

Abstract

Code readability and application performance in software development, especially in android application development, are important because good code quality can facilitate debugging errors and fixing bugs in the code can improve application performance. Along with the development of knowledge in software development, there are ways to build software that has reliable performance, is easy to maintain, easy to debug, can change and is easy to test, namely by applying design patterns or design patterns. one of them is to apply the MVVM (Model View ViewMode) design pattern. This architecture aims to separate the application components from other components so that the components have their respective roles. The Extreme Programming method is a software development method that has four stages in its implementation, namely planning, design, coding, and testing. By going through these four stages, it is expected to produce software that suits user needs. Therefore, this study aims to apply the MVVM software architecture and the Extreme Programming method. on the lecture schedule reminder application. In this study, using the Kotlin programming language to build android applications and the laravel framework to build web services.

Keywords : MVVM, Extreme Programming, reminder, Web service

DAFTAR ISI

LEMBAR PENGESAHAN.....	ii
PERNYATAAN.....	iii
LEMBAR PERSEMPAHAN.....	iv
KATA PENGANTAR.....	v
Abstrak.....	vii
<i>Abstract</i>	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	14
1.1 Latar Belakang.....	14
1.2 Rumusan Masalah.....	17
1.3 Tujuan.....	17
1.4 Manfaat.....	17
BAB II TINJAUAN PUSTAKA.....	18
2.1 Kajian Terdahulu.....	18
2.2 Tabel Perbandingan.....	23
2.3 Landasan Teori.....	29
2.3.1 Android.....	29
2.3.2 <i>Android Studio</i>	30
2.3.3 Kotlin.....	31
2.3.4 <i>MVVM Design Pattern</i>	32
2.3.5 <i>MVP Design Pattern</i>	33
2.3.6 PHP.....	34
2.3.7 Laravel Framework.....	35
2.3.8 <i>Extreme Programming</i>	36
2.3.9 MySQL.....	40
BAB III METODOLOGI PENELITIAN.....	41
3.1 Deskripsi Sistem.....	41
3.1.1 Data.....	41
3.1.2 Alat.....	42
3.1.3 Analisa Kebutuhan.....	44
3.1.4 Kebutuhan Fungsional.....	45
3.1.5 Kebutuhan Non-Fungsional.....	48
3.2 Alur Metode <i>Extreme Programming</i>	49
3.2.1 <i>Planning</i> (Perencanaan).....	49
3.2.2 <i>Design</i> (Perancangan).....	56
3.2.3 <i>Coding</i> (Implementasi).....	68
3.2.4 <i>Testing</i> (Pengujian).....	80
3.2.5 <i>Release</i> (Perilisan).....	120

BAB IV HASIL DAN PEMBAHASAN.....	121
4.1 Hasil.....	121
4.1.1 Tampilan Pilihan Tipe Pengguna.....	122
4.1.2 Tampilan Menu Pilihan Nama Dosen.....	123
4.1.3 Tampilan Jadwal Dosen.....	124
4.1.4 Tampilan Menu Pilihan Atur Waktu Notifikasi.....	126
4.1.5 Tampilan Fitur Mata Kuliah Yang Sedang Berlangsung.....	127
4.1.6 Tampilan Notifikasi Perkuliahan.....	129
4.1.7 Tampilan Alarm Jadwal Perkuliahan.....	130
4.2 Pembahasan.....	132
4.2.1 Penggunaan RAM.....	132
4.2.2 Penggunaan CPU.....	134
4.2.3 Waktu Eksekusi.....	135
BAB V PENUTUP.....	136
5.1 Kesimpulan.....	136
5.2 Saran.....	137
LAMPIRAN.....	138
DAFTAR PUSTAKA.....	139

DAFTAR TABEL

Tabel 2.1. Perbandingan.....	23
Tabel 3.1. Kebutuhan Fungsional Menu Pilih kelas.....	45
Tabel 3.2. Kebutuhan Fungsional Menu Pilih Prodi.....	46
Tabel 3.3. Kebutuhan Fungsional Menu Nama Dosen.....	46
Tabel 3.4. Kebutuhan Fungsional Menu Toggle Reminder.....	46
Tabel 3.5. Kebutuhan Fungsional Menu Mata Kuliah Sekarang.....	47
Tabel 3.6. Kebutuhan Fungsional Menu Atur Waktu Notifikasi.....	47
Tabel 3.7. Kebutuhan Fungsional Notifikasi Jadwal Perkuliahannya.....	47
Tabel 3.8. Kebutuhan Fungsional Daftar Jadwal Perkuliahannya.....	48
Tabel 3.9. Pertanyaan Wawancara.....	49
Tabel 3.10. <i>User Stories</i>	54
Tabel 3.11. Prioritas.....	55
Tabel 3.12. <i>Refactoring</i>	69
Tabel 3.13. <i>Unit Testing</i> Tipe Pengguna Dosen.....	81
Tabel 3.14. <i>Unit Testing</i> Tipe Pengguna Mahasiswa.....	84
Tabel 3.15. <i>Black Box Testing</i> Tipe Pengguna Dosen.....	87
Tabel 3.16. <i>Black Box Testing</i> Tipe Pengguna Mahasiswa.....	105
Tabel 1. Jadwal Penelitian.....	138

DAFTAR GAMBAR

Gambar 2.1. <i>Android</i>	29
Gambar 2.2. <i>Android Studio</i>	30
Gambar 2.3. <i>Kotlin</i>	31
Gambar 2.4. <i>MVVM Design Pattern</i>	32
Gambar 2.5. <i>MVP Design Pattern</i>	33
Gambar 2.6. <i>PHP</i>	34
Gambar 2.7. <i>Laravel Framework</i>	35
Gambar 2.8. <i>Extreme Programming</i>	36
Gambar 2.9. <i>MySQL</i>	40
Gambar 3.1. File Jadwal Perkuliahan.....	41
Gambar 3.2. File Data Mata Kuliah.....	42
Gambar 3.3. Sistem Yang Sedang Berjalan.....	44
Gambar 3.4. Sistem Yang Diusulkan.....	45
Gambar 3.5. Sampel Observasi Permasalahan.....	50
Gambar 3.6. Data Diagram Kuisioner Pertanyaan 1.....	51
Gambar 3.7. Data Diagram Kuisioner Pertanyaan 2.....	52
Gambar 3.8. Data Diagram Kuisioner Pertanyaan 2.....	53
Gambar 3.9. <i>Use Case</i> Aplikasi Reminder Perkuliahan.....	56
Gambar 3.10. <i>Class Diagram</i> Aplikasi Reminder Perkuliahan.....	58
Gambar 3.11. <i>CRC Card</i> Aplikasi Reminder Perkuliahan.....	60
Gambar 3.12. <i>Prototype</i> notifikasi mahasiswa.....	61
Gambar 3.13. <i>Prototype</i> jadwal perkuliahan mahasiswa.....	61
Gambar 3.14. <i>Prototype</i> atur waktu notifikasi.....	62
Gambar 3.15. <i>Prototype</i> Jadwal perkuliahan dosen.....	63
Gambar 3.16. <i>Prototype</i> notifikasi dosen.....	63
Gambar 3.17. <i>Prototype</i> atur waktu notifikasi.....	64
Gambar 3.18. <i>Prototype</i> pilihan jurusan dan semester.....	65
Gambar 3.19. <i>Prototype</i> pilihan nama dosen.....	65
Gambar 3.20. <i>Prototype</i> atur waktu notifikasi.....	66
Gambar 3.21. <i>Prototype</i> menu mata kuliah sedang berlangsung dosen.....	67
Gambar 3.22. <i>Prototype</i> menu mata kuliah sedang berlangsung mahasiswa.....	67
Gambar 3.23. <i>Pair Programming</i>	78
Gambar 3.24. <i>Continous Integration</i>	79
Gambar 3.25. <i>Release</i> Aplikasi <i>Reminder</i> Jadwal Perkuliahan.....	120
Gambar 4.1. Tampilan Menu Tipe Pengguna.....	122
Gambar 4.2. Tampilan Menu Nama Dosen.....	123
Gambar 4.3. Tampilan Jadwal Dosen.....	124
Gambar 4.4. Tampilan Menu Atur Waktu Notifikasi.....	126
Gambar 4.5. Tampilan Fitur Mata Kuliah Yang Sedang Berlangsung.....	127
Gambar 4.6. Tampilan Fitur Notifikasi Perkuliahan.....	129

Gambar 4.7. Tampilan Fitur Alarm Jadwal Perkuliahan.....	130
Gambar 4.8. Penggunaan RAM Arsitektur MVVM Dan MVP.....	133
Gambar 4.9. Perbandingan Penggunaan CPU Arsitektur MVVM Dan MVP.....	134
Gambar 4.10. Pengukuran Kecepatan Waktu Eksekusi Arsitektur MVVM.....	135

BAB I

PENDAHULUAN

1.1 Latar Belakang

Android adalah sistem operasi dan *platform* pemrograman yang dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya seperti tablet. *Android* dapat berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. *Android* menyertakan *kit development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna *Android*. *Android* juga menyediakan pasar untuk mendistribusikan aplikasi. Secara keseluruhan, *Android* menyatakan ekosistem untuk aplikasi seluler. (Developer Training Team, 2016)

Dalam hal pengembangan aplikasi *Android*, Google sebagai pemilik *Android* telah menyediakan kumpulan *tools* yang disebut dengan *Android Software Development Kit (SDK)*. Yang akan memudahkan *developer* untuk mengembangkan aplikasi *Android*. Dimana pada *SDK* terdiri dari serangkaian alat *development* yang menyertakan *debugger*, pustaka perangkat lunak, kode pratalis, perangkat emulator, dokumentasi dan contoh kode. (Developer Training Team, 2016)

Pada proses pengembangan aplikasi *Android* masih banyak terdapat permasalahan umum yang sering terjadi yaitu bercampurnya kode yang menangani tampilan dengan kode yang menangani logika aplikasi sehingga kedua kode tersebut berkumpul pada satu *file* atau *class* yang sama, dan membuat suatu *class* memiliki banyak tugas, dalam hal ini dengan bercampurnya logika untuk tampilan dengan logika untuk aplikasi akan menyulitkan *developer* ketika melakukan pengujian, pemeliharaan, menangani *error*, menambahkan fitur baru pada sistem dan masih banyak lagi. permasalahan tersebut dapat dihindari dengan menerapkan *design pattern* atau pola perancangan.

Untuk mengatasi permasalahan terkait penulisan kode aplikasi, terdapat beberapa *design pattern* yang sering digunakan salah satunya yang direkomendasikan oleh Google untuk digunakan dalam mengembangkan aplikasi *Android* yaitu arsitektur MVVM (*Model View ViewModel*) yaitu sebuah pola arsitektur yang memisahkan antara *user interface* dari logika aplikasi dengan tujuan untuk menjaga kode UI agar tetap sederhana agar mudah untuk dikelola.

Arsitektur MVVM adalah pola arsitektur yang dapat digunakan untuk merancang dan mengembangkan aplikasi dengan skala yang kompleks untuk mencapai *reusability*, *maintainability* dan *extensibility* yang lebih baik, dengan tingkat *coupling* yang rendah. (Nagaraj & Chethana, 2020)

Penerapan arsitektur MVP (*Model View Presenter*) dapat memecah fungsi dan tampilan, sehingga untuk pengembangan selanjutnya menjadi lebih mudah karena fungsi yang akan dikoreksi maupun fungsi yang akan ditambah tidak berpengaruh pada fungsi maupun tampilan yang lain. (Muttaqin et al., 2020)

Selain itu menurut Rizki, DKK dimana status penggunaan energi pada aplikasi smartphone tanpa pola perancangan adalah *heavy* sedangkan dengan menerapkan pola perancangan adalah *medium*. (Rizki et al., 2020)

Untuk membantu pengembang dalam membangun aplikasi secara efisien, Google telah menyediakan Lingkungan pengembangan yang terintegrasi atau disebut dengan IDE (*Integration Development Environment*) yaitu *Android Studio*, dengan menyediakan fitur lanjutan untuk pengembangan seperti, *debug*, dan pemaketan aplikasi *Android*.

Politeknik Negeri Bengkalis merupakan salah satu perguruan tinggi vokasi Negeri yang terletak di Provinsi Riau Kabupaten Bengkalis, pada saat ini di Politeknik Negeri Bengkalis, dalam hal melihat jadwal perkuliahan yang akan diikuti baik dosen maupun mahasiswa, berdasarkan pengamatan penulis dan hasil kuisioner dari mahasiswa. penulis menemukan permasalahan yang sering di alami

mahasiswa pada Politeknik Negeri Bengkalis dalam hal melihat jadwal perkuliahan yaitu, mahasiswa kesulitan pada saat membuka *file excel*, sulit mencari jadwal perkuliahan yang akan diikuti, karena semua jadwal perkuliahan digabung menjadi satu *file*. akibat nya sebagian dari mahasiswa sering men-screenshot jadwal perkuliahan yang akan diikuti dan menjadikanya sebagai *wallpaper* ponsel. Berdasarkan permasalahan yang dialami oleh mahasiswa, penulis tertarik untuk membuat aplikasi berbasis *Android* yang dapat menampilkan jadwal perkuliahan sekaligus terdapat fitur pengingat jadwal perkuliahan yang akan diikuti bagi mahasiswa dan dosen.

Dengan memanfaatkan keunggulan dari arsitektur MVVM penulis akan menerapkan arsitektur MVVM pada aplikasi *reminder* jadwal perkuliahan di Politeknik Negeri Bengkalis dengan menggunakan metode pengembangan *Extreme Programming*. Penulis akan membuat aplikasi *reminder* jadwal perkuliahan karena aplikasi tersebut akan bertugas sebagai pengingat dan tentunya aplikasi akan terus berjalan dilatar belakang, pada keadaan seperti ini aplikasi akan menggunakan sumber daya yang dimiliki dari perangkat seperti penggunaan memori RAM dan pengguna CPU. Dimana akan dilakukan pengujian dari segi performa dengan pengukuran berdasarkan tiga aspek yaitu, waktu eksekusi, penggunaan CPU, dan penggunaan memori yang akan dibandingkan dengan aplikasi serupa tetapi dibangun dengan menggunakan arsitektur MVP.

1.2 Rumusan Masalah

Berdasarkan pokok permasalahan yang telah di paparkan pada latar belakang diatas, terdapat beberapa permasalahan yang dapat indentifikasi yaitu :

1. Bagaimana menerapkan pola perancangan MVVM (*Model View ViewModel*) dalam pengembangan aplikasi *reminder* jadwal perkuliahan pada Politeknik Negeri Bengkalis.
2. Bagaimana mengukur performa aplikasi dari aspek waktu eksekusi, penggunaan CPU dan penggunaan memori pada aplikasi reminder jadwal perkuliahan pada Politeknik Negeri Bengkalis.

Berdasarkan rumusan masalah yang telah diuraikan diatas, maka penilitian ini difokuskan dengan batasan masalah sebagai berikut :

1. Objek penelitian ini adalah aplikasi reminder jadwal perkuliahan untuk Prodi Rekayasa Perangkat Lunak di Politeknik Negeri Bengkalis.
2. Penelitian difokuskan pada penerapan arsitektur MVVM pada aplikasi reminder jadwal perkuliahan di Politeknik Negeri Bengkalis.

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mengetahui kelebihan arsitektur MVVM terhadap performa aplikasi yang akan diterapkan dalam pengembangan aplikasi *reminder* jadwal perkuliahan pada Politeknik Negeri Bengkalis.

1.4 Manfaat

Penelitian ini diharapkan dapat berguna bagi pengembang lain dalam hal memilih arsitektur untuk digunakan dalam mengembangkan aplikasi *Android* dan untuk memudahkan pengembang lain dalam menambahkan fitur baru atau mengembangkan ulang aplikasi *reminder* jadwal perkuliahan.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Terdahulu

“*Performance Comparison of Native Android Application on MVP and MVVM*”. Pada penelitian ini mengatakan bahwa pada eksperimen yang dilakukan menghasilkan bahwa, aplikasi dengan arsitektur MVVM lebih baik dalam hal penggunaan CPU dan waktu eksekusi, sedangkan aplikasi dengan arsitektur MVP lebih baik dalam hal penggunaan memori. Penggunaan CPU pada arsitektur MVVM lebih rendah dengan selisih rata-rata 0.55%, untuk waktu eksekusi pada arsitektur MVVM lebih cepat dengan selisih rata-rata 126.21 ms, sedangkan untuk penggunaan memori aplikasi dengan arsitektur MVP lebih rendah dengan selisih rata-rata 0.92 mb. hal ini terjadi karena arsitektur MVVM mempunyai library tambahan yaitu data binding yang mengakibatkan penggunaan CPU yang lebih rendah dan waktu eksekusi yang lebih cepat, tetapi dalam hal penggunaan memori arsitektur MVVM lebih tinggi dari arsitektur MVP. (Wisnuadhi et al., 2020)

“*Analisis Pengaruh Arsitektur MVVM dan MVP pada Performa Database GreenDao*”. Pada penelitian ini mengatakan bahwa adanya pengaruh arsitektur terhadap performa *database* adalah benar. Pada eksperimen operasi *insert* data, Sebagian besar skenario *insert* diungguli oleh MVVM dan *greenDao*. MVP dan *greenDao* mengungguli pada waktu eksekusi *view* untuk 10.000, 100.000, dan 500.000 data. Pada eksperimen *select* data, sebagian besar diungguli oleh MVVM dan *greenDao*, namun MVP dan *greenDao* mengungguli pada sebagian besar waktu eksekusi *view*. Pada eksperimen operasi *update* data, sebagian besar skenario *update* diungguli oleh arsitektur MVP. Pada eksperimen *delete* data, dimana arsitektur MVP mengungguli banyak waktu eksekusi (*database* dan total), sedangkan MVVM dan *greenDao* unggul pada waktu eksekusi *view*. (Muharram et al., 2021)

“Android Application on MVVM Architecture to Change Database”. Pada penelitian ini mengatakan bahwa Arsitektur MVVM adalah pola arsitektur yang dapat digunakan untuk merancang dan mengembangkan aplikasi dengan skala yang kompleks untuk mencapai *reusability*, *maintainability* dan *extensibility* yang lebih baik, dengan tingkat *coupling* yang rendah. Salah satu cara untuk mengoptimalkan desain adalah dengan menerapkan arsitektur MVVM pada pengembangan aplikasi, pada arsitektur ini membagi desain yang kompleks menjadi bagian-bagian terpisah sehingga UI aplikasi dapat diubah tanpa mengganggu kode utama sedangkan untuk aplikasi yang sederhana arsitektur MVVM sebaiknya dihindari untuk implementasikan. (Nagaraj & Chethana, 2020)

“Perbandingan Kinerja Pola Perancangan *MVC* , *MVP* , dan *MVVM* Pada Aplikasi Berbasis *Android* (Studi kasus : Aplikasi Laporan Hasil Belajar Siswa SMA BSS)”. Pada penelitian ini mengatakan bahwa Perbedaan peforma aplikasi tanpa pola perancangan dengan menerapkan pola perancangan, berdasarkan penelitian ini terlihat dari penggunaan energi *smartphone* dan memori yang dipakai, yang tercantum pada *Android Studio profiler* pada saat tahap pengujian, dimana status penggunaan energi pada *smartphone* tanpa pola perancangan adalah *heavy* sedangkan dengan menerapkan pola perancangan adalah *medium*. Untuk memori yang digunakan pada aplikasi tanpa pola perancangan sebesar 92 MB sedangan dengan pola perancangan sebesar 59,7 MB pada *MVC*, 59 MB pada *MVP* dan 73,2 MB untuk *MVVM*. (Rizki et al., 2020)

“Implementation of Model-View-ViewModel (MVVM) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application”. Pada penelitian ini mengatakan bahwa penerapan arsitektur MVVM pada SIAKAD UNIDA Gontor untuk pengguna *Android* dalam meningkatkan *user friendly* telah berhasil. pada pengujian *black box* menampilkan hasil bahwa aplikasi telah berjalan lancar tanpa ada *error*. sedangkan pada hasil kuisioner angket menunjukkan implementasi dari arsitektur MVVM sudah sesuai dengan skenario

dan mendapat respon positif dari mahasiswa, dosen dan staf BAAK UNIDA Gontor. (Arif et al., 2020)

“Penerapan Arsitektur *MVP* dalam Pengembangan Aplikasi Pemesanan Tiket Seminar Berbasis *Android*”. Pada penelitian ini mengatakan Penerapan arsitektur *MVP* (*Model-View-Presenter*) dalam pengembangan aplikasi pemesanan tiket berbasis *Android* membuat pengembangan aplikasi menjadi lebih terstruktur dan lebih mudah dalam proses *debugging*. (I Made Sukarsa et al., 2020)

“*Mobile Data Monitoring System Based on MVP Mode*”. Pada penelitian ini mengatakan bahwa penerapan arsitektur *MVP* dapat mengurangi *coupling code*, meningkatkan keterbacaan kode dan memberikan efisiensi pada pengujian unit. Dan pada perangkat lunak yang diuji memiliki kelancaran dan stabilitas. (FAN et al., 2018)

“Aplikasi Stok Audit Di PT. Media Sarana Data Berbasis *Android* Dengan Arsitektur *Model View Presenter*”. Pada penelitian ini mengatakan Aplikasi Stok Audit Gmedia dengan menerapkan arsitektur *MVP* (*Model View Presenter*) dapat memecah fungsi dan tampilan, sehingga untuk pengembangan selanjutnya menjadi lebih mudah karena fungsi yang akan dikoreksi maupun fungsi yang akan ditambah tidak berpengaruh pada fungsi maupun tampilan yang lain. . (Muttaqin et al., 2020)

“*A Tale of Two Development Approach : Empirical Study on The Maintainability and Modularity of Android Mobile Application with Anti-Pattern and Model-View-Presenter Design Pattern*”. Pada penelitian ini mengatakan terdapat hubungan antara pengembangan aplikasi *mobile* berbasis *Android* dengan menggunakan pola Desain *MVP* atau tidak. Menggunakan pola desain, di sisi lain, akan meningkatkan jumlah baris kode dan upaya yang diperlukan untuk menulis kode tetapi dalam sisi lain dapat mengurangi kompleksitas kode program. Hasil dari modularitas perangkat lunak dapat meningkatkan pemeliharaan perangkat

lunak secara signifikan seiring dengan penggunaan Pola Desain MVP. Hal ini menunjukkan bahwa kode program yang ditulis dengan pola Desain MVP lebih mudah untuk diperluas dan dapat mencapai kemudahan dalam pemeliharaan perangkat lunak. (Prabowo, 2018)

“Rancang Bangun Aplikasi *Class Reminder* Berbasis *Android*”. Pada penelitian ini menunjukkan berdasarkan penelitian yang telah dilaksanakan, yaitu merancang dan membuat aplikasi pengingat kelas (*class remainder*) berbasis *Android* yang mampu mengingatkan mahasiswa tentang jadwal perkuliahan, jadwal *mid test*, jadwal *final test*, dan batas waktu pengumpulan tugas telah terpenuhi. Hal ini dibuktikan berdasarkan pengujian *whitebox* dan *blackbox* ditambah dengan survei menggunakan kuisioner ke pengguna. Hasil pengujian *whitebox* dengan menggunakan metode *Cyclomatic Complexity (CC)* membuktikan bahwa sistem telah benar dan tidak memiliki kesalahan baik dari segi logika maupun fungsi. Hasil pengujian *blackbox* membuktikan bahwa proses aplikasi yang dilakukan sudah benar sehingga fungsional sistem sudah dapat menghasilkan output yang diharapkan. (Tone, 2018)

“Metode *Extreme Programming* Pada Pembangunan *Web* Aplikasi Seleksi Peserta Pelatihan Kerja”. Pada penelitian ini mengatakan bahwa Berdasarkan hasil penelitian yang telah dilakukan ditemukan bahwa aplikasi ujian seleksi peserta pelatihan kerja dapat memberikan kemudahan kepada masyarakat untuk mendapatkan informasi terkait pelaksanaan pelatihan, memudahkan dalam melakukan pendaftaran dan melaksanakan ujian seleksi peserta pelatihan kerja. Metode *Extreme Programming (XP)* dapat digunakan untuk membangun atau mengembangkan perangkat lunak. Agar aplikasi ini berjalan sesuai dengan fungsinya oleh karena itu perlu dibuatkan standarisasi berupa petunjuk teknis penggunaan. (Supriyatna, 2018)

“Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android”. Pada penelitian ini mengatakan bahwa Penelitian ini menghasilkan aplikasi yang mampu menetukan tingkat kemiripan usulan judul skripsi mahasiswa. Sedangkan metode XP telah menghasilkan *software* yang sesuai dengan kebutuhan. Hal ini didukung dari hasil *black box testing* yang menunjukkan bahwa semua fitur dari aplikasi dapat berfungsi dengan baik, artinya aplikasi dapat digunakan. (Ahmad et al., 2020)

2.2 Tabel Perbandingan

Tabel 2.1. Perbandingan

No.	Penulis	Tahun	Judul	Metode	Hasil Penelitian	Perbedaan
1.	Bambang Wisnuadhi, Ghifari Munawar, dan Ujang Wahyu	2020	<i>Performance Comparison of Native Android Application on MVP and MVVM</i>		Asitektur <i>MVVM</i> menghasilkan peforma yang lebih baik dari arsitektur <i>MVP</i> dan <i>MVC</i> . Arsitektur <i>MVVM</i> lebih baik dari aspek penggunaan <i>CPU</i> dan waktu eksekusi.	Dari penelitian yang dilakukan oleh Bambang Wisnuadhi, DKK adalah berfokus pada membandingkan arsitektur <i>MVC</i> , <i>MVP</i> , <i>MVVM</i> sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder</i> jadwal perkuliahan menggunakan metode pengembangan <i>Extreme Programming</i> .
2.	Mumuh Kustino Muharram, Zikri Ariachandra, Bambang Wisnuadhi, dan Ghifari Munawar	2021	Analisis Pengaruh Arsitektur <i>MVVM</i> dan <i>MVP</i> pada Performa <i>Database GreenDao</i>		Aplikasi yang menerapkan arsitektur <i>MVVM</i> lebih unggul pada sebagian besar skenario, seperti <i>insert</i> data, <i>select</i> data, <i>delete</i> data, namun keunggulan <i>MVVM</i> dari aspek waktu eksekusi tidaklah signifikan bahkan pada waktu eksekusi <i>view</i> sebagian besar diungguli oleh <i>MVP</i> .	Dari penelitian yang dilakukan oleh Mumuh Kustino Muharram, DKK yang berfokus pada menganalisis pengaruh arsitektur <i>MVP</i> dan <i>MVVM</i> pada performa <i>Database GreenDao</i> , sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder</i> jadwal perkuliahan menggunakan metode pengembangan <i>Extreme Programming</i> .

No.	Penulis	Tahun	Judul	Metode	Hasil Penelitian	Perbedaan
3.	Nagaraj M dan Chethana G	2020	<i>Android Application on MVVM Architecture to Change Database</i>		Arsitektur <i>MVVM</i> lebih sesuai digunakan untuk mengembangkan aplikasi dengan skala yang kompleks. Untuk mencapai <i>reusability</i> , <i>maintainability</i> dan <i>extensibility</i> dengan tingkat <i>coupling</i> yang rendah.	Dari penelitian yang dilakukan Nagaraj M, DKK adalah berfokus pada implementasi arsitektur <i>MVVM</i> untuk <i>maintaining database</i> yang besar, sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder jadwal perkuliahan</i> menggunakan metode pengembangan <i>Extreme Programming</i>
4.	Bahrur Rizki Putra Surya, Agi Putra Kharisma, dan Novanto Yudistira	2020	Perbandingan Kinerja Pola Perancangan <i>MVC</i> , <i>MVP</i> , dan <i>MVVM</i> Pada Aplikasi Berbasis <i>Android</i> (Studi kasus : Aplikasi Laporan Hasil Belajar Siswa SMA BSS)		Arsitektur <i>MVP</i> lebih baik diterapkan pada aplikasi laporan hasil belajar karena pola perancangan <i>MVC</i> dan <i>MVP</i> memiliki performa yang sama sedangkan arsitektur <i>MVVM</i> memiliki konsumsi memori yang tinggi. Maka dari itu pola perancangan paling optimal untuk diterapkan pada aplikasi laporan hasil belajar ini adalah pola perancangan <i>MVC</i> dan <i>MVP</i> .	Dari penelitian yang dilakukan oleh Bahrur Rizki Putra Surya, DKK adalah berfokus pada perbandingan kinerja antara arsitektur <i>MVP</i> , <i>MVC</i> , dan <i>MVVM</i> pada aplikasi laporan hasil belajar Siswa SMA BSS, sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder jadwal perkuliahan</i> menggunakan metode pengembangan <i>Extreme Programming</i> .

No.	Penulis	Tahun	Judul	Metode	Hasil Penelitian	Perbedaan
5.	Muhammad Syakir Arif, Aziz Mustafa, dan Dihin Muriyatmoko	2020	<i>Implementation of Model-View-ViewModel (MVVM) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application</i>	<i>Waterfall</i>	Implementasi <i>MVVM</i> pada SIAKAD UNIDA Gontor berbasis <i>mobile</i> telah berhasil. Hasil pengujian <i>Black Box</i> menunjukkan bahwa aplikasi berjalan dengan lancar dan tanpa <i>error</i> . Hasil dari uji angket menunjukkan bahwa implementasi <i>MVVM</i> sesuai dengan skenario dan mendapat tanggapan positif tanggapan dari mahasiswa, dosen dan staf BAAK UNIDA Gontor	Pada penelitian yang dilakukan oleh Muhammad Syakir Arif, DKK adalah berfokus pada penerapan arsitektur <i>MVVM</i> pada Sistem Informasi Akademik UNIDA Gontor, sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> untuk aplikasi <i>reminder jadwal perkuliahan</i> menggunakan metode pengembangan <i>Extreme Programming</i> .
6.	I Made Sukarsa, I Nyoman Piersa, dan I Gede Bagus Premana Putra	2020	Penerapan Arsitektur <i>MVP</i> dalam Pengembangan Aplikasi Pemesanan Tiket Seminar Berbasis <i>Android</i>		Penerapan arsitektur <i>MVP</i> (<i>Model, View, Presenter</i>) dalam pengembangan aplikasi pemesanan tiket berbasis android membuat pengembangan aplikasi menjadi lebih terseruktur dan lebih mudah dalam proses <i>debugging</i> .	Pada penelitian yang dilakukan I Made Sukarsa, DKK adalah berfokus pada penelitian tentang penerapan arsitektur <i>MVP</i> dalam pengembangan aplikasi pemesanan tiket seminar berbasis <i>Android</i> , sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> aplikasi <i>reminder jadwal perkuliahan</i> dengan menggunakan metode pengembangan <i>Extreme Programming</i> .

No.	Penulis	Tahun	Judul	Metode	Hasil Penelitian	Perbedaan
7.	Prabowo Ginanjar	2018	<i>A Tale of Two Development Approach : Empirical Study on The Maintainability and Modularity of Android Mobile Application with Anti-Pattern and Model-View-Presenter Design Pattern</i>		Hasil dari modularitas perangkat lunak dapat meningkatkan pemeliharaan perangkat lunak secara signifikan seiring dengan penggunaan Pola Desain <i>MVP</i> . Hal ini menunjukkan bahwa kode program yang ditulis dengan pola Desain <i>MVP</i> lebih mudah untuk diperluas dan dapat mencapai kemudahan dalam pemeliharaan perangkat lunak.	Dari penelitian yang dilakukan Prabowo Ginanjar adalah berfokus pada membandingkan Anti- <i>Pattern</i> dan Model <i>View Presenter Design Pattern</i> untuk permasalahan pada pemeliharaan perangkat lunak, sedangkan penulis berfokus pada membuat aplikasi penjadwalan otomatis serta aplikasi <i>reminder</i> jadwal perkuliahan dengan menerapkan arsitektur <i>MVVM</i> menggunakan metode pengembangan <i>Extreme Programming</i> .
8.	Muttaqin, Mo-hamad Arif, Susanto Ajib, dan Muslih	2020	Aplikasi Stok Audit Di PT. Media Sarana Data Berbasis Android Dengan Arsitektur Model <i>View Presenter</i>		Penerapan arsitektur <i>MVP</i> pada Aplikasi Stok Audit Di PT. Media Sarana Data Berbasis <i>Android</i> , dapat mencegah fungsi dan tampilan, sehingga untuk pengembangan selanjutnya menjadi lebih mudah karena fungsi yang akan dikoreksi maupun yang akan ditambah tidak berpengaruh pada fungsi maupun tampilan yang lain.	Dari penelitian yang dilakukan Muttaqin Mohamad Arif, DKK adalah berfokus pada penerapan arsitektur <i>MVP</i> pada aplikasi Aplikasi Stok Audit Di PT. Media Sarana Data, sedangkan penulis berfokus pada membuat aplikasi penjadwalan otomatis serta aplikasi <i>reminder</i> jadwal perkuliahan dengan menerapkan arsitektur <i>MVVM</i> menggunakan metode pengembangan <i>Extreme Programming</i> .

No.	Penulis	Tahun	Judul	Metode	Hasil Penelitian	Perbedaan
9.	FAN, Jia-jie ZENG, Yu GAO, Ming-yu	2018	<i>Mobile Data Monitoring System Based on MVP Mode</i>		penerapan arsitektur <i>MVP</i> mengurangi coupling meningkatkan keterbacaan dan memberikan efisiensi pengujian unit.	Dari penelitian yang dilakukan oleh FAN, DKK adalah berfokus pada pembangunan aplikasi <i>Mobile Data Monitoring System</i> , dengan menggunakan arsitektur <i>MVP</i> sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder jadwal perkuliahan</i> menggunakan metode pengembangan <i>Extreme Programming</i> .
10.	Kamaruddin Tone	2018	Rancang Aplikasi <i>Class Reminder</i> Bangun <i>Class</i>	<i>waterfall</i>	Pada perancangan aplikasi <i>Class Reminder</i> , telah sesuai dengan yang diharapkan. dibuktikan dengan pengujian <i>whitebox</i> dan <i>blackbox</i> ditambah dengan survei menggunakan kuisioner ke pengguna. Hasil pengujian <i>whitebox</i> dengan menggunakan metode <i>Cyclomatic Complexity (CC)</i> membuktikan bahwa sistem telah benar dan tidak memiliki kesalahan baik dari segi logika maupun fungsi. Hasil pengujian <i>blackbox</i> membuktikan proses aplikasi yang dilakukan sudah benar sehingga dapat menghasilkan <i>output</i> yang diharapkan.	Pada penelitian yang dilakukan oleh Kamaruddin Tone adalah berfokus pada merancang dan membuat aplikasi pengingat kelas (<i>class remainder</i>) berbasis <i>Android</i> yang sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder jadwal perkuliahan</i> menggunakan metode pengembangan <i>Extreme Programming</i> .

No.	Penulis	Tahun	Judul	Metode	Hasil Penelitian	Perbedaan
11.	Adi Supriyatna	2018	Metode <i>Extreme Programming</i> Pada Pembangunan Web Aplikasi Seleksi Peserta	<i>Extreme Programming</i>	aplikasi ujian seleksi peserta pelatihan kerja dapat memberikan kemudahan kepada masyarakat terkait informasi pelaksanaan pelatihan dan <i>Metode Extreme Programming (XP)</i> dapat digunakan untuk membangun atau mengembangkan perangkat lunak.	Pada penelitian yang dilakukan oleh Adi Supriyatna berfokus pada penerapan metode <i>Extreme Programming</i> untuk membung aplikasi ujian seleksi peserta pelatihan kerja, sedang kanpenulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder</i> jadwal perkuliahan menggunakan metode pengembangan <i>Extreme Programming</i> .
12.	Imam Rohmat Ahmad, Indra Borman, Fakhruozi, dan Gavan Gorbi Caksana.	2020	Software Development Dengan <i>Extreme Programming (XP)</i> Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis <i>Android</i> .	<i>Extreme Programming</i>	Aplikasi yang dibangun dapat melakukan pencocokan kemiripan judul skripsi dengan menggunakan pendekatan <i>Booyer Moore</i> dan pengembangan sistem menggunakan metode <i>XP</i> telah menghasilkan <i>software</i> yang sesuai dengan kebutuhan.	Pada penelitia yang dilakukan oleh Imam Ahmad, DKK adalah fokus pada penerapan metode <i>Extreme Programming</i> pada aplikasi deteksi kemiripan judul skripsi, sedangkan penulis berfokus pada penerapan arsitektur <i>MVVM</i> pada aplikasi <i>reminder</i> jadwal perkuliahan menggunakan metode pengembangan <i>Extreme Programming</i> .

2.3 Landasan Teori

2.3.1 *Android*



Gambar 2.1. *Android*

Sumber : www.android.com diakses pada september 2021

Android merupakan *operating system*, *middleware*, dan *mobile application* yang berbasis pada *Linux Kernel*, dibangun oleh *Google* dan *Open Handset Alliance*. *Android* sistem operasi *mobile* yang memiliki sifat *open source*. Google memiliki tujuan untuk mengembangkan produk *real-world* pengalaman para pengguna dapat meningkat, itu sebabnya *Android* bersifat *open source project*. *Android* dibangun dengan memanfaatkan *SDK* (*Software Development Kit*) secara komprehensif menggunakan alat yang cukup untuk membangun *software* yang kaya akan fitur dan *powerful*. (Ahmad et al., 2020)

2.3.2 *Android Studio*



Gambar 2.2. *Android Studio*

Sumber : www.android-developers.googleblog.com diakses pada september 2021

Android Studio adalah *IDE* dari *Google* untuk aplikasi *Android*. *Android Studio* menyediakan *editor* kode tingkat lanjut dan serangkaian template aplikasi. Selain itu, *Android Studio* juga memiliki alat untuk *development*, men-*debug*, menguji, dan meningkatkan kinerja yang membuat pengembangan aplikasi lebih cepat dan mudah. Pengguna bisa menguji aplikasi dengan berbagai macam *emulator* yang telah dikonfigurasi sebelumnya atau dengan perangkat seluler pengguna sendiri. (Developer Training Team, 2016)

2.3.3 *Kotlin*

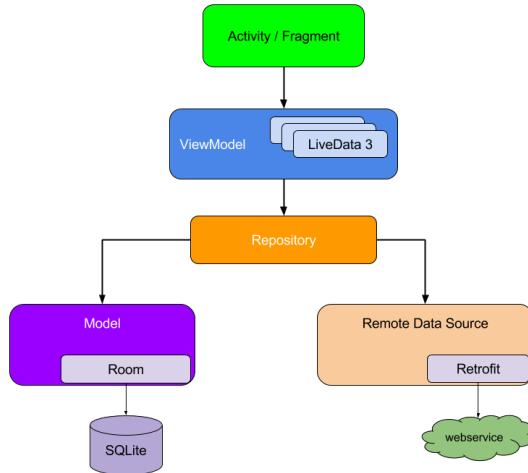


Gambar 2.3. *Kotlin*

Sumber : www.kotlinlang.org diakses pada september 2021

Kotlin adalah bahasa pemrograman bertipekan statis yang berjalan di *Java Virtual Machine* dan juga dapat dikompilasi ke kode sumber *JavaScript*. *Kotlin* dirilis ke publik pada Februari 2016. Dalam perusahaan pengembangan *Software* yang berasal dari tim berbasis *programmer JetBrains* di Saint Petersburg, Rusia. Bahkan, nama *Kotlin* datang dari Pulau Kotlin di Saint Petersburg. (Bose et al., 2018)

2.3.4 MVVM Design Pattern

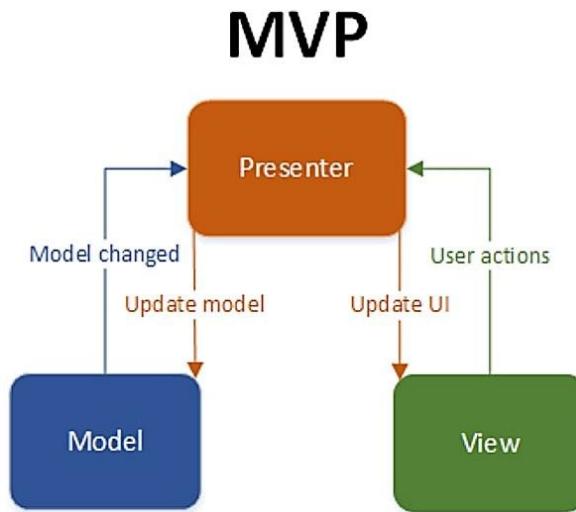


Gambar 2.4. MVVM Design Pattern

Sumber : developer.android.com diakses pada september 2021

MVVM adalah singkatan dari *Model View ViewModel*. *MVVM* adalah pola arsitektur yang tujuan utamanya adalah untuk memisahkan antara komponen aplikasi dengan komponen lain sehingga pada komponen memiliki peran masing-masing. dimana ada tiga komponen yang memiliki tugas atau tanggung jawab masing-masing seperti *Model* menyediakan sumber data yang dibutuhkan. Data bisa didapatkan baik dari *server* maupun dari *database* lokal menggunakan *SQLite*. *View* memiliki tanggung jawab untuk menampilkan data pada user *interface* dan menginformasikan lapisan lain tentang tindakan pengguna. *ViewModel* memiliki tanggung jawab untuk mengambil informasi yang diperlukan dari *Model* dan operasi yang diperlukan dalam memaparkan data untuk ditampilkan pada *View*. (Cheng & Domínguez, 2019)

2.3.5 MVP Design Pattern



Gambar 2.5. *MVP Design Pattern*

Sumber : www.researchgate.net diakses pada september 2021

Arsitektur *MVP* (*Model View Presenter*) memiliki persamaan dengan arsitektur *MVC*. seperti model adalah layer yang digunakan untuk *business logic* seperti mengambil data, menyimpan data dan mengubah data. *View* digunakan untuk menampilkan *UI*, dengan *MVP* peran ini secara khusus ditujukan untuk Aktivitas atau *Fragment*. *Presenter* adalah kelas penghubung antara *Model* dan *View*. Semua Kode yang tidak secara langsung menangani *UI* atau logika khusus kerangka kerja *Android* lainnya harus dipindahkan keluar dari *View* dan masuk ke *class Presenter*. (Cheng & Domínguez, 2019)

2.3.6 PHP



Gambar 2.6. PHP

Sumber : www.php.net diakses pada september 2021

PHP singkatan dari *PHP:Hypertext Preprocessor* adalah bahasa pemrograman *web server-side* yang bersifat *open source*. *PHP* merupakan *script* yang terintegrasi dengan *HTML* dan berada pada *server* (*server side HTML embeded scripting*). *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script* *PHP* dieksekusi pada *server* dimana *script* tersebut dijalankan. (Anhar, 2010)

2.3.7 Laravel Framework

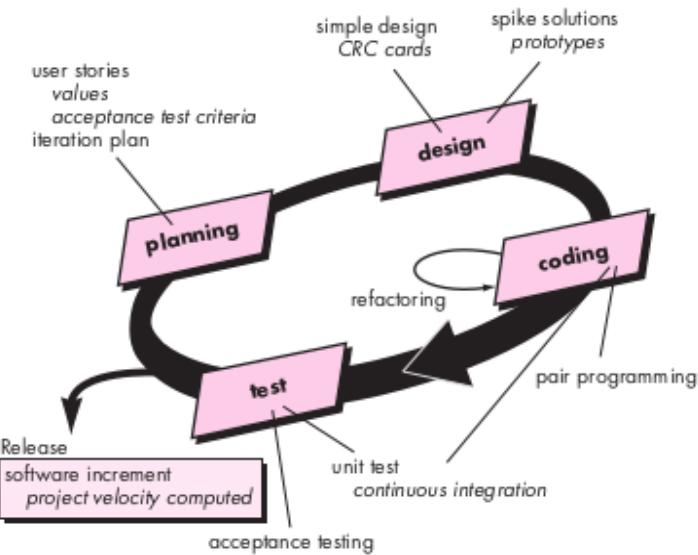


Gambar 2.7. *Laravel Framework*

Sumber : laravel.com diakses pada september 2021

Laravel merupakan *framework PHP* yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. *Laravel* dirilis dibawah lisensi *MIT* dengan sumber kode yang disediakan di *Github*. Sama seperti *framework PHP* lainnya, *Laravel* dibangun dengan basis *MVC (Model- View-Controller)*. *Laravel* dilengkapi command line tool yang bernama “Artisan” yang bisa digunakan untuk *packaging bundle* dan instalasi *bundle*. Menurut survey yang lakukan oleh Sitepoint.com pada Desember 2013 dalam popularitas *framework PHP*, *Laravel* menduduki urutan teratas. (Erinton, R. Negara, R. Sanjoyo, 2017)

2.3.8 *Extreme Programming*



Gambar 2.8. *Extreme Programming*

Sumber : Pressman, 2010

Menurut Pressman (Suryn, 2014) XP atau *Extreme Programming* adalah pendekatan pengembangan yang paling banyak digunakan dalam pengembangan perangkat lunak dengan pengembangan cepat. Adapun 4 tahapan metode pendekatan *Extreme Programming* adalah sebagai berikut :

1. *Planning* (Perencanaan)

Perencanaan yaitu suatu kegiatan yang bertujuan untuk mengumpulkan kebutuhan-kebutuhan untuk perangkat lunak yang akan dikembangkan untuk mendapatkan *output*, fitur-fitur utama, dan fungsionalitas.

a. *User Stories*

Pembuatan serangkaian “cerita” yang menggambarkan keluaran yang diperlukan, fitur-fitur, dan fungsionalitas- fungsionalitas yang akan dibangun pada perangkat lunak yang akan dikembangkan.

b. Values

Suatu nilai prioritas yang diberikan pada cerita-cerita tertentu berdasarkan seluruh nilai bisnis dari fitur atau fungsi.

c. Acceptance Criteria

Menjelaskan ruang lingkup dari sebuah *user stories* yang berupa daftar kriteria yang mengindikasikan sebuah *story* sudah diselesaikan.

d. Iteration Plan

Merencanakan apa yang akan dilakukan pada setiap iterasi untuk mengetahui perubahan pada kebutuhan pengguna.

2. Design (Perancangan)

Menurut pressman perancangan yang sederhana itu lebih banyak disukai dari pada gambaran-gambaran yang lebih kompleks. Dalam perancangan *XP* akan memberikan panduan penerapan untuk suatu cerita ketika cerita itu ditulis, tidak kurang, tidak lebih. Rancangan-rancangan dan fungsionalitas-fungsionalitas tambahan akan dianggap tidak disarankan oleh pengembang.

a. Simple Design

Sebuah desain sederhana pada perancangna perangkan lunak selalu membutuhkan waktu lebih sedikit untuk menyelesaikan daripada yang kompleks. Jadi selalu lakukan hal paling sederhana yang mungkin bisa berhasil selanjutnya.

b. CRC Card

Kartu yang digunakan untuk merancang sistem. dengan menyediakan pemodelan sederhana untuk mengidentifikasi dan mengatur kelas yang relevan dengan sistem atau kebutuhan sistem.

c. *Spike Solution Prototype*

Program atau desain program yang sangat sederhana yang digunakan untuk mengekplorasi solusi potensial. ketika menemui masalah desain yang sulit.

3. *Coding (Penerapan)*

Pengkodean dikembangkan setelah rancangan awal dilakukan, tetapi tidak langsung untuk melakukan pengkodean program melainkan membuat sebuah pengujian unit. Dalam pengkodean ini menggunakan metode *XP* menyarankan untuk dua orang pemrogram harus berkerja dalam satu *workstation*.

a. *Refactoring*

Refactoring dilakukan untuk menjaga agar kode tetap bersih dan ringkas sehingga lebih mudah untuk dipahami, dimodifikasi, dan diperluas. dan untuk menghindari kekacauan dan kerumitan yang tidak perlu.

b. *Pair Programming*

XP merekomendasikan agar dua orang bekerja bersama dalam satu waktu pada komputer workstation dalam membuat kode untuk sebuah *user stories*. Ini bertujuan menyediakan mekanisme untuk pemecahan masalah waktu dan penilaian kualitas kode yang ditinjau saat kode ditulis.

c. *Continous Integration*

Integrasi secara berkelanjutan dilakukan untuk menghindari upaya pengembangan yang menyimpang atau terfragmentasi, di mana pengembang tidak berkomunikasi satu sama lain tentang apa yang dapat digunakan kembali.

4. Testing (Pengujian)

Pengujian unit harus dibuat dan kemudian dijalankan menggunakan framework yang memungkinkan mereka untuk diotomatisasi dan dapat dijalankan dengan mudah secara berulang kali. Pengujian ini pada dasarnya berfokus pada fitur-fitur dan fungsionalitas pada sistem.

a. Unit Testing

Pengujian unit bertujuan melakukan verifikasi pada unit terkecil dari modul perangkat lunak pada sistem yang telah dikembangkan. Dimana pada pengujian unit berfokus pada pengujian logika pemrosesan internal dan struktur data.

b. Acceptance Testing

Dilakukan untuk memastikan hasil dari sistem yang telah dikembangkan telah sesuai dengan kebutuhan pengguna. Dimana pada pengujian ini menggunakan metode *black box testing* dan pengguna bertanggung jawab untuk memverifikasi kebenaran tes penerimaan dan meninjau nilai tes untuk memutuskan tes sesuai atau belum sesuai.

5. Release (Perilisan)

Setelah semua tahap pengembangan perangkat lunak dilakukan, tahap selanjutnya adalah melakukan perilisan sistem. Pada tahap ini sistem sudah dalam bentuk final dari sistem yang sudah dikembangkan dan siap untuk digunakan oleh pengguna.

2.3.9 MySQL



Gambar 2.9. MySQL

Sumber : www.mysql.com diakses pada september 2021

MySQL (My Structure Query Language) adalah sebuah perangkat lunak sistem menejeman basis data *SQL (Database Management System)* atau *DBMS* dari sekian *DBMS*, seperti *Oracle*, *MS SQL*, *Postagre SQL*, dan lain-lain. *MySQL* merupakan *DBMS multithread*, multi-user yang gratis dengan lisensi *GNU General Public License (GPL)*. *MySQL support* dengan Bahasa pemrograman *PHP*. (Anhar, 2010)

BAB III

METODOLOGI PENELITIAN

3.1 Deskripsi Sistem

Pada penelitian ini penulis menggunakan data dan alat penelitian yang digunakan dalam penelitian ini. Adapun data dan alat penelitian adalah sebagai berikut :

3.1.1 Data

Dalam proses pembuatan aplikasi *reminder* mata kuliah ini penulis membutuhkan data untuk mendukung dalam proses analisis dan perancangan sistem. Adapun data yang dibutuhkan yaitu, *file excel* jadwal perkuliahan, data mata kuliah, data dosen, data ruangan.

1. Data Jadwal Perkuliahan

Gambar 3.1. File Jadwal Perkuliahan
Sumber : Politeknik Negeri Bengkalis

2. Data Mata Kuliah

The screenshot shows two tables of course data side-by-side in Microsoft Excel. The left table is for Semester I (TA 2021/2022) and the right table is for Semester II (TA 2021/2022). Both tables include columns for course code, name, credits (SKS), theory/practical hours, room assignments, and teaching staff.

SEMESTER I	MataKuliah	SKS	JAM/MINGGU		DOSEN PENGAMPU MATAKULIAH	RUANGAN	
			Theor	Praktikum		KELAS 1A	KELAS 1B
1 KSD0101	Aplika	3	3	3	Jurnat, M.Pd	Ruang Kelas	
2 KSD0102	Percoba	2	2	2	Jon Hendi, M.Pd	Ruang Kelas	
3 KSD0103	Analisa dan Desain Sistem	2	2	2	Nurmi Hidayah, M.Kom	Ruang Kelas	
4 KSD0104	Matematika Terapan	3	3	3	An Salwa, M.Pd	Ruang Kelas	
5 KSD0105	Algoritma TK	2	2	2	Arta Deleininggah, M.Pd	Ruang Kelas	
6 KSD0106	Pengantar Keamanan Informati	2	2	2	Agus Tedyana, M.Kom	Lab Komputer Dasar	
7 KSD0107	Struktur dan Pemrograman	2	2	2	Darus, M.Kom	Lab Keamanan Informati	
8 KSD0108	Pemrograman Lanjut	2	2	2	Mansur, M.Kom	Lab Pemrograman Lanjut	
9 KSD0109	Pemrograman dan Pemrograman	2	2	2	Darus, M.Kom	Lab Keamanan Informati	
10 KSD0110	Algoritma dan Pemrograman	2	2	2	Jasa, M.Kom	Lab Pemrograman Dasar	
		20	14	8	14	16	29

SEMESTER II	MataKuliah	SKS	JAM/MINGGU		Rincinan Dosen Pengampu	RUANGAN	
			Theor	Praktikum		Jam	KELAS 3A
1 KSD0201	Analisa dan Pengembangan Sistem Informati	3	1	2	Agus Tedyana, M.Kom	KELAS 3A	Dosen Individu
2 KSD0202	Teknologi	3	1	2	Nurmi Hidayah, M.Kom	KELAS 3A	Lab Keamanan Informati
3 KSD0203	Keamanan Jaringan Komputer	2	0	2	0	Kelarangan *	Lab Basis Data
4 KSD0204	Analisa dan Desain Sistem	2	2	2	Arta Deleininggah, M.Kom	KELAS 3A	Lab Jaringan Komputer
5 KSD0205	Pemrograman Lanjut	3	1	2	Agus Tedyana, M.Kom	Ruang Kelas	Dosen Individu
6 KSD0206	Pengembangan Sistem Data	2	2	2	Darus, M.Kom	KELAS 3A	Lab Pemrograman Lanjut
7 KSD0207	Suhu dan Angin	2	2	2	Mansur, M.Kom	Ruang Kelas	Dosen Individu
8 KSD0208	Algoritma dan Pemrograman	2	2	2	Jon Hendi, M.Pd	Ruang Kelas	Ruang Kelas
		19	9	10	9	20	34

*MK yang dapat di ajarkan oleh Dosen Individu

Gambar 3.2. File Data Mata Kuliah
Sumber : Politeknik Negeri Bengkalis

3.1.2 Alat

Alat yang digunakan untuk melakukan penelitian ini meliputi perangkat keras dan perangkat lunak. Adapun alat yang digunakan adalah sebagai berikut :

1. Spesifikasi perangkat keras yang digunakan

a. Laptop Lenovo Thinkpad T430

- *Processor* : Intel i5-3320M (4) @ 2.60GHz
- *RAM* : 12 GB
- *OS* : Linux Mint 20.3 Una x86_64
- *System Type* : 64-bit Operating System

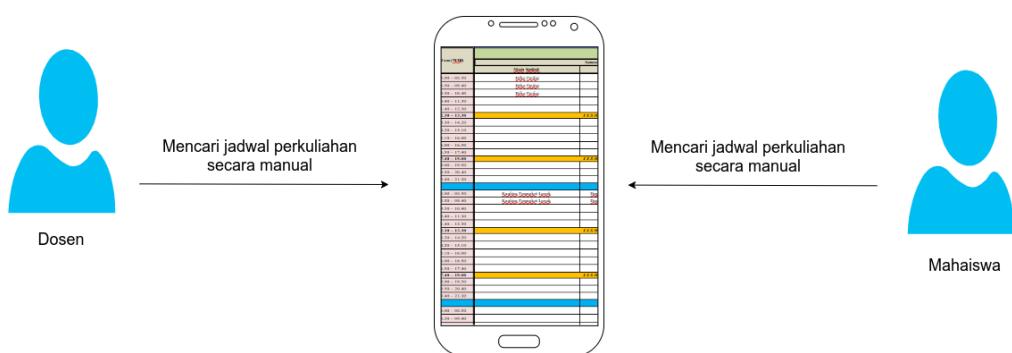
- b. Smartphone Sony Experia Z5 Compact
 - *OS* : *Android 7.0 Nougat*
 - *RAM* : *2 GB*
 - *CPU* : *Qualcomm Snapdragron 810 @2.00 GHz*
2. Perangkat lunak yang digunakan
 - a. Pengolah Kata : *Libreoffice 6.4*
 - b. *Browser* : *Google Chrome, Mozilla Firefox*
 - c. *IDE* : *Android Studio, IntelliJ*
 - d. *Text Editor* : *Visual Studion Code*
 - e. *Web Server* : *Apache Web Server*
 - f. Bahasa Pemrograman : *Kotlin, PHP*
 - g. *Database* : *Room Database, MySQL*

3.1.3 Analisa Kebutuhan

Sebelum menganalisis kebutuhan fungsional dan kebutuhan non fungsional terlebih dahulu melihat gambaran sistem yang ada dan gambaran sistem yang akan diusulkan pada penelitian ini.

a. Sistem Yang Sedang Berjalan

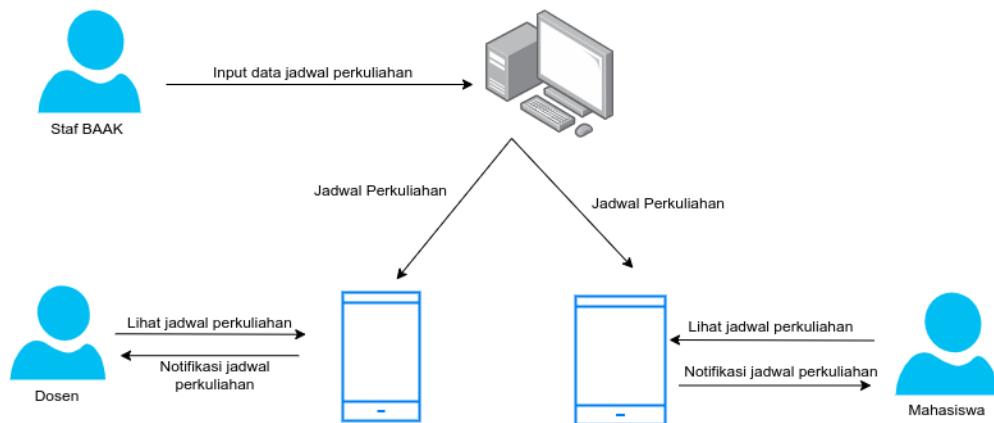
Saat ini permasalahan yang dialami oleh mahasiswa adalah kesulitan pada saat membuka *file excel*, dan sulit mencari jadwal perkuliahan yang akan diikuti karena semua jadwal perkuliahan digabung menjadi satu *file*. akibat nya sebagian mahasiswa sering men-*screenshot* jadwal perkuliahan yang akan diikuti dan menjadikanya sebagai *wallpaper* ponsel.



Gambar 3.3. Sistem Yang Sedang Berjalan
Sumber : Data Olahan

b. Sistem Yang Diusulkan

Pada sistem yang akan diusulkan, disini sistem dapat mempermudah, bagi dosen dan mahasiswa dalam melihat jadwal perkuliahan, karena pengguna cukup melakukan *request* data jadwal perkuliahan diaplikasi dan aplikasi dapat dapat menampilkan jadwal perkuliahan dan memberikan notifikasi jadwal perkuliahan.



Gambar 3.4. Sistem Yang Diusulkan

Sumber : Data Olahan

Setelah proses pengumpulan data selesai dilakukan dengan menganalisis sistem yang sedang berjalan maka didapat kebutuhan fungsional dan kebutuhan non fungsional sistem sebagai berikut :

3.1.4 Kebutuhan Fungsional

Tabel 3.1. Kebutuhan Fungsional Menu Pilih kelas

Nama Fungsi	Menu pilih kelas
Prioritas	<i>High</i>
Pemicu	Aplikasi dijalankan pertama kali
Kondisi Awal	Menampilkan menu <i>drop down</i>
Alur Normal	Pengguna memilih tipe pengguna Mahasiswa
Alur Alternatif	-
Kondisi Akhir	Menampilkan menu daftar kelas mahasiswa
Pengecualian	-

Tabel 3.2. Kebutuhan Fungsional Menu Pilih Prodi

Nama Fungsi	Menu pilih prodi
Prioritas	<i>High</i>
Pemicu	Aplikasi dijalankan pertama kali
Kondisi Awal	Menampilkan menu <i>drop down</i>
Alur Normal	Pengguna memilih tipe pengguna Mahasiswa
Alur Alternatif	-
Kondisi Akhir	Menampilkan menu daftar program studi
Pengecualian	-

Tabel 3.3. Kebutuhan Fungsional Menu Nama Dosen

Nama Fungsi	Menu nama dosen
Prioritas	<i>High</i>
Pemicu	Aplikasi dijalankan pertama kali
Kondisi Awal	Menampilkan menu <i>drop down</i>
Alur Normal	Pengguna memilih tipe pengguna Dosen
Alur Alternatif	-
Kondisi Akhir	Menampilkan menu daftar nama dosen
Pengecualian	-

Tabel 3.4. Kebutuhan Fungsional Menu Toggle Reminder

Nama Fungsi	<i>Toggle Reminder</i>
Prioritas	High
Pemicu	-
Kondisi Awal	Pengguna berhasil Login
Alur Normal	Semua jadwal perkuliahan memiliki reminder yang secara default statusnya susah aktif.
Alur Alternatif	-
Kondisi Akhir	Menampilkan toggle status reminder pada jadwal perkuliahan
Pengecualian	-

Tabel 3.5. Kebutuhan Fungsional Menu Mata Kuliah Sekarang

Nama Fungsi	Mata Kuliah Sekarang
Prioritas	High
Pemicu	Sudah memasuki waktu jadwal perkuliahan
Kondisi Awal	Pengguna berhasil Login
Alur Normal	Aplikasi akan menampilkan mata kuliah yang sedang berlangsung berdasarkan waktu yang telah ditentukan pada jadwal perkuliahan dan waktu tersebut sudah memasuki masanya.
Alur Alternatif	-
Kondisi Akhir	Menampilkan mata kuliah yang sedang berlangsung atau sudah memasuki waktunya.
Pengecualian	-

Tabel 3.6. Kebutuhan Fungsional Menu Atur Waktu Notifikasi

Nama Fungsi	Atur Waktu Notifikasi
Prioritas	High
Pemicu	Pengguna memilih menu Atur Waktu Notifikasi
Kondisi Awal	Pengguna berhasil Login
Alur Normal	Aplikasi akan menampilkan menu pilihan waktu delay untuk menampilkan notifikasi
Alur Alternatif	Delay waktu notifikasi default adalah 5 menit sebelum masuk waktunya
Kondisi Akhir	Menampilkan halaman yang memiliki pilihan delay waktu notifikasi
Pengecualian	-

Tabel 3.7. Kebutuhan Fungsional Notifikasi Jadwal Perkuliahan

Nama Fungsi	Notifikasi Jadwal Perkuliahan
Prioritas	High
Pemicu	Sudah memasuki waktu jadwal perkuliahan
Kondisi Awal	Pengguna berhasil Login
Alur Normal	Aplikasi akan menampilkan notifikasi apabila pada jadwal perkuliahan telah memasuki waktunya.
Alur Alternatif	-
Kondisi Akhir	Menampilkan notifikasi yang memiliki informasi tentang jadwal perkuliahan yang akan berlangsung
Pengecualian	-

Tabel 3.8. Kebutuhan Fungsional Daftar Jadwal Perkuliahuan

Nama Fungsi	Daftar Jadwal Perkuliahuan
Prioritas	High
Pemicu	Pengguna Login
Kondisi Awal	Pengguna berhasil Login
Alur Normal	Pengguna melakukan login pada aplikasi dan memilih menu yang ada pada halaman login, selanjutnya aplikasi akan menampilkan daftar jadwal perkuliahan sesuai dengan tipe pengguna.
Alur Alternatif	Apabila pengguna sudah login sebelumnya maka aplikasi akan langsung menampilkan jadwal perkuliahan
Kondisi Akhir	Halaman yang menampilkan jadwal perkuliahan
Pengecualian	-

3.1.5 Kebutuhan Non-Fungsional

1. Sistem membutuhkan jaringan internet saat *request* data jadwal perkuliahan.
2. Sistem dapat menjadwalkan ulang *reminder* ketika perangkat selesai *reboot*.

3.2 Alur Metode *Extreme Programming*

Dalam pengembangan perangkat lunak terdapat beberapa metode pengembangan yang dapat digunakan, pada penelitian ini penulis menggunakan metode *Extreme Programming (XP)* untuk mengembangkan aplikasi reminder jadwal perkuliahan. Pada metode *Extreme Programming (XP)* terdapat empat tahapan yang harus dikerjakan yaitu :

3.2.1 *Planning* (Perencanaan)

a. Pengumpulan Data

Pada tahapan pengumpulan data penulis melakukan kegiatan dalam bentuk wawancara, observasi dan Penyebaran Kuisisioner. Adapun Kegiatan nya adalah sebagai berikut :

- Wawancara

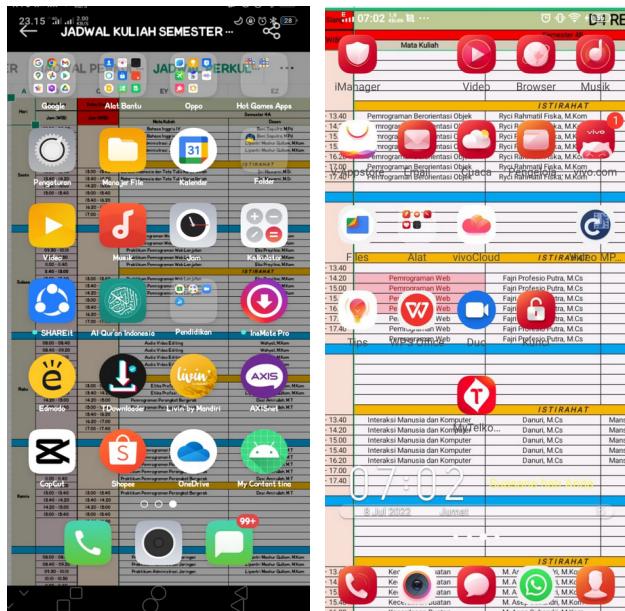
Pada tahap wawancara ini dilakukan proses pengumpulan informasi dan mencari data apa yang dibutuhkan dan permasalahan yang dialami oleh Dosen dan Mahasiswa.

Tabel 3.9. Pertanyaan Wawancara

No.	Pertanyaan
1.	Apakah anda kesulitan saat melihat jadwal perkuliahan dari file excel jadwal perkuliahan?
2.	Permasalahan apa yang anda alami ketika melihat jadwal perkuliahan dari file excel?
3.	Apa cara anda untuk mempermudah saat melihat jadwal perkuliahan yang akan diikuti lagi?
4.	Apa saran anda yang dapat diberikan untuk mengatasi permasalahan terkait file excel jadwal perkuliahan tersebut?

- Observasi

Pada proses observasi penulis melakukan pengamatan tentang permasalahan yang dialami oleh Mahasiswa.



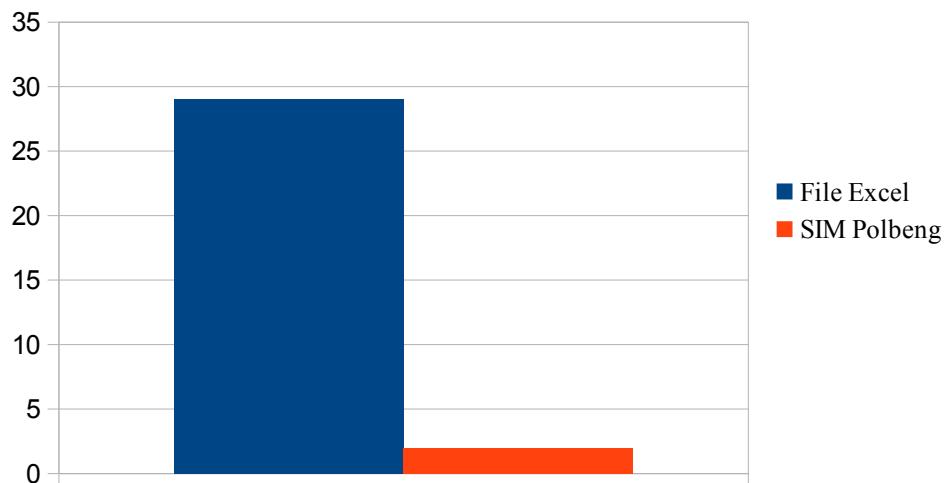
Gambar 3.5. Sampel Observasi Permasalahan

Sumber : Data Olahan

- Kuisioner

Dilakukan pengumpulan data melalui penyebaran kuisioner kepada mahasiswa di Poltekne Negeri Bengkalis yang telah dibuat sebelumnya dengan menggunakan Google *form*. Data yang diperoleh dari kuisioner ini kemudian diolah dan dianalisis, sehingga dapat ditampilkan sebagai berikut :

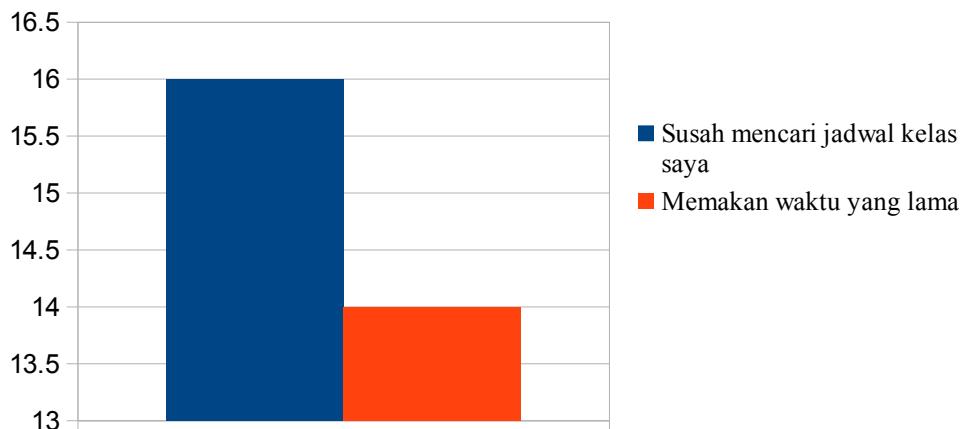
Dari mana anda biasanya melihat jadwal perkuliahan yang akan diikuti?



Gambar 3.6. Data Diagram Kuisioner Pertanyaan 1
Sumber : Data Olahan

Dapat dilihat dari grafik diatas, berdasarkan pertanyaan dan opsi jawaban yang telah disediakan pada form kuisioner sebanyak 30 orang mahasiswa yang ikut serta berpartisipasi dalam kuisioner ini rata-rata mahasiswa menjawab melihat jadwal perkuliahan yang akan diikuti dari *file excel* jadwal perkuliahan dari pada melihat dari SIM Polbeng.

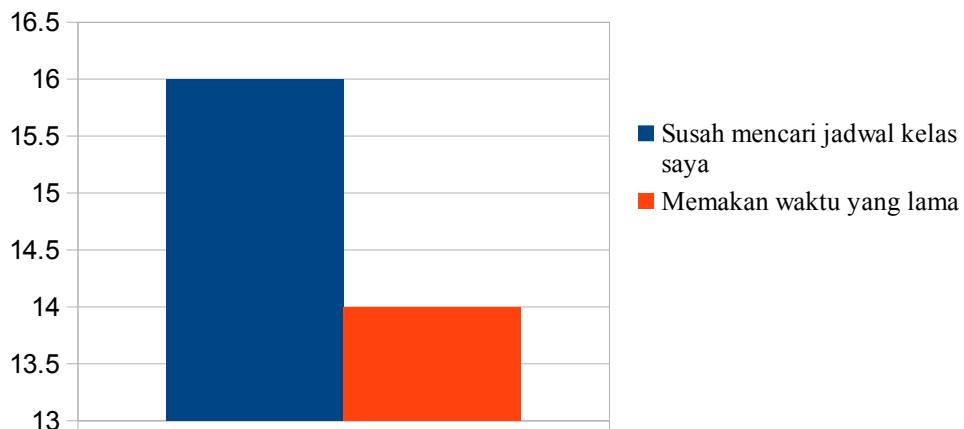
Permasalahan apa yang anda alami ketika melihat jadwal perkuliahan dari dari file excel?



Gambar 3.7. Data Diagram Kuisioner Pertanyaan 2
Sumber : Data Olahan

Pada grafik diatas berdasarkan pertanyaan dan pilihan jawaban yang telah disediakan pada form kuisioner sebanyak 16 orang mahasiswa mengalami kesusahan dalam mencari jadwal perkuliahan yang akan diikuti pada *file excel* dan sebanyak 14 orang mahasiswa merasakan butuh waktu yang lama dalam mencari jadwal perkuliahan yang akan diikuti pada *file excel*.

Permasalahan apa yang anda alami ketika melihat jadwal perkuliahan dari dari file excel?



Gambar 3.8. Data Diagram Kuisioner Pertanyaan 2
Sumber : Data Olahan

Pada grafik diatas berdasarkan pertanyaan dan pilihan jawaban yang telah disediakan pada form kuisioner sebanyak 19 orang mahasiswa mengalami Masalahan jaringan saat akan membuka SIM Polbeng, selanjutnya terdapat 1 orang mengatakan bahwa jika pada jadwal perkuliahan yang ada pada SIM Polbeng tidak ada keterangan ruang dan jam perkuliahanya, selanjutnya terdapat sebanyak 9 orang mahasiswa merasakan pada saat login ke SIM Polbeng cukup memakan waktu yang lama, dan terakhir terdapat 1 orang mahasiswa yang tidak pernah melihat jadwal perkuliahan dari SIM Polbeng.

Dari hasil wawancara dan hasil kuisioner yang telah didapatkan dari mahasiswa Politeknik Negeri Bengkalis, penulis mendefenisikan kebutuhan sistem dalam bentuk user stories sebagai berikut :

b. *User Stories*

Tabel 3.10. *User Stories*

No.	User Stories	Acceptance Criteria	Values
1.	Sebagai mahasiswa, saya ingin dapat melihat daftar perkuliahan yang akan saya ikuti mudah dan cepat.	Terdapat fitur untuk menampilkan jadwal perkuliahan yang diikuti, sesuai dengan jurusan, prodi, dan kelas.	9
2.	Sebagai mahasiswa, saya ingin menerima notifikasi terkait jadwal kuliah yang akan saya ikuti.	Terdapat fitur notifikasi yang menampilkan informasi mata kuliah, yang akan diikuti	10
3.	Sebagai mahasiswa saya ingin, dapat mengatur delay waktu untuk menampilkan notifikasi jadwal perkuliahan	Terdapat menu untuk mengatur waktu menampilkan notifikasi	8
4.	Sebagai mahasiswa saya ingin dapat memilih data jadwal perkuliahan sesuai dengan kelas mahasiswa	Terdapat menu untuk request data jadwal perkuliahan mahasiswa	5
5.	Sebagai dosen saya ingin dapat melihat daftar perkuliahan yang akan saya laksanakan	Terdapat fitur untuk menampilkan daftar jadwal perkuliahan yang akan dilaksanakan dosen	9
6.	Sebagai dosen saya ingin menerima notifikasi terkait jadwal perkuliahan yang akan saya laksanakan	Terdapat fitur untuk notifikasi yang menampilkan informasi mata kuliah yang akan dilaksanakan	10
7.	Sebagai dosen saya ingin dapat mengatur delay waktu untuk menampilkan notifikasi jadwal perkuliahan	Terdapat menu untuk mengatur waktu menampilkan notifikasi	8
8.	Dosen dapat memilih data jadwal perkuliahan sesuai dengan nama dosen	Terdapat menu untuk request data jadwal perkuliahan mahasiswa	5
9.	Menu pilihan tipe pengguna bagi dosen dan mahasiswa	Terdapat menu untuk memilih tipe pengguna bagi dosen dan mahasiswa	3
10.	Sebagai mahasiswa saya ingin dapat melihat jadwal kuliah yang sedang berlangsung	Terdapat menu untuk menampilkan mata kuliah yang sedang berlangsung untuk mahasiswa	6
11.	Sebagai dosen saya ingin dapat melihat jadwal kuliah yang sedang berlangsung	Terdapat menu untuk menampilkan mata kuliah yang sedang berlangsung	6
12.	Sebagai dosen saya ingin dapat mematikan reminder	Terdapat menu untuk mematikan reminder jadwal dosen	7
13.	Sebagai mahasiswa saya ingin dapat mematikan reminder	Terdapat menu untuk mematikan reminder jadwal mahasiswa	7

c. *Prioritas*

Tabel 3.11. Prioritas

Iteration Plan	Feature	User	Value	Priority
1	Fitur menampilkan jadwal perkuliahan	Mahasiswa	9	<i>High</i>
	Fitur notifikasi jadwal perkuliahan yang akan diikuti	Mahasiswa	10	<i>High</i>
	Menu pengaturan delay waktu notifikasi	Mahasiswa	8	<i>High</i>
	Menu non aktifkan reminder mahasiswa	Mahasiswa	7	<i>High</i>
2	Fitur untuk menampilkan jadwal perkuliahan	Dosen	9	<i>High</i>
	Fitur notifikasi jadwal perkuliahan yang akan diikuti	Dosen	10	<i>High</i>
	Menu pengaturan delay waktu notifikasi	Dosen	8	<i>High</i>
	Menu non aktifkan reminder dosen	Dosen	7	<i>High</i>
3	Menu <i>request</i> data jadwal perkuliahan	Mahasiswa	5	<i>High</i>
	Menu <i>request</i> data jadwal perkuliahan	Dosen	5	<i>High</i>
	Menu pilihan tipe pengguna	Mahasiswa	3	<i>Low</i>
	Menu pilihan tipe pengguna	Dosen	3	<i>Low</i>
4	Menu lihat jadwal mata kuliah sekarang yang sedang berlangsung	Dosen	6	<i>High</i>
	Menu lihat jadwal mata kuliah sekarang yang sedang berlangsung	Mahasiswa	6	<i>High</i>

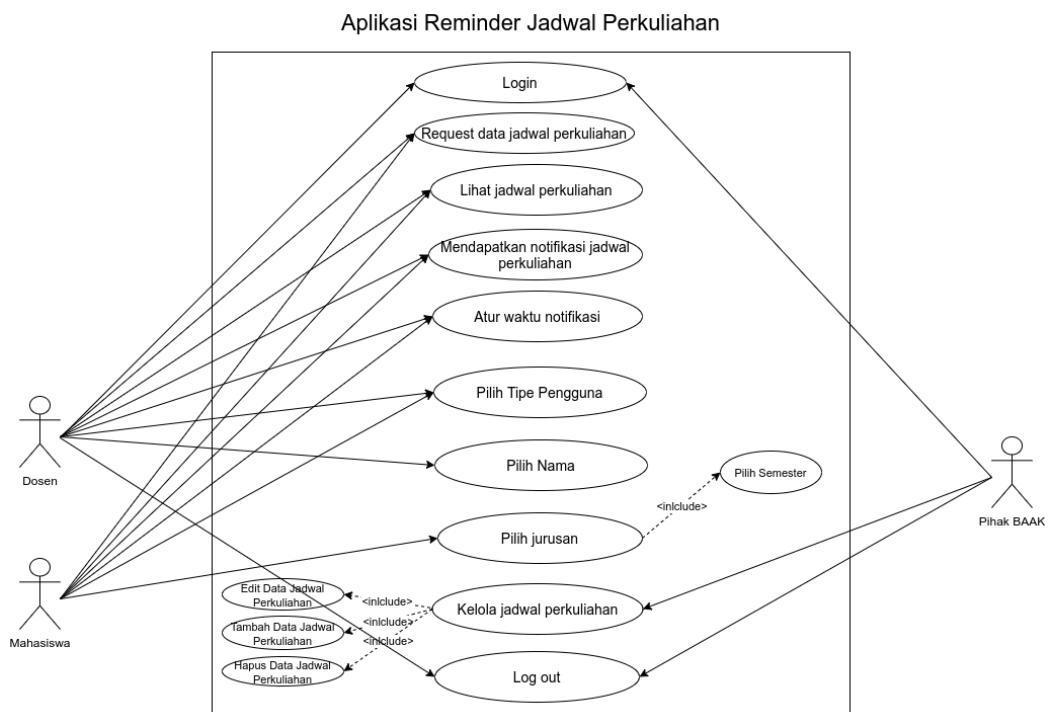
3.2.2 Design (Perancangan)

Selanjutnya pada tahap ini dilakukan kegiatan perancangan sistem, yang akan diterapkan pada aplikasi reminder jadwal perkuliahan dengan menggunakan *Use Case, Diagram Class, CRC Card* dan *Spike Solution Prototypes*.

c. Use Case

Adapun perancangan yang telah dilakukan adalah membuat diagram *use case*, Pada penelitian ini diagram use case memiliki 3 aktor yaitu :

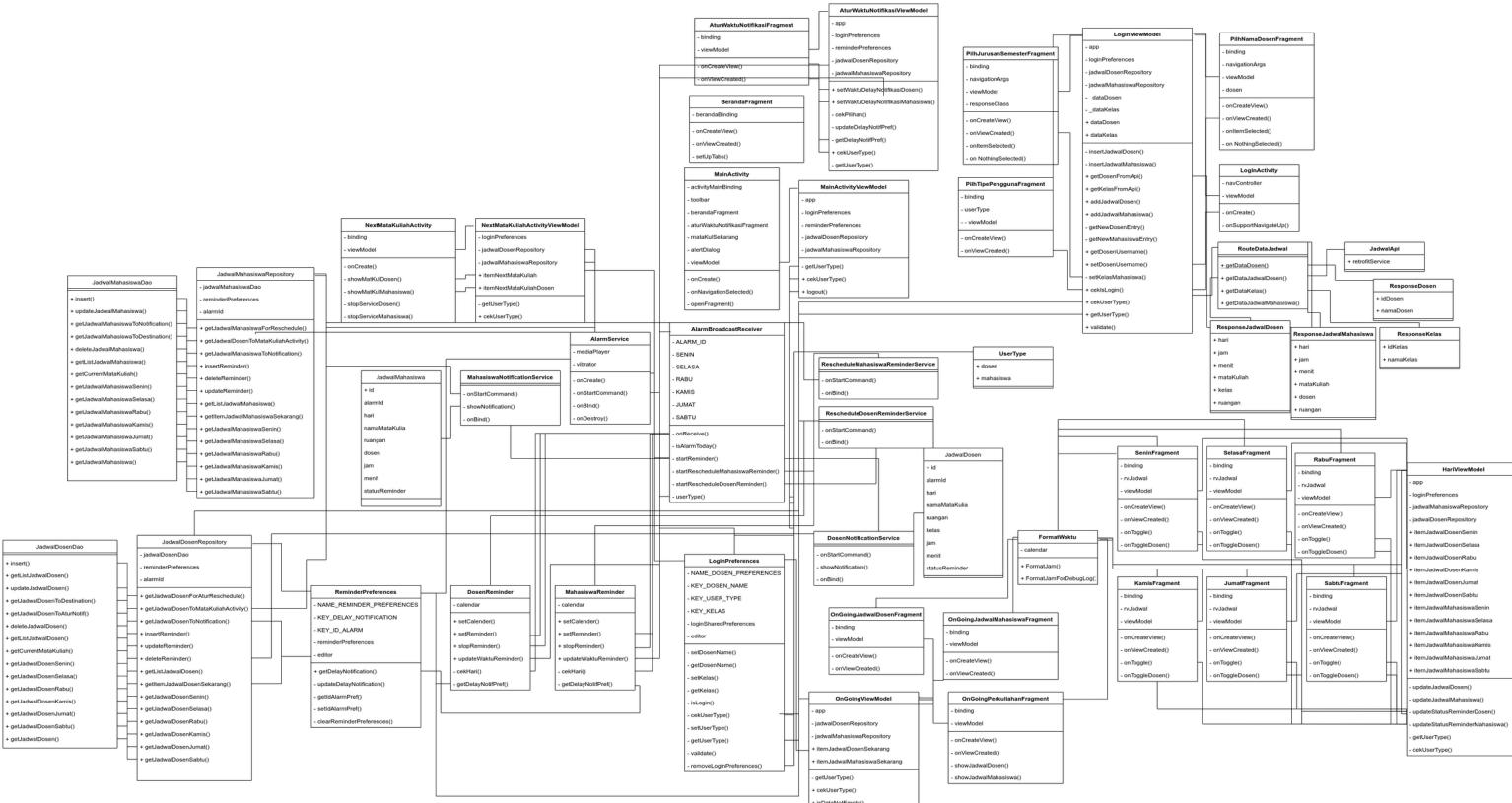
1. Dosen
2. Mahasiswa
3. Pihak BAAK



Gambar 3.9. *Use Case* Aplikasi Reminder Perkuliahan
Sumber : Data Olahan

b. *Diagram Class*

Struktur statis dari aplikasi *reminder* jadwal perkuliahan digambarkan menggunakan *diagram class* seperti gambar berikut.

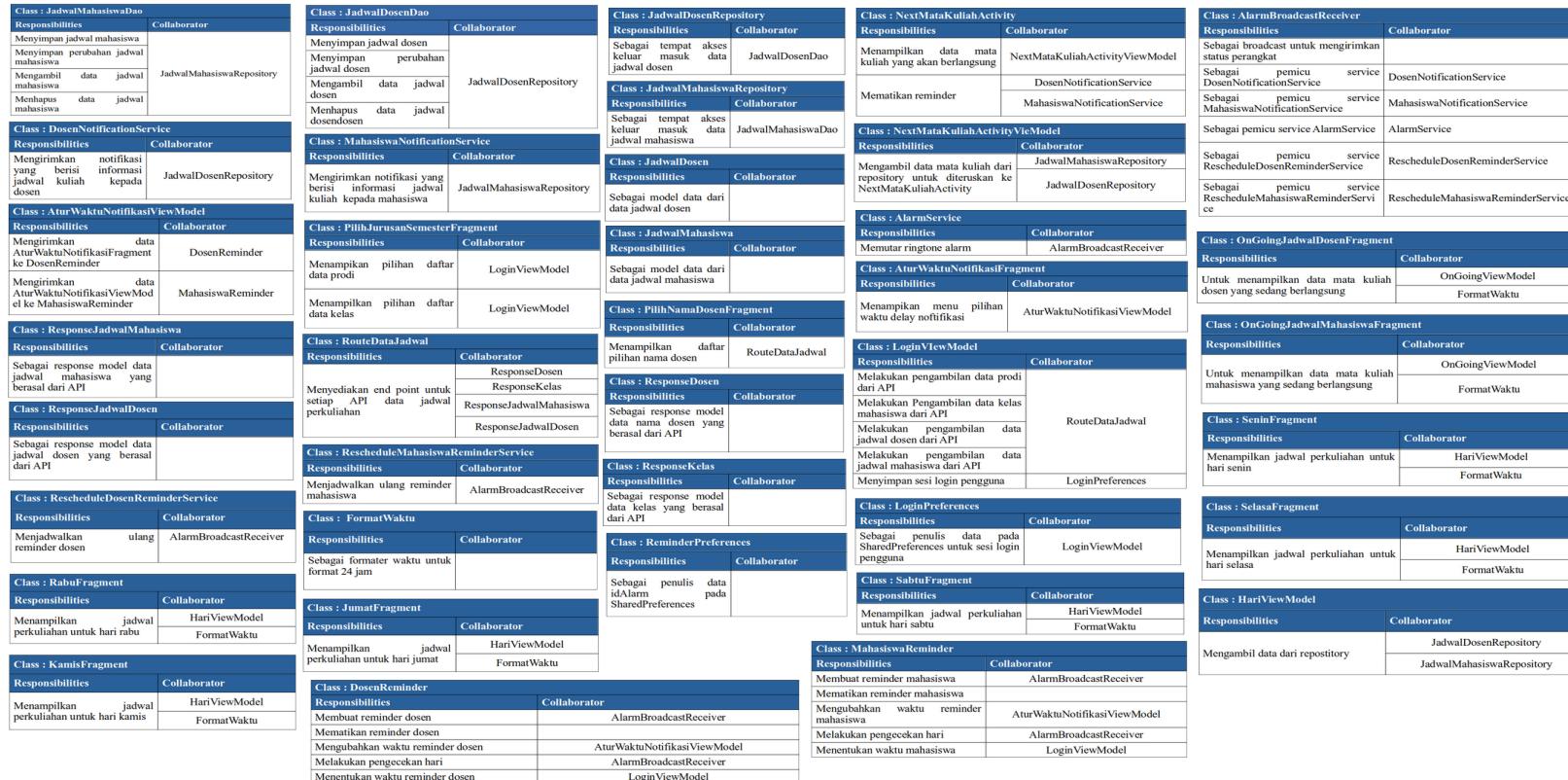


Gambar 3.10. *Class Diagram* Aplikasi Reminder Perkuliahan

Sumber : Data Olahan

c. *Collaboration Responsibilities Class (CRC Card)*

CRC Card diperoleh dari penjabaran hubungan antara *responsibility* antar *class* di sistem sesuai dengan yang digambarkan dalam *class diagram*, pada aplikasi reminder jadwal perkuliahan sebagai berikut.



Gambar 3.11. CRC Card Aplikasi Reminder Perkuliahan
Sumber : Data Olahan

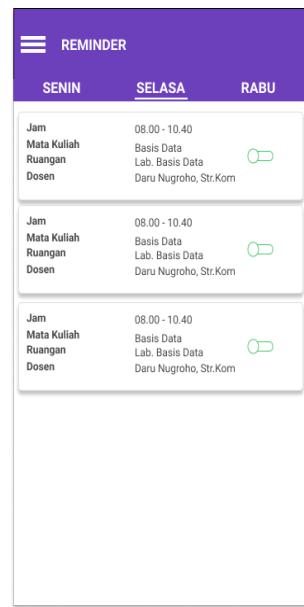
d. *Spike Solutions Prototype*

Berikut adalah *prototype* atau gambaran hasil dari aplikasi *reminder* jadwal perkuliahan yang telah di rancang dengan menggunakan aplikasi *figma*.

1. Iterasi Pertama



Gambar 3.12. *Prototype*
Notifikasi Mahasiswa



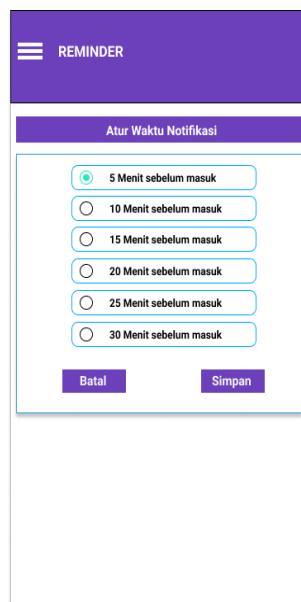
Gambar 3.13. *Prototype*
Jadwal Perkuliahan
Mahasiswa

Sumber : Data Olahan

Pada Gambar 3.12 untuk fitur notifikasi akan memunculkan notifikasi 5 menit sebelum perkuliahan dimulai dan didalam notifikasi tersebut menampilkan informasi tentang mata kuliah yang akan dimulai, ruangan kuliah, lama waktu kuliah, dan tombol untuk menghilangkan notifikasi.

Pada Gambar 3.13 tampilan jadwal perkuliahan untuk mahasiswa terdapat 7 tab yang menampilkan hari perkuliahan dari hari senin sampai hari sabtu. Pada setiap tab menampilkan daftar tentang mata kuliah sesuai dengan harinya. Dimana

pada setiap daftar mata kuliah menampilkan informasi seperti jam perkuliahan, nama mata kuliah, dan ruangan kuliah.



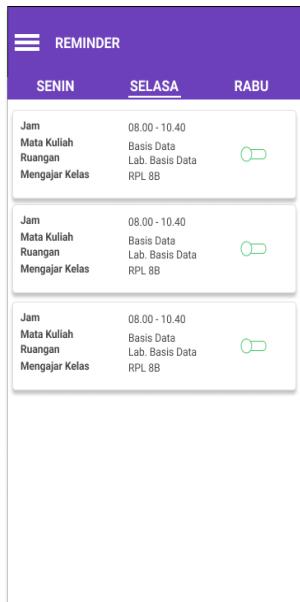
Gambar 3.14. *Prototype*

Atur Waktu Notifikasi

Sumber : Data Olahan

Pada Gambar 3.14 fitur untuk mengatur waktu memunculkan notifikasi untuk mahasiswa, terdapat beberapa pilihan waktu yang disediakan mulai dari 5 menit sampai 30 menit sebelum waktu masuk. Pada bagian bawah terdapat tombol batal untuk membatalkan pengaturan waktu dan tombol simpan untuk menyimpan pengaturan waktu notifikasi.

2. Iterasi Kedua



Gambar 3.15. Prototype Jadwal Perkuliahan Dosen



Gambar 3.16. Prototype Notifikasi Dosen

Sumber : Data Olahan

Pada Gambar 3.15 tampilan jadwal perkuliahan untuk dosen terdapat 7 tab yang menampilkan hari perkuliahan dari hari senin sampai hari sabtu. Pada setiap tab menampilkan daftar tentang mata kuliah sesuai dengan harinya. Dimana pada setiap daftar mata kuliah menampilkan informasi seperti jam perkuliahan, nama mata kuliah, ruangan kuliah dan akan mengajar dikelas yang mana.

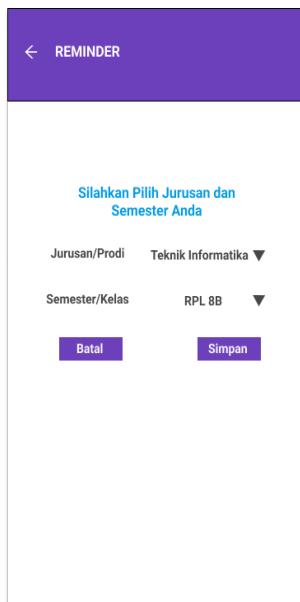
Pada Gambar 3.16 untuk fitur notifikasi akan memunculkan notifikasi 5 menit sebelum perkuliahan dimulai dan didalam notifikasi tersebut menampilkan informasi tentang mata kuliah yang akan dimulai, ruangan kuliah, lama waktu kuliah, dan tombol untuk menghilangkan notifikasi.



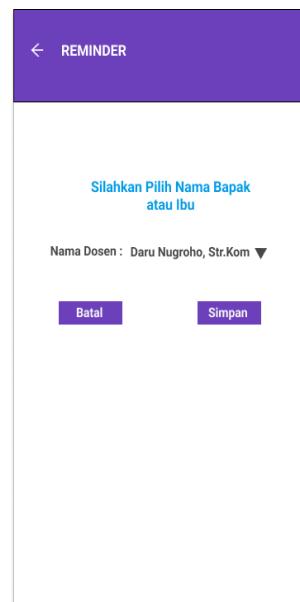
Gambar 3.17. *Prototype*
Atur Waktu Notifikasi
Sumber : Data Olahan

Pada Gambar 3.17 fitur untuk mengatur waktu memunculkan notifikasi untuk mahasiswa, terdapat beberapa pilihan waktu yang disediakan mulai dari 5 menit sampai 30 menit sebelum waktu masuk. Pada bagian bawah terdapat tombol batal untuk membatalkan pengaturan waktu dan tombol simpan untuk menyimpan pengaturan waktu notifikasi.

3. Iterasi Ketiga



Gambar 3.18. *Prototype*
Pilihan Jurusan Dan
Semester

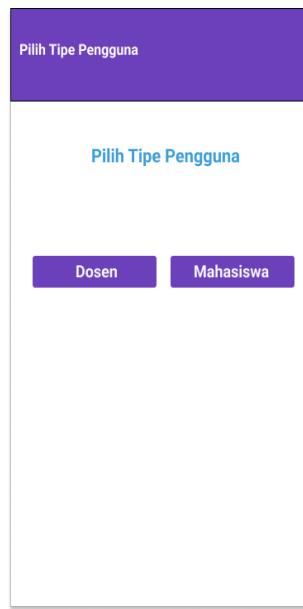


Gambar 3.19. *Prototype*
Pilihan Nama Dosen

Sumber : Data Olahan

Pada Gambar 3.18 menu pilihan daftar jurusan atau prodi dan menu pilihan semester atau kelas, terdapat menu drop down yang akan menampilkan list daftar dari jurusan dan semester yang diambil dari data yang ada di file excel jadwal perkuliahan, selanjutnya terdapat tombol simpan untuk menyimpan pilihan dan tombol batal untuk membatalkan pilihan.

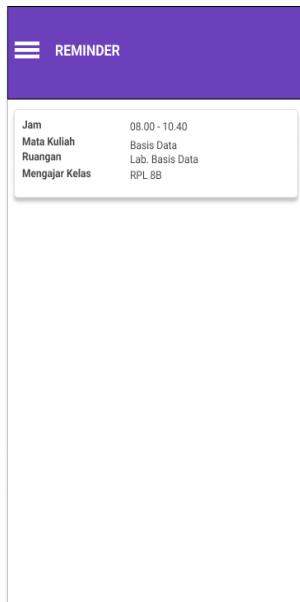
Pada Gambar 3.19 menu pilihan daftar nama dosen, terdapat menu drop down yang akan menampilkan nama dosen dimana datanya diambil dari *API Backend* jadwal mata kuliah, selanjutnya terdapat tombol simpan untuk menyimpan pilihan dan tombol batal untuk membatalkan pilihan



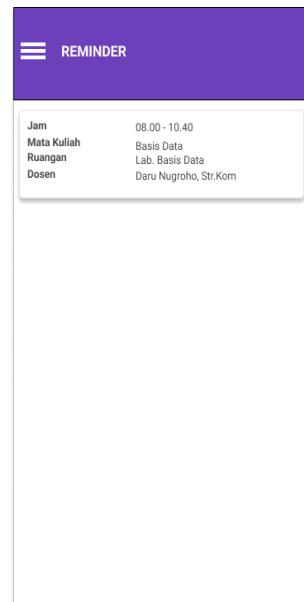
Gambar 3.20. *Prototype*
Atur Waktu Notifikasi
Sumber : Data Olahan

Pada Gambar 3.20 aplikasi terdapat pilihan tipe pengguna untuk yaitu dosen dan mahasiswa, selanjutnya terdapat tombol untuk melanjutya ke tampilan berikut nya, diamana setiap sesi dari tipe pengguna terdapat perbedaan fitur.

4. Iterasi Keempat



Gambar 3.21. *Prototype*
Menu Mata Kuliah Sedang
Berlangsung Dosen



Gambar 3.22. *Prototype*
Menu Mata Kuliah Sedang
Berlangsung Mahasiswa

Sumber : Data Olahan

Pada Gambar 3.21 Halaman ini digunakan untuk menampilkan mata kuliah yang sedang berlangsung atau sudah memasuki waktunya kepada pengguna dosen, dimana pada data mata kuliah menampilkan informasi seperti jam, nama mata kuliah, ruangan kuliah dan akan mengajar dikelas yang mana.

Pada Gambar 3.22 Halaman ini digunakan untuk menampilkan mata kuliah yang sedang berlangsung atau sudah memasuki waktunya kepada pengguna mahasiswa, dimana pada data mata kuliah menampilkan informasi seperti jam, nama mata kuliah, ruangan kuliah dosen pengampu.

3.2.3 Coding (Implementasi)

Selanjutnya pada tahap ini dilakukan kegiatan penerapan desain yang sudah dibuat kedalam bentuk aplikasi *Android* dengan menggunakan bahasa pemrograman. Adapun bahasa pemograman yang penulis digunakan adalah *PHP* dan *framework laravel* untuk bagian *back end* dan bahasa pemrograman *Kotlin* untuk bagian *front end Android* dan untuk *database* menggunakan *MySQL* dan *Room Database*. Dimana dalam proses pengkodean akan dilakukan *refactoring code* dengan tujuan untuk melakukan pengecekan kode program agar tidak menimbulkan *error*.

a. *Refactoring*

Pada proses *refactoring* disini penulis melakukan *refactoring* yaitu membersihkan *source code* dari *code-code* yang sudah tidak digunakan lagi, agar kode program tetap bersih dan mudah untuk dibaca. Adapun untuk hasil *refactoring* yang telah dilakukan dapat dilihat pada tabel berikut.

Tabel 3.12. Refactoring

Source Code : AlarmBroadcastReceiver	Refactoring	Keterangan
<pre> private fun userType(context: Context) : Boolean{ // val preferences = context.getSharedPreferences("login_preferences", Context.MODE_PRIVATE) // return preferences.getString("user_type", null) == UserType.Dosen.toString() return LoginPreferences(context).cekUserType() } </pre>	<pre> private fun userType(context: Context) : Boolean{ return LoginPreferences(context).cekUserType() } </pre>	Menhapus code yang sudah tidak dipakai pada fungsi userType()
Source Code : JadwalDosenRepository	Refactoring	Keterangan
<pre> private var jadwalDosenDao : JadwalDosenDao private var jadwalDosenForReschedule : LiveData<List<JadwalDosen>> private var jadwalDosenToNotification : LiveData<JadwalDosen> private var jadwalDosenSekarang : LiveData<JadwalDosen> private var jadwalDosenToMataKuliahActivity : LiveData<JadwalDosen> init { // val pref = application.applicationContext.getSharedPreferences("data_reminder", Context.MODE_PRIVATE) // val alarmId = pref.getInt("id_alarm_pref", 0) val reminderPreferences = ReminderPreferences(application) val alarmId = reminderPreferences.getAlarmPref() val db = JadwalDosenRoomDatabase.getDatabase(application) jadwalDosenDao = db.jadwalDosenDao() jadwalDosenForReschedule = jadwalDosenDao.getJadwalDosen() jadwalDosenToNotification = jadwalDosenDao.getMataKuliahToNotification(alarmId) jadwalDosenSekarang = jadwalDosenDao.getCurrentMataKuliah(alarmId) jadwalDosenToMataKuliahActivity = jadwalDosenDao.getMataKuliahToDestination(alarmId) } </pre>	<pre> private var jadwalDosenDao: JadwalDosenDao private val reminderPreferences = ReminderPreferences(application) private val alarmId = reminderPreferences.getAlarmPref() init { val db = JadwalDosenRoomDatabase.getDatabase(application) jadwalDosenDao = db.jadwalDosenDao() } </pre>	Menghapus code yang sudah tidak dipakai pada cakupan init
Source Code : JadwalMahasiswaRepository	Refactoring	Keterangan
<pre> private var jadwalMahasiswaDao : JadwalMahasiswaDao private var jadwalMahasiswaForReschedule : LiveData<List<JadwalMahasiswa>> private var jadwalMahasiswaToNotification : LiveData<JadwalMahasiswa> private var jadwalMahasiswaSekarang : LiveData<JadwalMahasiswa> private lateinit var jadwalMahasiswaForAturWaktuNotif : List<JadwalMahasiswa> init { // val pref = application.applicationContext.getSharedPreferences("data_reminder", Context.MODE_PRIVATE) // val alarmId = pref.getInt("id_alarm_pref", 0) val reminderPreferences = ReminderPreferences(application) val alarmId = reminderPreferences.getAlarmPref() } </pre>	<pre> private val jadwalMahasiswaDao: JadwalMahasiswaDao private val reminderPreferences = ReminderPreferences(application) private val alarmId = reminderPreferences.getAlarmPref() init { val db = JadwalMahasiswaRoomDatabase.getDatabase(application) jadwalMahasiswaDao = db.jadwalMahasiswaDao() } </pre>	Menghapus code yang sudah tidak dipakai pada cakupan init

<pre>val db = JadwalMahasiswaRoomDatabase.getDatabase(application) jadwalMahasiswaDao = db.jadwalMahasiswaDao() jadwalMahasiswaForReschedule = jadwalMahasiswaDao.getJadwalMahasiswa() jadwalMahasiswaToNotification = jadwalMahasiswaDao.getMataKuliahToNotification(alarmId) jadwalMahasiswaSekarang = jadwalMahasiswaDao.getCurrentMataKuliah(alarmId)</pre>		
<i>Source Code : DosenReminder</i>	<i>Refactoring</i>	Keterangan
<pre>private fun getDelayNotifPref(context: Context): Int { // val pref = context.getSharedPreferences("data_reminder", AppCompatActivity.MODE_PRIVATE) // return pref.getInt("delay_notifikasi", context.getString(R.string.default_delay).toInt()) return ReminderPreferences(context).getDelayNotification() }</pre>	<pre>private fun getDelayNotifPref(context: Context): Int { return ReminderPreferences(context).getDelayNotification() }</pre>	Menghapus code yang sudah tidak dipakai pada fungsi getDelayNotifPre()
<i>Source Code : MahasiswaReminder</i>	<i>Refactoring</i>	Keterangan
<pre>private fun getDelayNotifPref(context: Context): Int { // val pref = context.getSharedPreferences("data_reminder", AppCompatActivity.MODE_PRIVATE) // return pref.getInt("delay_notifikasi", context.getString(R.string.default_delay).toInt()) return ReminderPreferences(context).getDelayNotification() }</pre>	<pre>private fun getDelayNotifPref(context: Context): Int { return ReminderPreferences(context).getDelayNotification() }</pre>	Menghapus code yang sudah tidak dipakai pada fungsi getDelayNotifPref()
<i>Source Code : DosenNotificationService</i>	<i>Refactoring</i>	Keterangan
<pre>val notification = NotificationCompat.Builder(this, ReminderApplication.CHANNEL_ID) .setSmallIcon(R.drawable.ic_baseline_notifications_active_24) .setContentTitle("My MahasiswaReminder Dosen") .setStyle(NotificationCompat.BigTextStyle() .bigText("\${reminderPreferences.getDelayNotification()} menit lagi kuliah \$ {jadwalDosen.namaMataKuliah} akan dimulai! \n" + "Ruangan : \${jadwalDosen.ruangan} \n" + "Jam : \${FormatWaktu.formatJam(jadwalDosen.jam, jadwalDosen.menit)} \n" + "Selama : \${jadwalDosen.jam}") .setContentIntent(pendingIntent) .build()</pre>	<pre>val notification = NotificationCompat.Builder(this, ReminderApplication.CHANNEL_ID) .setSmallIcon(R.drawable.ic_baseline_notifications_active_24) .setContentTitle(application.getString(R.string.content_title_dosen_notification_service)) .setStyle(NotificationCompat.BigTextStyle() .bigText(getString(R.string.content_dosen_notification_service, reminderPreferences.getDelayNotification().toString(), jadwalDosen.namaMataKuliah, jadwalDosen.ruangan, FormatWaktu.formatJam(jadwalDosen.jam, jadwalDosen.menit))) .setContentIntent(pendingIntent) .build()</pre>	Memindahkan content pada BigText() ke resource string.xml

Source Code : MahasiswaNotificationService	Refactoring	Keterangan
<pre>val notification = NotificationCompat.Builder(this, ReminderApplication.CHANNEL_ID) .setSmallIcon(R.drawable.ic_baseline_notifications_active_24) .setContentTitle(getString(R.string.content_title_mahasiswa_notification_service)) .setStyle(NotificationCompat.BigTextStyle() .bigText(getString(R.string.content_mahasiswa_notification_service, reminderPreferences.getDelayNotification().toString(), jadwalMahasiswa.namaMataKuliah, jadwalMahasiswa.dosen, jadwalMahasiswa.ruangan, FormatWaktu.formatJam(jadwalMahasiswa.jam, jadwalMahasiswa.menit)))) .setContentIntent(pendingIntent) .build()</pre>	<pre>val notification = NotificationCompat.Builder(this, ReminderApplication.CHANNEL_ID) .setSmallIcon(R.drawable.ic_baseline_notifications_active_2 4) .setContentTitle("My MahasiswaReminder Mahasiswa") .setStyle(NotificationCompat.BigTextStyle() .bigText("\${reminderPreferences.getDelayNotification()}" menit lagi kuliah \${jadwalMahasiswa.namaMataKuliah} akan dimulai! \n" + "Ruangan : \${jadwalMahasiswa.ruangan} \n" + "Jam : \$" {FormatWaktu.formatJam(jadwalMahasiswa.jam, jadwalMahasiswa.menit)} \n" + "Selama : \${jadwalMahasiswa.jam}") .setContentIntent(pendingIntent) .build()</pre>	Memindahkan content pada BigText() ke resource string.xml
Source Code : AlarmService	Refactoring	Keterangan
<pre>override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int { val reminderPreferences = ReminderPreferences(application) val idAlarm = intent.getIntExtra(AlarmBroadcastReceiver.ALARM_ID,0) // val pref = this.getSharedPreferences("data_reminder", MODE_PRIVATE) // val editor = pref.edit() // editor.putInt("id_alarm_pref", data) // editor.apply() reminderPreferences.setIdAlarmPref(idAlarm) mediaPlayer.start() vibrator.vibrate(1000) return START_STICKY }</pre>	<pre>override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int { val reminderPreferences = ReminderPreferences(application) val idAlarm = intent.getIntExtra(AlarmBroadcastReceiver.ALARM_ID,0) reminderPreferences.setIdAlarmPref(idAlarm) mediaPlayer.start() vibrator.vibrate(1000) return START_STICKY }</pre>	Menghapus code yang sudah tidak dipakai pada fungsi onStartCommand()
Source Code : AturWaktuViewModel	Refactoring	Keterangan
<pre>private val loginPreferences = LoginPreferences(app) private val reminderPreferences = ReminderPreferences(app) private val jadwalDosenRepository = JadwalDosenRepository(app) private val jadwalMahasiswaRepository = JadwalMahasiswaRepository(app) val itemJadwalDosen : LiveData<List<JadwalDosen>> = jadwalDosenRepository.getJadwalDosenForReschedule() // karena dijadwalkan ulang val itemJadwalMahasiswa: LiveData<List<JadwalMahasiswa>> =</pre>	<pre>private val loginPreferences = LoginPreferences(app) private val reminderPreferences = ReminderPreferences(app) private val jadwalDosenRepository = JadwalDosenRepository(app) private val jadwalMahasiswaRepository = JadwalMahasiswaRepository(app) fun setWaktuDelayNotifikasiDosen(</pre>	Mengganti tipe LiveData pada fungsi setWaktuDelayNotifikasiDosen dan setWaktuDelayNotifikasiMahasiswa dengan tipe data biasa.

<pre> jadwalMahasiswaRepository.getJadwalMahasiswaForReschedule() // fun setWaktuDelayNotifikasiDosen(// opt1: Boolean, opt2: Boolean, opt3: Boolean, // opt4: Boolean, opt5: Boolean, opt6: Boolean, // jadwalDosen : JadwalDosen //) { // val dosenReminder = DosenReminder(jadwalDosen) // val menit = cekPilihan(opt1, opt2, opt3, opt4, opt5, opt6) // updateDelayNotifPref(menit) // dosenReminder.updateWaktuReminder(app.applicationContext, menit) // } /** * Terdapat problem fungsi ini * jika menggunakan live data, * tidak dapat menjalankan fungsi ini * ketika tombol simpan di klik */ fun setWaktuDelayNotifikasiMahasiswa(opt1: Boolean, opt2: Boolean, opt3: Boolean, opt4: Boolean, opt5: Boolean, opt6: Boolean, jadwalMahasiswa: JadwalMahasiswa) { val reminder = MahasiswaReminder(jadwalMahasiswa) val menit = cekPilihan(opt1, opt2, opt3, opt4, opt5, opt6) updateDelayNotifPref(menit) reminder.updateWaktuReminder(app.applicationContext, menit) } </pre>	<pre> opt1: Boolean, opt2: Boolean, opt3: Boolean, opt4: Boolean, opt5: Boolean, opt6: Boolean,) { val menit = cekPilihan(opt1, opt2, opt3, opt4, opt5, opt6) viewModelScope.launch { for (jadwalDosen in jadwalDosenRepository.getJadwalMahasiswaForAturWaktu Notif())){ val dosenReminder = DosenReminder(jadwalDosen) dosenReminder.updateWaktuReminder(app.applicationConte xt) } } updateDelayNotifPref(menit) } // Tidak menggunakan live data fun setWaktuDelayNotifikasiMahasiswa(opt1: Boolean, opt2: Boolean, opt3: Boolean, opt4: Boolean, opt5: Boolean, opt6: Boolean,) { val menit = cekPilihan(opt1, opt2, opt3, opt4, opt5, opt6) viewModelScope.launch { for (jadwalMahasiswa in jadwalMahasiswaRepository.getJadwalMahasiswaForAturW aktuNotif()){ val mahasiswaReminder = MahasiswaReminder(jadwalMahasiswa) mahasiswaReminder.updateWaktuReminder(app.applicationC ontext) } } updateDelayNotifPref(menit) } </pre>	
<p>Source Code : AturWaktuViewModel</p> <pre> /** * TODO("hapus cekHari() sudah tidak digunakan lagi") */ /** * Untuk mengubah nilai boolean pada hari di class MahasiswaReminder() * * DEPRECATED -> WILL REMOVE */ private fun cekHari(jadwalMahasiswa: JadwalMahasiswa, reminder: MahasiswaReminder) { when (jadwalMahasiswa.hari) { </pre>	<p><i>Refactoring</i></p>	<p>Keterangan</p>
		<p>Menghapus fungsi cekHari pada karena fungsi tersebut sudah tidak digunakan lagi.</p>

<pre> "senin" -> { reminder.senin = true } "selasa" -> { reminder.selasa = true } "rabu" -> { reminder.rabu = true } "kamis" -> { reminder.kamis = true } "jumat" -> { reminder.jumat = true } else -> { reminder.sabtu = true } } </pre>		
<p>Source Code : AturWaktuViewModel</p> <pre> /** * Untuk menulis nilai delay notifikasi pada shared preferences */ private fun updateDelayNotifPref(delay: Int){ // val pref = app.getSharedPreferences("data_reminder", Context.MODE_PRIVATE) // val editor = pref.edit() // editor.putInt("delay_notifikasi", delay) // editor.apply() reminderPreferences.updateDelayNotification(delay) } </pre>	<p>Refactoring</p> <pre> /** * Untuk menulis nilai delay notifikasi pada shared preferences */ private fun updateDelayNotifPref(delay: Int){ reminderPreferences.updateDelayNotification(delay) } </pre>	<p>Keterangan</p> <p>Menghapus code yang sudah tidak dipakai pada fungsi updateDelayNotifPref()</p>
<p>Source Code : AturWaktuViewModel</p> <pre> /** * Untuk mengambil nilai delay notifikasi pada shared preferences */ fun getDelayNotifPref(): Int { // val pref = app.getSharedPreferences("data_reminder", Context.MODE_PRIVATE) // return pref.getInt("delay_notifikasi", app.getString(R.string.default_delay).toInt()) return reminderPreferences.getDelayNotification() } </pre>	<p>Refactoring</p> <pre> /** * Untuk mengambil nilai delay notifikasi pada shared preferences */ fun getDelayNotifPref(): Int { return reminderPreferences.getDelayNotification() } </pre>	<p>Keterangan</p> <p>Menghapus code yang sudah tidak dipakai pada fungsi getDelayNotifPref()</p>

Source Code : MainActivityViewModel	Refactoring	Keterangan
<pre>private const val USER_TYPE = "user_type" private const val LOGIN_PREFERENCES = "login_preferences" private const val REMINDER_PREFERENCES = "dataReminder" class MainActivityViewModel(private val app : Application) : AndroidViewModel(app)</pre>	<pre>class MainActivityViewModel(private val app : Application) : AndroidViewModel(app)</pre>	Menghapus variable yang sudah tidak dipakai pada class MainActivityViewModel
Source Code : MainActivity	Refactoring	Keterangan
<pre>/** * TODO("Pindahkan text ke resource string.xml") */ R.id.nav_logout -> { alertDialog.apply { setTitle("Perhatian") setIcon(R.drawable.ic_baseline_warning_24) setMessage("Apakah anda yakin ingin keluar?\n" + "Ini akan menghapus semua data") setPositiveButton("Ya") { _, _ -> viewModel.logout() finish() } setNegativeButton("Tidak") { batal, _ -> batal.cancel() } create() show() } } activityMainBinding.drawerLayout.closeDrawer(GravityCompat.START)</pre>	<pre>R.id.nav_logout -> { alertDialog.apply { setTitle(getString(R.string.title_alert_dialog)) setIcon(R.drawable.ic_baseline_warning_24) setMessage(getString(R.string.message_alert_dialog)) setPositiveButton(getString(R.string.txt_positive_button_alert_dialog)) { _, _ -> viewModel.logout() finish() } setNegativeButton(getString(R.string.txt_negative_button_alert_dialog)) { batal, _ -> batal.cancel() } create() show() } } activityMainBinding.drawerLayout.closeDrawer(GravityCompat.START)</pre>	Memindahkan teks string yang secara hardcoded ke resource strings.xml

Source Code : LoginPreferences	Keterangan
<pre> class LoginPreferences(context: Context) { companion object { private const val NAME_LOGIN_PREFERENCES = "login_preferences" private const val KEY_DOSEN_NAME = "dosen_name" private const val KEY_USER_TYPE = "user_type" private const val KEY_KELAS = "kelas" } private val loginSharedPreferences = context.getSharedPreferences(NAME_LOGIN_PREFERENCES, Context.MODE_PRIVATE) private val editor = loginSharedPreferences.edit() fun setDosenName(name : String) { editor.putString(KEY_DOSEN_NAME, name).apply() } fun getDosenName() : String? { return loginSharedPreferences.getString(KEY_DOSEN_NAME, null) } fun setKelas(kelas: String) { editor.putString(KEY_KELAS, kelas).apply() } fun getKelas() : String? { return loginSharedPreferences.getString(KEY_KELAS, null) } fun isLoggedIn() : Boolean{ return loginSharedPreferences.getString(KEY_USER_TYPE, null) != null } fun cekUserType() : Boolean { return getUserType().equals(UserType.Dosen.toString()) } fun setUserType(user: UserType) { if (user == UserType.Dosen user == UserType.Mahasiswa){ editor.putString(KEY_USER_TYPE, user.toString()).apply() } } fun getUserType() : String? { return loginSharedPreferences.getString(KEY_USER_TYPE, null) } fun validate(user: UserType) : Boolean { return user == UserType.Dosen user == UserType.Mahasiswa } } </pre>	<p>Menambahkan class LoginPreferences yang digunakan untuk memudahkan dalam mengakses data pada login_preferences.xml</p>

<pre> } fun removeLoginPreferences() { editor.clear().apply() } } </pre>	
<i>Source Code : ReminderPreferences</i>	<i>Keterangan</i>
<pre> class ReminderPreferences(private val context: Context) { companion object { private const val NAME_RemINDER_PREFERENCES = "reminder_preferences" private const val KEY_DELAY_NOTIFICATION = "delay_notification" private const val KEY_ID_ALARM = "id_alarm" } private val reminderPreferences = context.getSharedPreferences(NAME_RemINDER_PREFERENCES, Context.MODE_PRIVATE) private val editor = reminderPreferences.edit() fun getDelayNotification(): Int { return reminderPreferences.getInt(KEY_DELAY_NOTIFICATION, context.getString(R.string.default_delay).toInt()) } fun updateDelayNotification(delay: Int) { editor.putInt(KEY_DELAY_NOTIFICATION, delay).apply() } fun getIdAlarmPref(): Int { return reminderPreferences.getInt(KEY_ID_ALARM, 0) } fun setIdAlarmPref(idAlarm: Int) { editor.putInt(KEY_ID_ALARM, idAlarm).apply() } fun clearReminderPreferences() { editor.clear().apply() } } </pre>	Menambahkan class ReminderPreferences yang digunakan untuk memudahkan dalam mengakses data pada reminder_preferences.xml

Source Code : FormatWaktu	Keterangan
<pre>object FormatWaktu { private lateinit var calendar: Calendar fun formatJam(jam: Int, menit: Int): String { val formatter = SimpleDateFormat("HH.mm", Locale.getDefault()) calendar = Calendar.getInstance() calendar[Calendar.HOUR_OF_DAY] = jam calendar[Calendar.MINUTE] = menit return formatter.format(calendar.time) } fun formatJamForDebugLog(calendar: Date): String { return SimpleDateFormat("HH.mm", Locale.getDefault()).format(calendar.time) } }</pre>	Menambahkan object FormatWaktu yang digunakan untuk memudahkan dalam menampilkan waktu dalam format 24 jam.
Source Code : UserType	Keterangan
<pre>enum class UserType { Dosen, Mahasiswa }</pre>	Menambahkan enum class UserType yang akan digunakan untuk mendefinisikan tipe pengguna agar mengurangi kesalahan dalam pengetikan.

b. *Pair Programming*

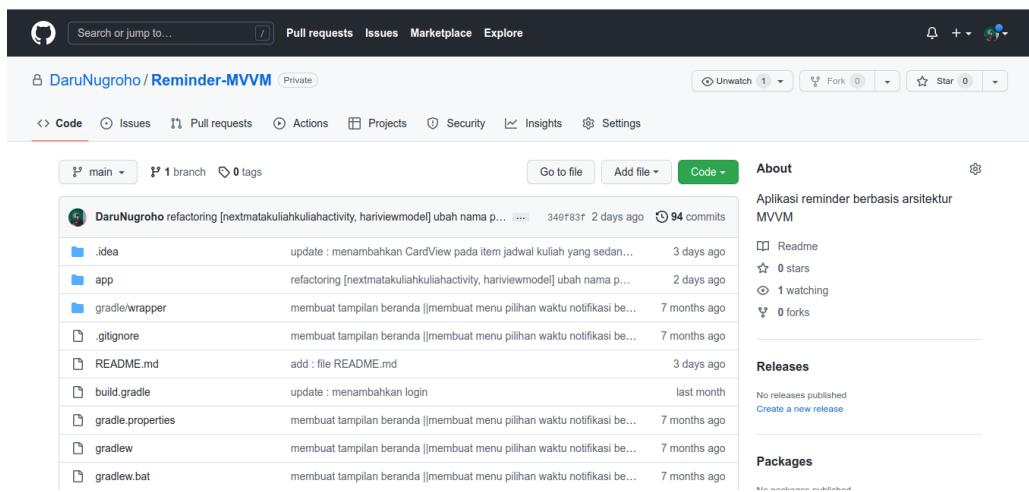
Pada *Pair Programming* disini penulis bekerja sama dengan dosen pembimbing untuk penulisan kode program, dimana dosen pembimbing mengoreksi *code* program yang telah dibuat.



Gambar 3.23. *Pair Programming*
Sumber : Data Olahan

c. *Continous Interation*

Untuk mencapai *Continous Integrations* disini penulis menggunakan *tools Git* dan *Github* untuk megelola dan menyimpan source code dari aplikasi reminder jadwal perkuliahan agar mempermudah penulis saat mengidentifikasi bug dan memperbaikinya selama proses pengembangan.



Gambar 3.24. *Continous Integration*
Sumber : Data Olahan

3.2.4 *Testing* (Pengujian)

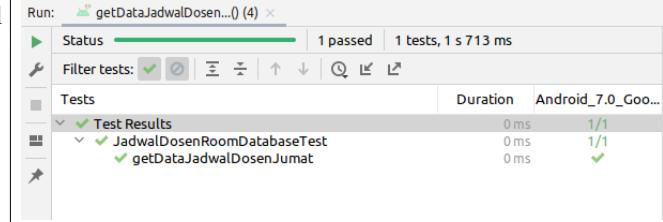
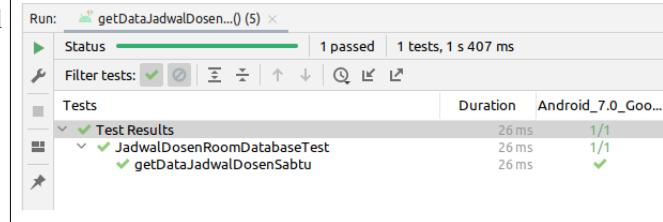
Setelah tahap pengkodean selesai, tahap selanjutnya adalah melakukan pengujian sistem untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Metode pengujian yang penulis gunakan pada tahap ini adalah metode *unit testing* dan *blackbox testing*, dimana pengujian dilakukan pada fitur-fitur yang tersedia pada aplikasi apakah sudah berjalan sesuai dengan fungsinya masing-masing.

a. *Unit Testing*

- Tipe Pengguna Dosen

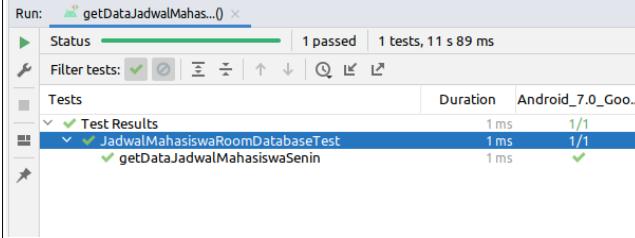
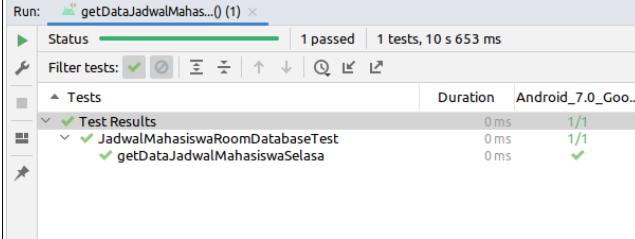
Tabel 3.13. *Unit Testing* Tipe Pengguna Dosen

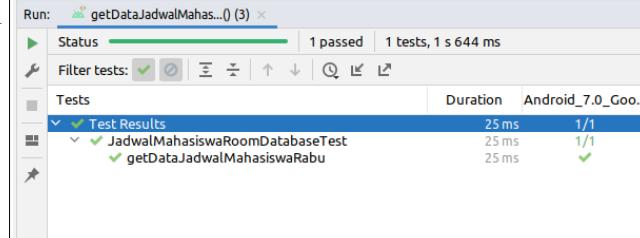
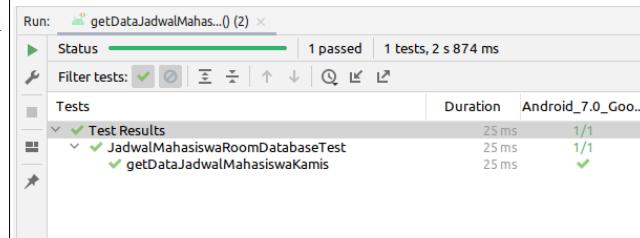
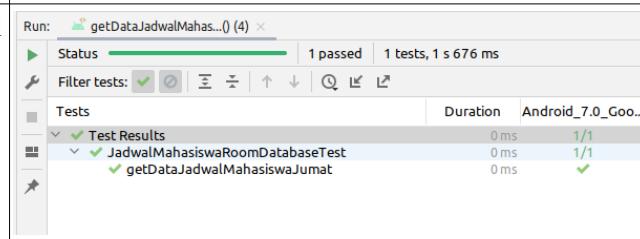
<i>Unit Test</i>	Skenario	<i>Acceptance Test</i>	Keterangan
getJadwalDosenSelasa()	Mengambil data jadwal dosen untuk hari selasa	Dapat menampilkan data jadwal dosen yang ada pada hari selasa	
getJadwalDosenRabu()	Mengambil data jadwal dosen untuk hari rabu	Dapat menampilkan data jadwal dosen yang ada pada hari rabu	
getJadwalDosenKamis()	Mengambil data jadwal dosen untuk hari kamis	Dapat menampilkan data jadwal dosen yang ada pada hari kamis	

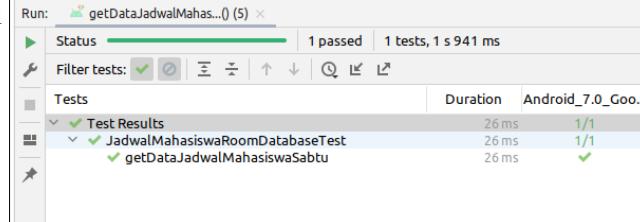
<i>Unit Test</i>	<i>Skenario</i>	<i>Acceptance Test</i>	<i>Keterangan</i>																		
getJadwalDosenJumat()	Mengambil data jadwal dosen untuk hari jumat	Dapat menampilkan data jadwal dosen yang ada pada hari jumat	 <p>Run: <code>getDataJadwalDosenJumat()</code> (4) ×</p> <table border="1"> <thead> <tr> <th>Status</th> <th>1 passed</th> <th>1 tests, 1 s 713 ms</th> </tr> </thead> <tbody> <tr> <td>Filter tests:</td> <td>✓</td> <td></td> </tr> <tr> <td>Tests</td> <td></td> <td>Duration Android_7.0_Goo...</td> </tr> <tr> <td>Test Results</td> <td>0 ms</td> <td>1/1</td> </tr> <tr> <td> JadwalDosenRoomDatabaseTest</td> <td>0 ms</td> <td>1/1</td> </tr> <tr> <td> getDataJadwalDosenJumat</td> <td>0 ms</td> <td>✓</td> </tr> </tbody> </table>	Status	1 passed	1 tests, 1 s 713 ms	Filter tests:	✓		Tests		Duration Android_7.0_Goo...	Test Results	0 ms	1/1	JadwalDosenRoomDatabaseTest	0 ms	1/1	getDataJadwalDosenJumat	0 ms	✓
Status	1 passed	1 tests, 1 s 713 ms																			
Filter tests:	✓																				
Tests		Duration Android_7.0_Goo...																			
Test Results	0 ms	1/1																			
JadwalDosenRoomDatabaseTest	0 ms	1/1																			
getDataJadwalDosenJumat	0 ms	✓																			
getJadwalDosenSabtu()	Mengambil data jadwal dosen untuk hari sabtu	Dapat menampilkan data jadwal dosen yang ada pada hari sabtu	 <p>Run: <code>getDataJadwalDosenSabtu()</code> (5) ×</p> <table border="1"> <thead> <tr> <th>Status</th> <th>1 passed</th> <th>1 tests, 1 s 407 ms</th> </tr> </thead> <tbody> <tr> <td>Filter tests:</td> <td>✓</td> <td></td> </tr> <tr> <td>Tests</td> <td></td> <td>Duration Android_7.0_Goo...</td> </tr> <tr> <td>Test Results</td> <td>26 ms</td> <td>1/1</td> </tr> <tr> <td> JadwalDosenRoomDatabaseTest</td> <td>26 ms</td> <td>1/1</td> </tr> <tr> <td> getDataJadwalDosenSabtu</td> <td>26 ms</td> <td>✓</td> </tr> </tbody> </table>	Status	1 passed	1 tests, 1 s 407 ms	Filter tests:	✓		Tests		Duration Android_7.0_Goo...	Test Results	26 ms	1/1	JadwalDosenRoomDatabaseTest	26 ms	1/1	getDataJadwalDosenSabtu	26 ms	✓
Status	1 passed	1 tests, 1 s 407 ms																			
Filter tests:	✓																				
Tests		Duration Android_7.0_Goo...																			
Test Results	26 ms	1/1																			
JadwalDosenRoomDatabaseTest	26 ms	1/1																			
getDataJadwalDosenSabtu	26 ms	✓																			

- Tipe Pengguna Mahasiswa

Tabel 3.14. *Unit Testing* Tipe Pengguna Mahasiswa

<i>Unit Test</i>	Skenario	<i>Acceptance Test</i>	Keterangan
getJadwalMahasiswaSenin()	Mengambil data jadwal kuliah mahasiswa untuk hari senin	Dapat menampilkan data jadwal mahasiswa yang ada pada hari senin	
getJadwalMahasiswaSelasa()	Mengambil data jadwal kuliah mahasiswa untuk hari selasa	Dapat menampilkan data jadwal mahasiswa yang ada pada hari selasa	

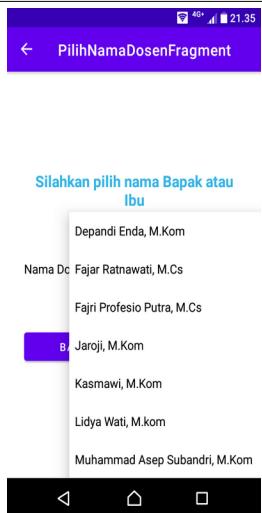
<i>Unit Test</i>	<i>Skenario</i>	<i>Acceptance Test</i>	<i>Keterangan</i>
getJadwalMahasiswaRabu()	Mengambil data jadwal kuliah mahasiswa untuk hari rabu	Dapat menampilkan data jadwal mahasiswa yang ada pada hari rabu	
getJadwalMahasiswaKamis()	Mengambil data jadwal kuliah mahasiswa untuk hari kamis	Dapat menampilkan data jadwal mahasiswa yang ada pada hari kamis	
getJadwalMahasiswaJumat()	Mengambil data jadwal kuliah mahasiswa untuk hari jumat	Dapat menampilkan data jadwal mahasiswa yang ada pada hari jumat	

<i>Unit Test</i>	<i>Skenario</i>	<i>Acceptance Test</i>	<i>Keterangan</i>												
getJadwalMahasiswaSabtu()	Mengambil data jadwal kuliah mahasiswa untuk hari sabtu	Dapat menampilkan data jadwal mahasiswa yang ada pada hari sabtu	 <p>Run: <code>getDataJadwalMahasiswaSabtu()</code> (5) ×</p> <table border="1"> <thead> <tr> <th>Tests</th> <th>Duration</th> <th>Android_7.0_Goo...</th> </tr> </thead> <tbody> <tr> <td>Test Results</td> <td>26 ms</td> <td>1/1</td> </tr> <tr> <td> JadwalMahasiswaRoomDatabaseTest</td> <td>26 ms</td> <td>1/1</td> </tr> <tr> <td> getDataJadwalMahasiswaSabtu</td> <td>26 ms</td> <td>✓</td> </tr> </tbody> </table>	Tests	Duration	Android_7.0_Goo...	Test Results	26 ms	1/1	JadwalMahasiswaRoomDatabaseTest	26 ms	1/1	getDataJadwalMahasiswaSabtu	26 ms	✓
Tests	Duration	Android_7.0_Goo...													
Test Results	26 ms	1/1													
JadwalMahasiswaRoomDatabaseTest	26 ms	1/1													
getDataJadwalMahasiswaSabtu	26 ms	✓													

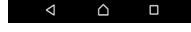
b. *Black Box Testing*

- Tipe Pengguna Dosen

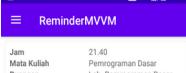
Tabel 3.15. *Black Box Testing* Tipe Pengguna Dosen

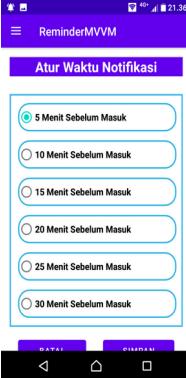
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
1.	Mengambil data nama dosen dari API	Dapat mengambil data nama dosen dan menampilkan pada menu <i>drop down</i>	Sesuai	 <p>The screenshot shows a mobile application interface. At the top, there is a purple header bar with the text "PilihNamaDosenFragment". Below this is a white search bar with the placeholder text "Silahkan pilih nama Bapak atau Ibu". A list of names is displayed below the search bar, each preceded by a small blue square icon with a letter (e.g., 'B'). The names listed are: Depandi Enda, M.Kom; Nama Dc Fajar Ratnawati, M.Cs; Fajri Profesio Putra, M.Cs; B Jarnoji, M.Kom; Kasmawi, M.Kom; Lidya Wati, M.kom; and Muhammad Asep Subandri, M.Kom. At the bottom of the screen is a black navigation bar with three white icons: a left arrow, a home button, and a right arrow.</p>

No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
2.	Memilih nama dosen pada menu <i>drop down</i>	Dapat menyimpan nama dosen yang telah dipilih	Sesuai	 <p>Silahkan pilih nama Bapak atau ibu</p> <p>Nama Dosen : Depandi Enda, M.K..</p> <p>BATAL SIMPAN</p>
3.	Membuka tab jadwal dosen untuk hari senin	Menampilkan jadwal dosen untuk hari senin	Sesuai	 <p>SENIN SELASA RABU KAMIS JUMAT SABTU</p> <p>Jam Kuliah 10.40 Ruang Mengajar Kelas</p>

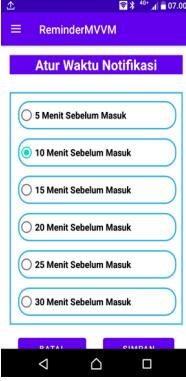
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
4.	Membuka tab jadwal dosen untuk hari selasa	Menampilkan jadwal dosen untuk hari selasa	Sesuai	 
5.	Membuka tab jadwal dosen untuk hari rabu	Menampilkan jadwal dosen untuk hari rabu	Sesuai	 

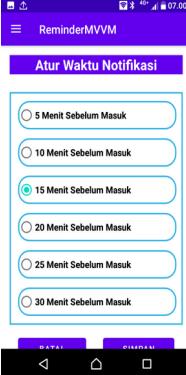
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
6.	Membuka tab jadwal dosen untuk hari kamis	Menampilkan jadwal dosen untuk hari kamis	Sesuai	 
7.	Membuka tab jadwal dosen untuk hari jumat	Menampilkan jadwal dosen untuk hari jumat	Sesuai	 

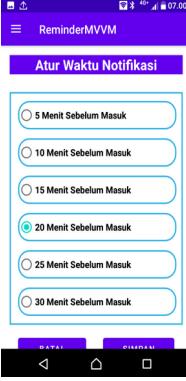
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
8.	Membuka tab jadwal dosen untuk hari sabtu	Menampilkan jadwal dosen untuk hari sabtu	Sesuai	
9.	Memilih menu Mata Kuliah Sekarang	Menampilkan informasi mata kuliah yang telah memasuki waktunya	Sesuai	

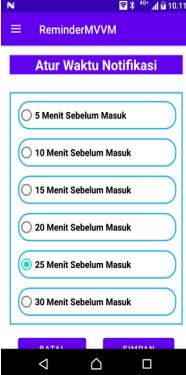
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
10.	Mata kuliah telah memasuki waktu jadwal kuliah yang telah ditentukan	Menampilkan notifikasi yang berisi informasi tentang mata kuliah yang akan berlangsung	Sesuai	
11.	Memilih menu Atur Waktu Notifikasi	Menampilkan pilihan menu delay waktu notifikasi	Sesuai	

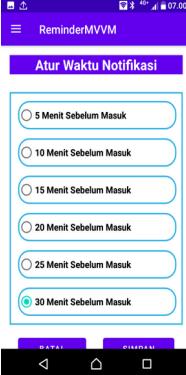
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
12.	Memilih menu Keluar	Menampilkan alert dialog dan menghapus semua data jadwal perkuliahan yang tersimpan, ketika aplikasi dijalankan lagi maka akan menampilkan halaman pilihan tipe pengguna	Sesuai	 

No.	Test Case	Acceptance Testing	Status	Hasil
13.	Memilih menu delay 10 Menit Sebelum Masuk	Menampilkan notifikasi 10 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 

No.	Test Case	Acceptance Testing	Status	Hasil
14.	Memilih menu delay 15 Menit Sebelum Masuk	Menampilkan notifikasi 15 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 

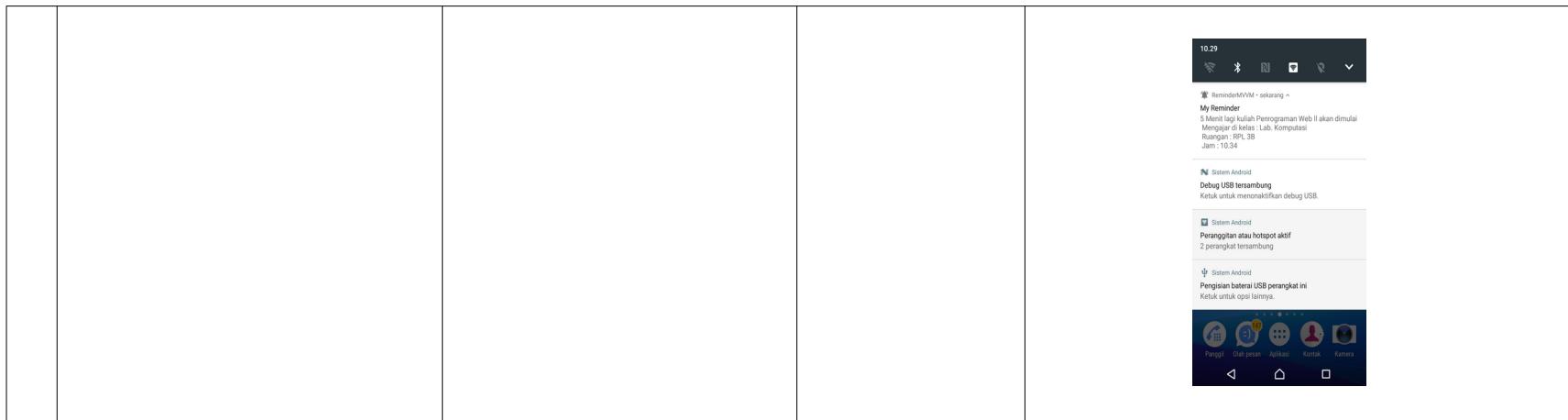
No.	Test Case	Acceptance Testing	Status	Hasil
15.	Memilih menu delay 20 Menit Sebelum Masuk	Menampilkan notifikasi 20 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 

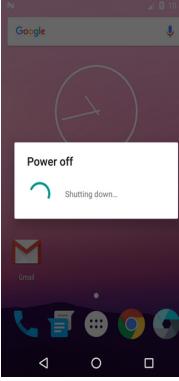
No.	Test Case	Acceptance Testing	Status	Hasil
16.	Memilih menu delay 25 Menit Sebelum Masuk	Menampilkan notifikasi 25 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 

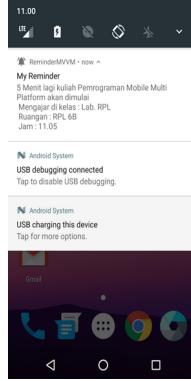
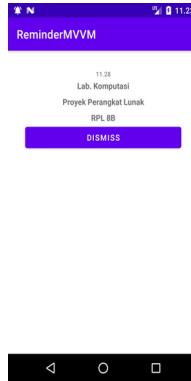
No.	Test Case	Acceptance Testing	Status	Hasil
17.	Memilih menu delay 30 Menit Sebelum Masuk	Menampilkan notifikasi 30 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 

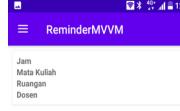
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
18.	Mematikan reminder	Tidak menampilkan notifikasi ketika jadwal perkuliahan sudah memasuki waktu nya	Sesuai	 

No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
19.	Mematikan reminder dan menghidupkan lagi	Menampilkan notifikasi perkuliahan yang akan berlangsung	Sesuai	 



No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
20.	Mematikan perangkat dan menghidupkan lagi	Reminder tetap berjalan dan dapat menampilkan notifikasi perkuliahan saat sudah memasuki waktunya	Sesuai	 

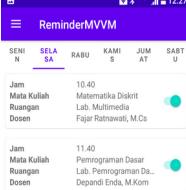
				
21.	Klik pada notifikasi perkuliahan yang ditampilkan	Menampilkan halaman yang berisi informasi tentang mata kuliah yang akan berlangsung selanjutnya, dan terdapat tombol Dismiss untuk mematikan alarm	Sesuai	

No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
22.	Memilih menu “Mata Kuliah Sekarang” saat pertama kali aplikasi dijalankan	Menampilkan keterangan kosong	Sesuai	 

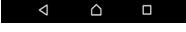
- Tipe Pengguna Mahasiswa

Tabel 3.16. *Black Box Testing* Tipe Pengguna Mahasiswa

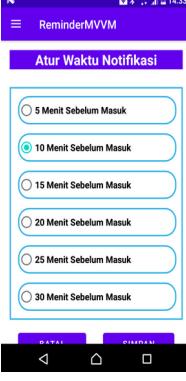
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
1.	Klik pada menu <i>drop down</i> kelas	Dapat menampilkan kelas pada menu <i>drop down</i>	Sesuai	 <p>The screenshot shows a mobile application interface. At the top, there is a header with icons for battery, signal, and time (12.26). Below the header, a back arrow and the text "Pilih Jurusan Semester Frag..." are visible. A message "Silahkan Pilih Jurusan Anda dan Semester Anda" is displayed. Below this, there are two dropdown menus: "Jurusan/Prodi : Teknik Informatika" which is currently set to "RPL1A", and "Semester/Kelas : RPL1A" which is currently set to "RPL1B". A list of options is shown below the dropdown: RPL1B (highlighted in blue), RPL1C, RPL2A, RPL2B, and RPL2C. At the bottom of the screen, there are three navigation icons: a left arrow, a right arrow, and a square.</p>

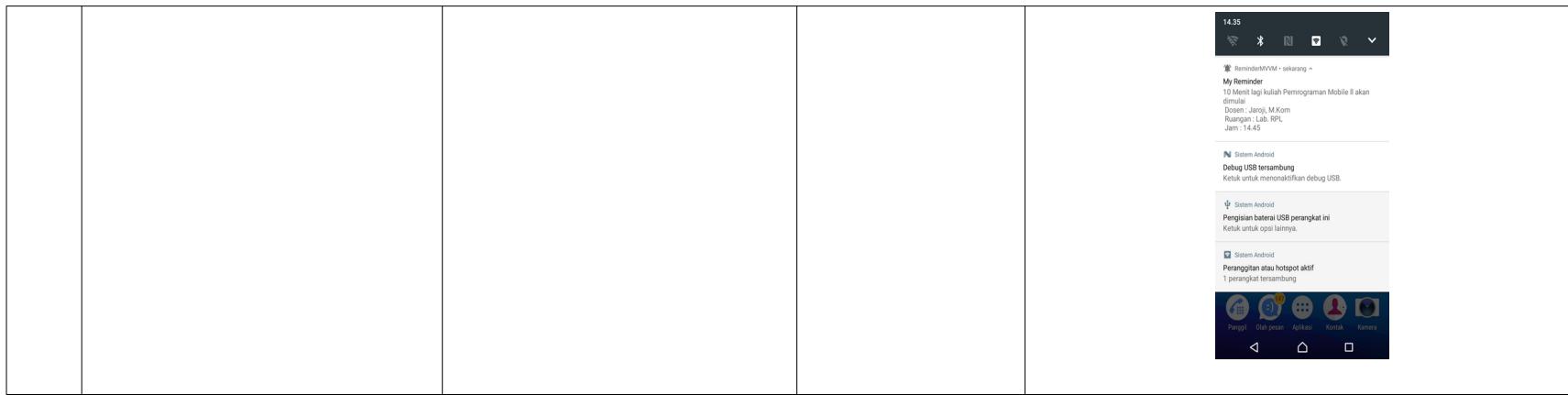
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
2.	Pilih kelas pada menu drop down	Dapat menyimpan kelas yang telah dipilih	Sesuai	
3.	Membuka tab jadwal mahasiswa hari selasa	Dapat menampilkan jadwal perkuliahan mahasiswa untuk hari selasa	Sesuai	

No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
4.	Membuka tab jadwal mahasiswa hari rabu	Dapat menampilkan jadwal perkuliahan mahasiswa untuk hari rabu	Sesuai	 
5.	Membuka tab jadwal mahasiswa hari kamis	Dapat menampilkan jadwal perkuliahan mahasiswa untuk hari kamis	Sesuai	 

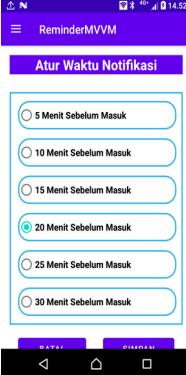
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
6.	Membuka tab jadwal mahasiswa hari jumat	Dapat menampilkan jadwal perkuliahan mahasiswa untuk hari jumat	Sesuai	 
7.	Membuka tab jadwal mahasiswa hari sabtu	Dapat menampilkan jadwal perkuliahan mahasiswa untuk hari sabtu	Sesuai	 

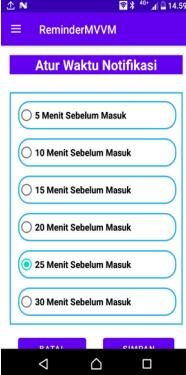
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
8.	Memilih menu “Mata Kuliah Sekarang” saat pertama kali aplikasi dijalankan	Menampilkan keterangan kosong	Sesuai	 
9.	Memilih menu “Mata Kuliah Sekarang” ketika ada mata kuliah yang telah memasuki waktu jadwalnya	Menampilkan informasi mata kuliah yang telah memasuki waktu jadwalnya	Sesuai	 

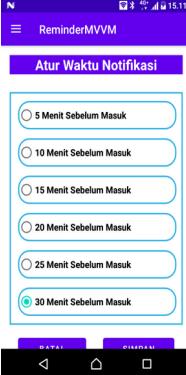
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
10.	Mata kuliah telah memasuki waktu jadwal kuliah yang telah ditentukan	Menampilkan notifikasi yang berisi informasi tentang mata kuliah yang akan berlangsung	Sesuai	 <p>14:28 RemindernMVM - sekarang ~ My Reminder 5 Menit lagi kuliah Pemrograman Web II akan dimulai. Dosen: Banur, M.Cs Ruangan: Lab. Komputasi Jml: 14:33</p> <p>■ Sistem Android Debug USB tersambung Ketuk untuk menonaktifkan debug USB.</p> <p>□ Sistem Android Pengisian baterai USB perangkat ini Ketuk untuk opsi lainnya.</p> <p>□ Sistem Android Pengalihan atau hotspot aktif 1 perangkat tersambung</p>
11.	Memilih menu delay 10 Menit Sebelum Masuk	Menampilkan notifikasi 10 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 <p>14:33 RemindernMVM Atur Waktu Notifikasi</p> <p>5 Menit Sebelum Masuk 10 Menit Sebelum Masuk 15 Menit Sebelum Masuk 20 Menit Sebelum Masuk 25 Menit Sebelum Masuk 30 Menit Sebelum Masuk</p>

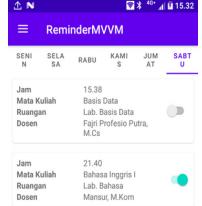


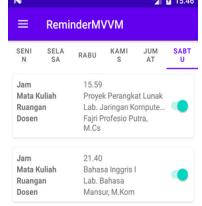
No.	Test Case	Acceptance Testing	Status	Hasil
12.	Memilih menu delay 15 Menit Sebelum Masuk	Menampilkan notifikasi 15 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	    

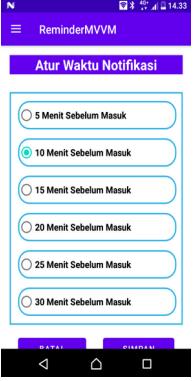
No.	Test Case	Acceptance Testing	Status	Hasil
13.	Memilih menu delay 20 Menit Sebelum Masuk	Menampilkan notifikasi 20 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	  

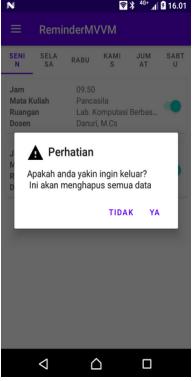
No.	Test Case	Acceptance Testing	Status	Hasil
14.	Memilih menu delay 25 Menit Sebelum Masuk	Menampilkan notifikasi 25 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 

No.	Test Case	Acceptance Testing	Status	Hasil
15.	Memilih menu delay 30 Menit Sebelum Masuk	Menampilkan notifikasi 30 menit sebelum masuk waktu perkuliahan yang akan berlangsung	Sesuai	 

No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
16,	Mematikan reminder	Tidak menampilkan notifikasi ketika jadwal perkuliahan sudah memasuki waktu nya	Sesuai	 

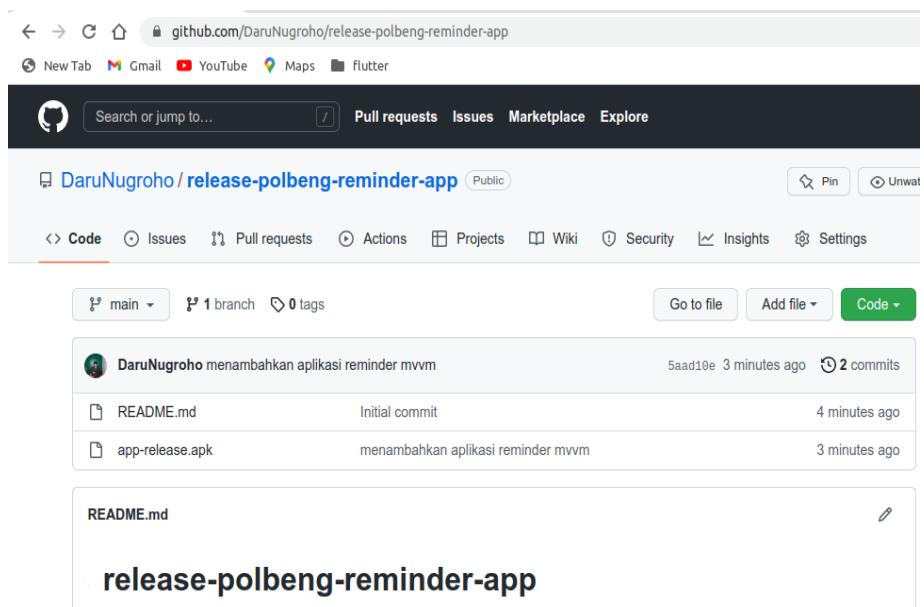
No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
17.	Klik pada notifikasi perkuliahan yang ditampilkan	Menampilkan halaman yang berisi informasi tentang mata kuliah yang akan berlangsung selanjutnya, dan terdapat tombol Dismiss untuk mematikan alarm	Sesuai	
18.	Mematikan perangkat dan menghidupkan lagi	Reminder tetap berjalan dan dapat menampilkan notifikasi perkuliahan saat sudah memasuki waktunya	Sesuai	

No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
19.	Memilih menu Atur Waktu Notifikasi	Menampilkan pilihan menu delay waktu notifikasi	Sesuai	
20.	Memilih kelas pada menu drop down	Dapat menyimpan kelas yang telah dipilih	Sesuai	

No.	<i>Test Case</i>	<i>Acceptance Testing</i>	Status	Hasil
21.	Memilih menu “Keluar”	Menampilkan alert dialog dan menghapus semua data jadwal perkuliahan yang tersimpan, ketika aplikasi dijalankan lagi maka akan menampilkan halaman pilihan tipe pengguna	Sesuai	 

3.2.5 Release (Perilisan)

Setelah tahap pengujian perangkat lunak dilakukan, tahap selanjutnya adalah melakukan perilisan aplikasi *reminder* jadwal perkuliahan. Pada tahap ini sistem sudah dalam bentuk final dan siap untuk digunakan oleh pengguna. Disini penulis menggunakan *Github* untuk menyimpan berkas *file apk* dari aplikasi *Reminder Jadwal Perkuliahan* agar bisa di *download* oleh pengguna. Berikut adalah alamat *Github* dari aplikasi Remider Jadwal Perkuliahan <https://github.com/DaruNugroho/release-polbeng-reminder-app>



Gambar 3.25. *Release* Aplikasi *Reminder Jadwal Perkuliahan*
Sumber : Data Olahan

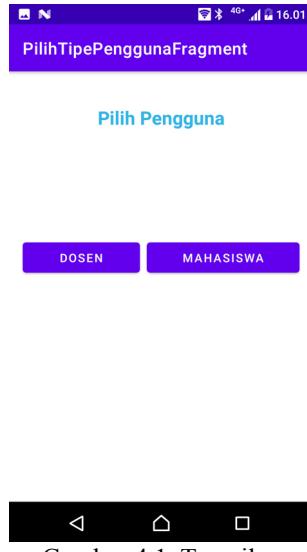
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Aplikasi *Reminder Jadwal Perkuliahan* telah berhasil dibuat dengan menerapkan arsitektur MVVM (*Model View ViewModel*). Dimana aplikasi ini akan menampilkan notifikasi perkuliahan apa bila mata kuliah telah memasuki waktu jadwalnya. Aplikasi ini dilengkapi juga dengan fitur alarm perkuliahan. Pada aplikasi ini terdapat juga fitur *on-off* reminder yang digunakan untuk mengaktifkan atau mematikan reminder perkuliahan, baik bagi dosen maupun mahasiswa, selain itu terdapat juga menu untuk atur waktu delay notifikasi, menu untuk memilih mata kuliah yang sedang berlangsung atau mata kuliah yang telah memasuki waktu jadwalnya. Berikut adalah tampilan aplikasi *Reminder Jadwal Perkuliahan*.

4.1.1 Tampilan Pilihan Tipe Pengguna



Gambar 4.1. Tampilan
Menu Tipe Pengguna
Sumber : Data Olahan

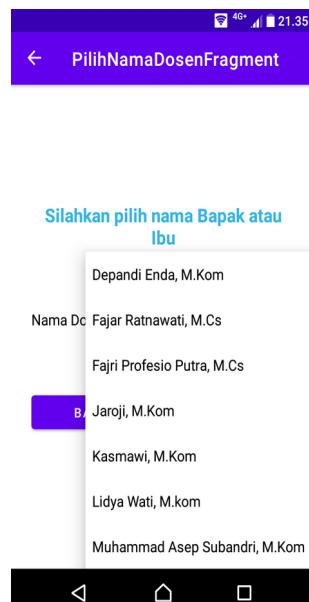
Pada tampilan aplikasi *Reminder* diatas akan menampilkan pilihan tipe pengguna yaitu Dosen dan Mahasiswa yang dapat dipilih salah satu oleh pengguna. Untuk *source code* tampilan menu tipe pengguna aplikasi *Reminder* dapat dilihat seperti dibawah ini.

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)

    binding.btnDosen.setOnClickListener {
        userType = UserType.Dosen
        val direction =
            PilihTipePenggunaFragmentDirections.actionPilihTipePenggunaFragmentToPilihNamaDosenFragm
            ent(userType)
        findNavController().navigate(direction)
    }

    binding.btnMahasiswa.setOnClickListener {
        userType = UserType.Mahasiswa
        val direction =
            PilihTipePenggunaFragmentDirections.actionPilihTipePenggunaFragmentToPilihJurusanSemesterFr
            agment(userType)
        findNavController().navigate(direction)
    }
}
```

4.1.2 Tampilan Menu Pilihan Nama Dosen



Gambar 4.2. Tampilan
Menu Nama Dosen
Sumber : Data Olahan

Pada tampilan aplikasi *Reminder* diatas terdapat menu *drop down* yang digunakan untuk menampilkan daftar nama dosen, dimana pada nama-nama dosen tersebut hanya dapat dipilih salah satu oleh pengguna dosen sesuai dengan namanya, dan aplikasi akan mengambil data jadwal perkuliahan sesuai dengan pilihan pengguna. Untuk *source code* tampilan menu pilihan nama dosen aplikasi *Reminder* dapat dilihat seperti dibawah ini.

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    binding.spnListDosen.apply {
        adapter = arrayAdapter
        onItemSelectedListener = this@PilihNamaDosenFragment
    }

    binding.btnBatal.setOnClickListener {
        val direction =
            PilihNamaDosenFragmentDirections.actionPilihNamaDosenFragmentToPilihTipePenggunaFragmen
            t()
        findNavController().navigate(direction)
    }

    binding.btnSimpan.setOnClickListener {
```

```

val success = viewModel.validate(userType)
if (success){
    viewModel.setUserType(userType)
    viewModel.addJadwalDosen(dosen) // masukkan jadwal dosen ke database
    val direction =
PilihNamaDosenFragmentDirections.actionPilihNamaDosenFragmentToMainActivity()
    val extras = ActivityNavigator.Extras.Builder()
        .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK)
        .addFlags(Intent.FLAG_ACTIVITY_NEW_TASK)
        .addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP)
        .build()
    findNavController().navigate(direction, extras)
}
}
}

```

4.1.3 Tampilan Jadwal Dosen



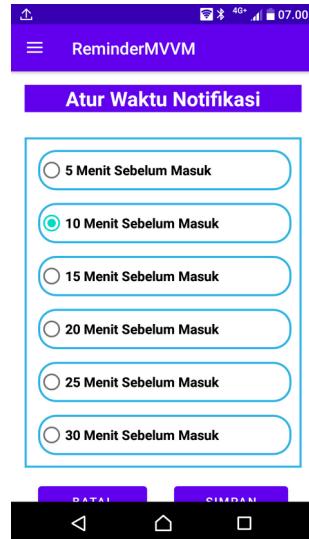
Gambar 4.3. Tampilan Jadwal Dosen

Sumber : Data Olahan

Pada tampilan aplikasi Reminder diatas digunakan untuk menampilkan jadwal dosen untuk hari senin dimana pada setiap *Card View* berisi informasi seperti jam mulai perkuliahan, mata kuliah, ruangan, akan mengajar dikelas berapa dan terdapat juga *Toggle* reminder yang digunakan untuk me nonaktifkan atau mengaktifkan reminder perkuliahan. Untuk *source code* tampilan fitur jadwal dosen aplikasi Reminder dapat dilihat seperti berikut ini.

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    if (viewModel.cekUserType()) {
        // untuk menampilkan jadwal dosen
        val adapter = ListJadwalDosenAdapter(this)
        rvJadwal = binding.rvJadwal
        rvJadwal.adapter = adapter
        viewModel.itemJadwalDosenSenin.observe(this.viewLifecycleOwner) { items ->
            items.let {
                Log.d(TAG_SENIN_FRAGMENT_DOSEN, it.toString())
                adapter.setData(it)
            }
        }
        rvJadwal.layoutManager = LinearLayoutManager(activity?.applicationContext)
    } else {
        // untuk menampilkan jadwal mahasiswa
        val adapter = ListJadwalMahasiswaAdapter(this)
        rvJadwal = binding.rvJadwal
        rvJadwal.adapter = adapter
        viewModel.itemJadwalMahasiswaSenin.observe(this.viewLifecycleOwner) { items ->
            items.let {
                Log.d(TAG_SENIN_FRAGMENT_MAHASISWA, it.toString())
                adapter.setData(it)
            }
        }
        rvJadwal.layoutManager = LinearLayoutManager(activity?.applicationContext)
    }
}
```

4.1.4 Tampilan Menu Pilihan Atur Waktu Notifikasi



Gambar 4.4. Tampilan
Menu Atur Waktu
Notifikasi
Sumber : Data Olahan

Pada tampilan aplikasi *Reminder* diatas digunakan untuk menampilkan opsi delay waktu untuk menampilkan notifikasi perkuliahan yang akan berlangsung. Untuk *source code* tampilan atur waktu *delay* notifikasi aplikasi *Reminder* dapat dilihat seperti berikut ini.

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)

    binding.btnSimpan.setOnClickListener {
        if (viewModel.cekUserType()) {
            viewModel.setWaktuDelayNotifikasiDosen(
                binding.rb5Mnt.isChecked,
                binding.rb10Mnt.isChecked,
                binding.rb15Mnt.isChecked,
                binding.rb20Mnt.isChecked,
                binding.rb25Mnt.isChecked,
                binding.rb30Mnt.isChecked,
            )
        } else {
            viewModel.setWaktuDelayNotifikasiMahasiswa(
                binding.rb5Mnt.isChecked,
                binding.rb10Mnt.isChecked,
                binding.rb15Mnt.isChecked,
                binding.rb20Mnt.isChecked,
                binding.rb25Mnt.isChecked,
                binding.rb30Mnt.isChecked,
            )
        }
    }
}
```

```

        }

when(viewModel.getDelayNotifPref()){
    5 -> binding.rb5Mnt.isChecked = true
    10 -> binding.rb10Mnt.isChecked = true
    15 -> binding.rb15Mnt.isChecked = true
    20 -> binding.rb20Mnt.isChecked = true
    25 -> binding.rb25Mnt.isChecked = true
    else -> binding.rb30Mnt.isChecked = true
}

```

4.1.5 Tampilan Fitur Mata Kuliah Yang Sedang Berlangsung



Gambar 4.5. Tampilan Fitur
Mata Kuliah Yang Sedang
Berlangsung
Sumber : Data Olahan

Pada tampilan aplikasi *Reminder* diatas digunakan untuk menampilkan mata kuliah yang sedang berlangsung atau sudah memasuki waktu jadwalnya dimana pada tampilan tersebut berisi informasi seperti jam mulai perkuliahan, mata kuliah, ruangan, dan akan mengajar dikelas berapa. Untuk *source code* tampilan fitur mata kuliah yang sedang berlangsung aplikasi *Reminder* dapat dilihat seperti berikut ini.

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    if (viewModel.isDataJadwalNotEmpty()) {
        viewModel.itemJadwalDosenSekarang.observe(this.viewLifecycleOwner) { items ->
            items.let {
                binding.tvItemMataKuliah.text = it.namaMataKuliah
                binding.tvItemJam.text = FormatWaktu.formatJam(it.jam, it.menit)
                binding.tvItemLab.text = it.ruangan
                binding.tvItemMengajar.text = it.kelas
            }
        }
    }
}
```

4.1.6 Tampilan Notifikasi Perkuliah



Gambar 4.6. Tampilan Fitur
Notifikasi Perkuliah
Sumber : Data Olahan

Pada tampilan notifikasi aplikasi *Reminder* diatas akan menampilkan notifikasi jadwal perkuliahan dimana pada notifikasi tersebut berisi informasi seperti jam mulai perkuliahan, mata kuliah, ruangan, dan akan mengajar dikelas berapa. Untuk *source code* tampilan fitur notifikasi perkuliahan aplikasi *Reminder* dapat dilihat seperti berikut ini.

```
private fun showNotification(jadwalDosen: JadwalDosen) {
    val intentToNextMatkulActivity = Intent(this, NextMataKuliahActivity::class.java)
    intentToNextMatkulActivity.flags = Intent.FLAG_ACTIVITY_NEW_DOCUMENT
    val pendingIntent = PendingIntent.getActivity(this, 0, intentToNextMatkulActivity, 0)
    val reminderPreferences = ReminderPreferences(application)

    val notification = NotificationCompat.Builder(this, ReminderApplication.CHANNEL_ID)
        .setSmallIcon(R.drawable.ic_baseline_notifications_active_24)
        .setContentTitle(getString(R.string.content_title_dosen_notification_service))
        .setStyle(
            NotificationCompat.BigTextStyle()
                .bigText(
                    getString(
                        R.string.content_dosen_notification_service,
                        reminderPreferences.getDelayNotification().toString(),
                        jadwalDosen.namaMataKuliah,
                        jadwalDosen.kelas,
                        jadwalDosen.ruangan,
                        FormatWaktu.formatJam(jadwalDosen.jam), jadwalDosen.menit
                    )
                )
        )
}
```

```

        )
        .setContentIntent(pendingIntent)
        .build()
    startForeground(1, notification)
}

```

4.1.7 Tampilan Alarm Jadwal Perkuliahan



Gambar 4.7. Tampilan Fitur

Alarm Jadwal Perkuliahan

Sumber : Data Olahan

Pada tampilan alarm aplikasi *Reminder* diatas akan menampilkan halaman yang berisi informasi seperti jam mulai perkuliahan, mata kuliah, ruangan, akan mengajar dikelas berapa dan terdapat sebuah tombol “*Dismis*” yang digunakan untuk mematikan alarm. Untuk *source code* tampilan fitur alarm jadwal perkuliahan aplikasi *Reminder* dapat dilihat seperti berikut ini.

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = DataBindingUtil.setContentView(this, R.layout.activity_next_mata_kuliah)

    if (viewModel.cekUserType()) {
        // tampilkan jadwal untuk dosen
        showMatKulDosen()
    } else {
        // tampilkan jadwal untuk mahasiswa
        showMatKulMahasiswa()
    }
}

```

```
binding.btnExit.setOnClickListener {
    if (viewModel.cekUserType()) {
        //dosen
        stopServiceDosen()
    } else {
        // mahasiswa
        stopServiceMahasiswa()
    }
}
```

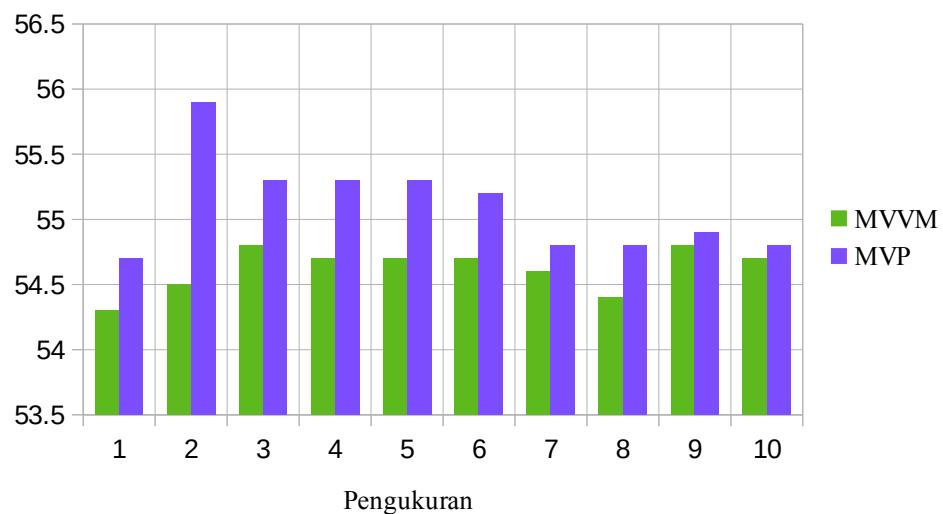
4.2 Pembahasan

Untuk melakukan pengukuran penggunaan RAM, penggunaan CPU, dan waktu eksekusi pada aplikasi Reminder Jadwal Perkuliahan yang telah dibuat dengan menggunakan dua arsitektur berbeda yakni MVVM dan MVP, penulis menggunakan *tools* pada *Android Studio* yakni *Android Profiler*, Sebagai berikut.

4.2.1 Penggunaan RAM

Pada pengukuran penggunaan RAM aplikasi Reminder Jadwal Perkuliahan yang menggunakan arsitektur MVVM penulis melakukan 10 kali pengukuran penggunaan RAM ketika aplikasi dijalankan untuk mengambil data dari API lalu menyimpannya ke *database* dan menampilkan data jadwal perkuliahan, dari 10 kali pengukuran didapat rata-rata penggunaan RAM sebesar 54,62 MB. Sedangkan untuk aplikasi Reminder Jadwal Perkuliahan yang menggunakan arsitektur MVP dari 10 kali pengukuran penggunaan RAM ketika aplikasi dijalankan untuk mengambil data dari API kemudian menampilkan data jadwal perkuliahan, didapat rata-rata penggunaan RAM sebesar 55,10 MB berdasarkan dari pengukuran penggunaan RAM yang telah dilakukan, arsitektur MVVM sedikit lebih rendah penggunaan RAM dibandingkan arsitektur MVP yaitu dengan selisih sebesar 0,48 MB, ini berarti arsitektur MVVM lebih unggul jika dibandingkan dengan arsitektur MVP. Berikut adalah hasil pengukuran penggunaan RAM dari kedua arsitektur tersebut.

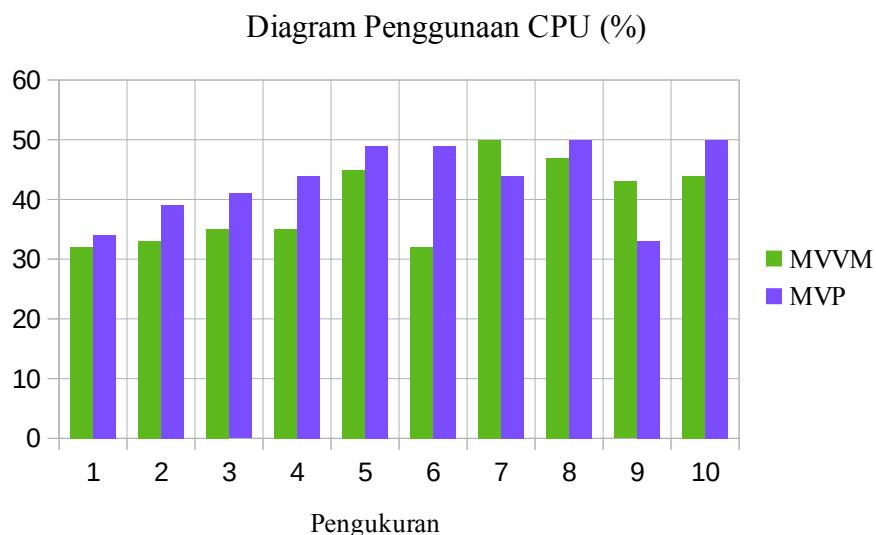
Diagram Penggunaan RAM (MB)



Gambar 4.8. Penggunaan RAM Arsitektur MVVM Dan MVP
Sumber : Data Olahan

4.2.2 Penggunaan CPU

Pada pengukuran penggunaan CPU aplikasi Reminder Jadwal Perkuliahan dengan menggunakan arsitektur MVVM penulis melakukan 10 kali pengukuran ketika aplikasi dijalankan untuk untuk mengambil data dari API lalu menyimpannya ke *database* dan menampilkan data jadwal perkuliahan, dari 10 kali pengukuran, di dapat rata-rata penggunaan penggunaan CPU sebesar 39.60 %. Sedangkan untuk aplikasi Reminder Jadwal Perkuliahan yang menggunakan arsitektur MVP dari 10 kali pengukuran dengan kasus yang sama di dapat rata-rata penggunaan penggunaan CPU sebesar 43.30 %. Berdasarkan dari pengukuran penggunaan CPU yang telah dilakukan, arsitektur MVVM lebih rendah penggunaan CPU dibandingkan dengan arsitektur MVP yaitu dengan selisih sebesar 3,7 %. Berikut adalah hasil pengukuran penggunaan CPU dari kedua arsitektur tersebut.



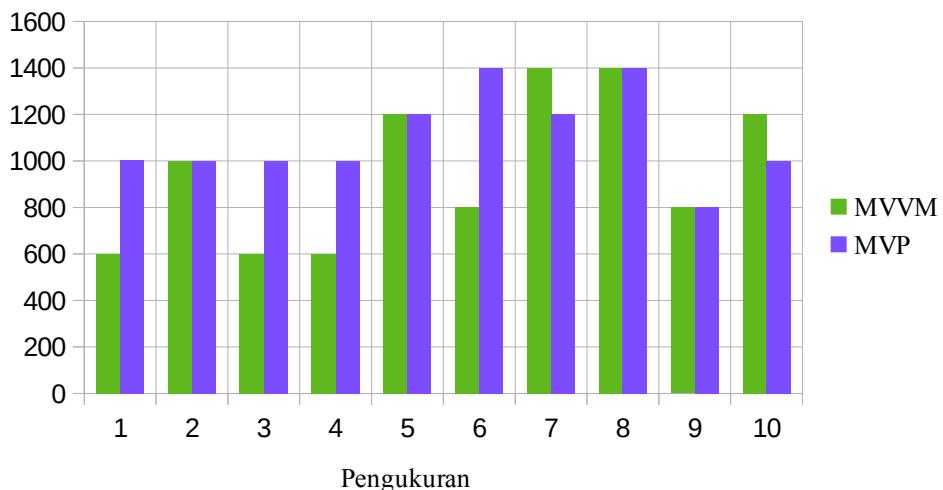
Gambar 4.9. Perbandingan Penggunaan CPU Arsitektur MVVM Dan MVP

Sumber : Data Olahan

4.2.3 Waktu Eksekusi

Pada pengukuran kecepatan waktu eksekusi aplikasi *Reminder Jadwal Perkuliahuan* dengan menggunakan arsitektur MVVM penulis melakukan 10 kali pengukuran ketika aplikasi dijalankan untuk mengambil data dari API lalu menyimpannya ke *database* dan menampilkan data jadwal perkuliahan, dari 10 kali pengukuran, di dapat rata-rata kecepatan waktu eksekusi sebesar 960 ms. Sedangkan untuk aplikasi *Reminder Jadwal Perkuliahuan* yang menggunakan arsitektur MVP dari 10 kali pengukuran dengan kasus yang sama, di dapat rata-rata kecepatan waktu eksekusi sebesar 1101 ms. Berdasarkan dari pengukuran penggunaan kecepatan waktu eksekusi yang telah dilakukan, arsitektur MVVM lebih cepat waktu eksekusi dibandingkan dengan arsitektur MVP yaitu dengan selisih sebesar 141 ms. Berikut adalah hasil pengukuran penggunaan CPU dari kedua arsitektur tersebut.

Diagram Kecepatan Waktu Eksekusi (ms)



Gambar 4.10. Pengukuran Kecepatan Waktu Eksekusi Arsitektur MVVM
Sumber : Data Olahan

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pengukuran yang telah penulis lakukan pada aplikasi *Reminder Jadwal Perkuliahan* dimana aplikasi tersebut dibuat dengan dua arsitektur berbeda yakni arsitektur MVVM dan MVP. saat dilakukan pengujian penggunaan RAM, CPU, dan kecepatan waktu eksekusi aplikasi *Reminder* dengan menggunakan arsitektur MVVM memiliki kelebihan dalam penggunaan RAM dan CPU yang rendah, serta memiliki waktu eksekusi lebih cepat dibandingkan dengan aplikasi *Reminder Jadwal Perkuliahan* yang menggunakan arsitektur MVP, ini dikarena kan pada arsitektur MVVM memiliki *library* tambahan seperti seperti *live data*, *data binding* dan *viewmodel* sehingga mampu mengurangi penggunaan *resource* pada perangkat *Android*. Sedangkan pada arsitektur MVP tidak memiliki *library* tambahan untuk menunjang performa aplikasi. Namun pada MVP terdapat penggunaan *contract class* dimana hal ini dapat dijadikan sebagai *guide line* dalam menentukan *method* yang ada pada *view* dan *presenter*, di sisi lain penggunaan *contract* memiliki kelemahan karena pengembang tidak dapat leluasa dalam membuat *method* didalam *view* atau *presenter*.

Dari kedua arsitektur tersebut, MVVM memilki performa yang lebih baik jika dibandingkan dengan arsitektur MVP ini karena arsitektur MVVM memiliki *library* tambahan seperti *live data*, *data binding* dan *viewmodel* sehingga mengurangi penggunaan *resource* pada perangkat *Android* dan cocok digunakan untuk mengembangkan aplikasi *Android* baik dalam skala kecil maupun kompleks Sedangkan pada arsitektur MVP tidak memiliki *library* tambahan untuk menunjang performa aplikasi tetapi terdapat penggunaan *contract class* yang dapat digunakan sebagai *guide line* ketika bekerja secara tim sehingga arsitektur cocok digunakan untuk mengembangkan aplikasi dari skala kecil hingga menengah.

5.2 Saran

Untuk meningkatkan fungsionalitas aplikasi *Reminder Jadwal Perkuliahan* terdapat beberapa saran peningkatan pada aplikasi ini sebagai berikut :

1. Aplikasi *Reminder* jadwal perkuliahan ini, berjalan lancar pada perangkat yang menjalankan sistem operasi *Android* 7.0 kedepannya dapat dikembangkan agar berjalan baik mulai dari versi *Android* 5.0 dan diatasnya.
2. Dapat ditambahkan fitur *reminder* jadwal ujian.
3. Dapat ditambahkan fitur *update* otomatis jadwal perkuliahan ketika aplikasi tersambung ke jaringan internet.

LAMPIRAN

Tabel 1. Jadwal Penelitian

No.	Jenis Kegiatan	Bulan																							
		1				2				3				4				5				6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	<i>Planning</i>																								
2.	<i>Design</i>																								
3.	<i>Coding</i>																								
4.	<i>Testing</i>																								
5.	Pembuatan laporan skripsi																								
6.	<i>Publish</i>																								

DAFTAR PUSTAKA

- Ahmad, I., Borman, R. I., Fakhrurozi, J., & Caksana, G. G. (2020). Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android. *INOVTEK Polbeng - Seri Informatika*, 5(2), 297. <https://doi.org/10.35314/isi.v5i2.1654>
- Arif, M. S., Musthafa, A., & Muriyatmoko, D. (2020). Implementation of Model-View-ViewModel (MVVM) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application. *Proceeding International Conference on Science and Engineering*, 3(November), 283–289. <https://doi.org/10.14421/icse.v3.514>
- Bose, S., Kundu, A., Mukherjee, M., & Banerjee, M. (2018). *A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT*. 9(3), 41–45.
- Cheng, Y., & Domínguez, A. O. (2019). *Advanced Android App Architecture*. Razeware LLC.
- Developer Training Team, G. (2016). Android Developer Fundamentals Course-Concept Reference. *CIRED - Open Access Proceedings Journal*, 2017(July), 1–67.
[http://www.eskom.co.za/CustomerCare/TariffsAndCharges/Documents/RSA Distribution Tariff Code Vers 6.pdf](http://www.eskom.co.za/CustomerCare/TariffsAndCharges/Documents/RSA%20Distribution%20Tariff%20Code%20Vers%206.pdf)
<http://www.nersa.org.za/>
- Erinton, R. Negara, R. Sanjoyo, D. (2017). Analisis Performasi Framework Codeigniter Dan Laravel Menggunakan Web Server Apache. *EProceedings of Engineering*, 4(3), 3565–3572.
<http://libraryeproceeding.telkomuniversity.ac.id/index.php/engineering/article/view/4965>
- FAN, J., ZENG, Y., & GAO, M. (2018). Mobile Data Monitoring System Based on MVP Mode. *DEStech Transactions on Computer Science and Engineering*, CCNT, 361–366.
<https://doi.org/10.12783/dtcse/ccnt2018/24726>

- I Made Sukarsa, I Nyoman Piarsa, & I Gede Bagus Premana Putra. (2020). Application of MVP Architecture in Developing Android-Based Seminar Ticket Booking Applications. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(3), 513–520. <https://doi.org/10.29207/resti.v4i3.1396>
- Muharram, M. K., Ariachandra, Z., Wisnuadhi, B., Munawar, G., & Kunci, K. (2021). *Analisis Pengaruh Arsitektur MVVM dan MVP pada Performa Database GreenDao*. 4–5.
- Muttaqin, M. A., Susanto, A., & Muslih. (2020). Aplikasi Stok Audit Di Pt . Media Sarana Data Berbasis Android Dengan Arsitektur Model View Presenter. *Jurnal Maklumatika*, 6(2), 112–123.
- Nagaraj, M., & Chethana, G. (2020). Android Application on MVVM Architecture to Change Database. *International Research Journal of Engineering and Technology*, 7(6), 1465–1469.
- Prabowo, G. (2018). A Tale of Two Development Approach : Empirical Study on The Maintainability and Modularity of Android Mobile Application with Anti-Pattern and Model-View-Presenter Design Pattern. *2018 International Conference on Electrical Engineering and Informatics (ICELTICs)(44501)*, 149–154.
- Rizki, B., Surya, P., Kharisma, A. P., & Yudistira, N. (2020). *Perbandingan Kinerja Pola Perancangan MVC , MVP , dan MVVM Pada Aplikasi Berbasis Android (Studi kasus : Aplikasi Laporan Hasil Belajar Siswa SMA BSS)*. 4(11).
- Supriyatna, A. (2018). Metode Extreme Programming Pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja. *Jurnal Teknik Informatika*, 11(1), 1–18. <https://doi.org/10.15408/jti.v11i1.6628>
- Suryn, W. (2014). Software Quality Engineering: A Practitioner's Approach. In *Software Quality Engineering: A Practitioner's Approach* (Vol. 9781118592). <https://doi.org/10.1002/9781118830208>
- Tone, K. (2018). *RANCANG BANGUN APLIKASI CLASS REMINDER BERBASIS ANDROID*. 148, 148–162.
- Wisnuadhi, B., Munawar, G., & Wahyu, U. (2020). *Performance Comparison of Native Android Application on MVP and MVVM*. 198(Issat), 276–282.