

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Pengembangan Produk

Produk yang dikembangkan merupakan suatu aplikasi notifikasi jadwal sekolah berbasis Android yang dapat diakses dimana dan kapan saja selama perangkat Android terhubung dengan internet. Aplikasi notifikasi jadwal sekolah ini diberi nama “**Skedi**” yang merupakan singkatan dari “*Schedule for study*”. Skedi dikembangkan menggunakan bahasa pemrograman Java dengan alat pengembangan Android Studio. Di dalam aplikasi tersebut memiliki jadwal mata pelajaran siswa yang sedang *Sign in* dan tugas-tugas yang dapat ditambahkan, serta terdapat notifikasi untuk peringatan *deadline* tugas maupun peringatan jadwal pelajaran hari selanjutnya dan mata pelajaran selanjutnya.

Pada pengembangannya, dibuat desain sistem, desain basis data, dan desain *User Interfaces* dan *User Experience* (UI/UX) terlebih dahulu. Dimana UX digambarkan dengan menggunakan diagram UML yaitu *use case diagram*, *activity diagram* dan *sequence diagram*. Aplikasi Skedi ini kemudian di uji kualitasnya menggunakan standar pengujian ISO/IEC 25010 dengan empat aspek menurut teori Assaf Ben David, yaitu aspek *functional suitability*, *performance efficiency*, *usability*, dan *compatibility*.

1. Komunikasi (*Communication*)

Tahap komunikasi adalah tahapan awal yang dilakukan dalam proses pengembangan aplikasi notifikasi jadwal sekolah. Tahapan komunikasi terdiri dari analisis kebutuhan fungsional, analisis kebutuhan perangkat keras dan perangkat lunak. Hasil dari tahap komunikasi yang telah dilakukan adalah sebagai berikut:

- 1) SMK Negeri 2 Depok memiliki 15 jurusan dengan masa studi 3 hingga 4 tahun.

- 2) Siswa-siswi SMK Negeri 2 Depok memiliki jadwal mata pelajaran yang berbeda pada setiap jurusan.
- 3) Tingkatan maupun jumlah kelas pada tiap-tiap jurusan berbeda.
- 4) Model kelas yang digunakan adalah *moving class*, dimana siswa akan berpindah pada setiap pergantian mata pelajaran.
- 5) Guru pengampu mata pelajaran teori dapat berasal dari guru jurusan, sehingga memungkinkan siswa tidak mengenal guru tersebut sebelumnya.
- 6) Beberapa kelas memiliki jam masuk dan pulang yang berbeda.
- 7) Mata pelajaran akan berubah ketika pergantian semester.
- 8) *Print out* jadwal mata pelajaran hanya dibagikan ketika pergantian semester.
- 9) Tugas setiap siswa pada suatu kelas dapat berbeda tergantung dari praktikum/mata pelajaran yang berlangsung.

Setelah permasalahan diketahui, dapat ditarik kesimpulan bahwa perlu diadakannya aplikasi notifikasi jadwal untuk menampilkan jadwal mata pelajaran yang terdapat di SMK Negeri 2 Depok. Selanjutnya didapatkan spesifikasi yang harus dimiliki aplikasi notifikasi jadwal sebagai berikut:

- 1) Pengguna dapat melihat jadwal mata pelajaran masing-masing sesuai kelas dan jurusannya.
- 2) Pengguna dapat melihat detail jadwal yaitu mata pelajaran, guru pengampu, kelas dan jam pelajaran.
- 3) Pengguna dapat menambahkan tugas pada tiap mata pelajaran yang diikuti.
- 4) Aplikasi memiliki notifikasi jadwal pelajaran perhari sebelum berangkat ke sekolah.
- 5) Aplikasi memiliki notifikasi setiap berpindah mata pelajaran.

- 6) Aplikasi memiliki notifikasi tugas ketika harus dikumpulkan.

Dari hasil kesimpulan tersebut kemudian dilakukan analisis kebutuhan untuk mengetahui kebutuhan yang diperlukan dalam pengembangan aplikasi. Hasil analisis kebutuhan tersebut adalah sebagai berikut:

a. Analisis kebutuhan Perangkat Keras dan Perangkat Lunak

Proses pengembangan Aplikasi notifikasi jadwal sekolah berbasis Android, membutuhkan perangkat untuk digunakan dalam proses pengembangan. Berikut adalah daftar spesifikasi perangkat yang digunakan:

1) Kebutuhan perangkat keras

Perangkat keras merupakan faktor yang penting untuk pengembangan perangkat lunak. Perangkat keras yang digunakan dalam pengembangan aplikasi notifikasi jadwal sekolah berbasis Android adalah satu buah laptop dan satu buah *smartphone* dengan spesifikasi seperti tertera pada Tabel 7 dan Tabel 8.

Tabel 1. Spesifikasi *Smartphone*

Komponen	Spesifikasi minimal	Spesifikasi Rekomendasi
Sistem operasi	Lollipop	Pie
Versi	5.0	9.0
RAM	256 MB	2 GB
CPU	800 MHz	1,3 GHz

Tabel 2. Spesifikasi Laptop

Komponen	Spesifikasi
<i>Processor</i>	Intel(R) Core(TM) i5-4210 U @ 1.70GHz (4 CPUs)
RAM	4 GB
<i>Hard Disk Drive</i>	500 GB
<i>Graphics</i>	Intel(R) HD Graphics Family
<i>Operating System</i>	Windows 10 Enterprise 64-bit
Merek	ASUS A455L

2) Kebutuhan perangkat lunak

Dalam pengembangan aplikasi notifikasi jadwal sekolah tentu perangkat lunak sangatlah penting untuk mengembangkan aplikasi dengan baik. Terdapat beberapa perangkat lunak yang digunakan untuk mengembangkan aplikasi notifikasi jadwal sekolah, antara lain:

- a) Android Studio 3.0 digunakan sebagai IDE (*Integrated development environment*) untuk membangun aplikasi berbasis Android.
- b) Android SDK digunakan sebagai kerangka kerja untuk membangun aplikasi Android.
- c) Star UML digunakan sebagai tools untuk membuat desain UML.
- d) MySQL Workbench digunakan sebagai tools untuk membuat desain basis data.
- e) Balsamiq digunakan sebagai tools untuk membuat desain *User Interface* (UI).
- f) *Server hosting* dan *domain* digunakan sebagai tempat untuk menyimpan data.

b. Analisis kebutuhan fungsional

Aplikasi notifikasi jadwal sekolah berbasis Android yang dikembangkan harus sesuai dengan kebutuhan yang telah di analisis, maka didapatkan kebutuhan fungsional minimum yang harus dimiliki oleh aplikasi notifikasi jadwal sekolah, yaitu:

- 1) Aplikasi bisa menampilkan halaman registrasi.
- 2) Aplikasi bisa menampilkan halaman login.
- 3) Aplikasi bisa menampilkan jadwal mata pelajaran.
- 4) Aplikasi bisa menampilkan detail jadwal mata pelajaran.
- 5) Aplikasi bisa menampilkan halaman tambah tugas.
- 6) Aplikasi bisa menampilkan halaman edit tugas.
- 7) Aplikasi bisa menampilkan halaman hapus tugas.

- 8) Aplikasi bisa menampilkan detail tugas.
- 9) Aplikasi bisa menampilkan tugas mata pelajaran.
- 10) Aplikasi bisa menampilkan halaman pengaturan alarm perhari.
- 11) Aplikasi bisa menampilkan halaman pengaturan alarm menit sebelum.
- 12) Aplikasi bisa menampilkan halaman profil.
- 13) Aplikasi bisa menampilkan halaman edit password.
- 14) Aplikasi bisa menampilkan halaman tentang aplikasi.

2. Perencanaan (*Planning*)

Tahap perencanaan dilakukan untuk membuat pedoman pembuatan aplikasi notifikasi jadwal sekolah berbasis Android adalah seperti Tabel 9:

Tabel 3. Tahap Perencanaan.

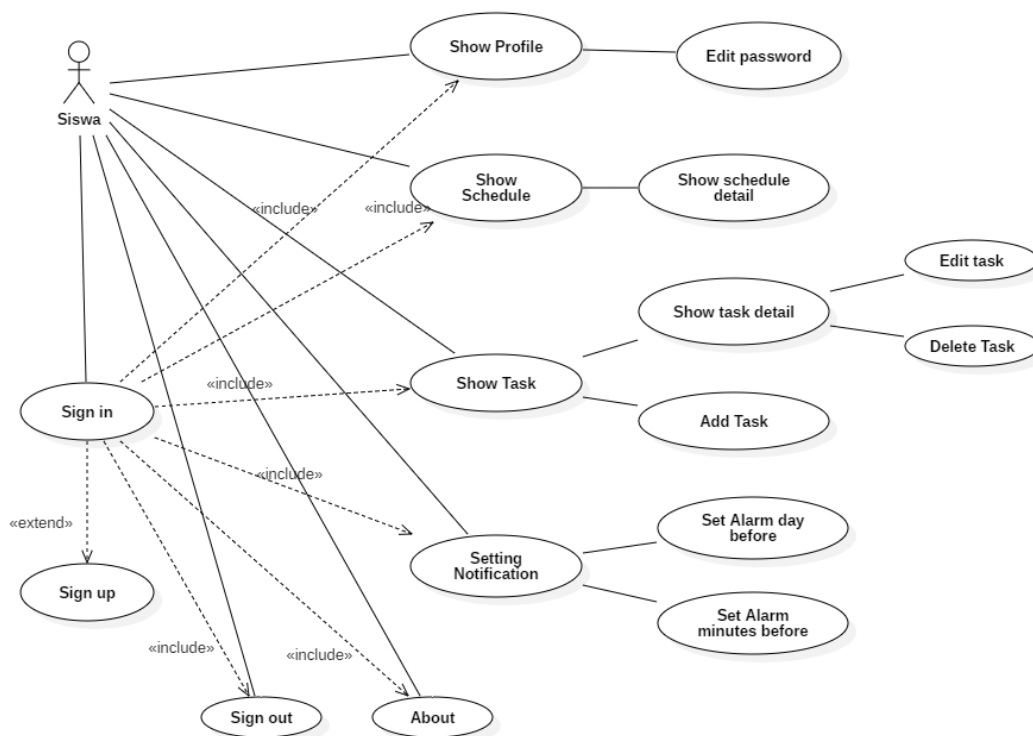
No	Nama Kegiatan	Durasi	Mulai	Berakhir
1	Analisis permasalahan	2 Minggu	15/10/2018	28/10/2018
2	Analisis kebutuhan	2 Minggu	1/11/2018	15/11/2018
3	Pemodelan dan pembuatan diagram UML	2 Minggu	19/11/2018	2/12/2018
4	Desain Database	2 Minggu	10/12/2018	23/12/2018
5	Pembuatan Desain UI (<i>User Interface</i>)	2 Minggu	7/01/2019	21/01/2019
6	Pembuatan aplikasi (<i>Coding dan Debugging</i>)	11 Minggu	01/02/2019	15/04/2019
7	Uji coba aplikasi	1 Minggu	20/04/2019	27/04/2019
8	Membangun aplikasi siap rilis dan merilis perangkat lunak ke <i>play store</i>	3 Hari	28/04/2019	30/04/2019

3. Pemodelan (*Modelling*)

Setelah tahapan analisis kebutuhan telah selesai, data yang didapatkan selanjutnya dituang dalam tahap desain. Tahapan desain yang dibuat berdasarkan analisis kebutuhan yang didapat adalah desain *User Experience* (UX) dengan menggunakan *Unified Modelling Language* (UML) yang terdiri dari *use case*, *sequence diagram*, dan *activity diagram* serta desain *User Interface* (UI).

a. Desain *Use Case Diagram*

Rancangan use case diagram berfungsi untuk menjabarkan fungsi-fungsi yang terdapat di dalam aplikasi beserta siapa saja yang dapat mengakses fungsi-fungsi tersebut. Pada Gambar 10 berikut adalah diagram *Use Case* dalam pengembangan aplikasi notifikasi jadwal sekolah:



Gambar 1. Use case Diagram

1) Definisi Aktor

Aktor berfungsi mendeskripsikan pengguna dalam aplikasi. Tabel 10 berikut adalah deskripsi pendefinisian aktor pada aplikasi notifikasi jadwal sekolah:

Tabel 4. Tabel definisi aktor

No	Aktor	Deskripsi
1.	Siswa	Orang yang menggunakan aplikasi

2) Definisi *Use case*

Pada Tabel 11 adalah deskripsi dari *use case* aplikasi notifikasi jadwal sekolah:

Tabel 5. Tabel Definisi Use case

No	Use Case	Deskripsi
1	Sign in	<i>Use case Sign in</i> berfungsi untuk menampilkan halaman login yang digunakan untuk user dapat masuk ke aplikasi notifikasi jadwal.
2	Sign up	<i>Use case Sign up</i> berfungsi untuk menampilkan halaman registrasi yang digunakan untuk user dapat mendaftar ke aplikasi notifikasi jadwal.
3	Show profile	<i>Use case Show profile</i> ini berfungsi untuk melihat profile user.
4	Edit password	<i>Use case Show profile</i> berfungsi untuk mengubah password yang telah diinput agar dapat di edit sesuai keinginan dari user.
5	Show Schedule	<i>Use case Show Schedule</i> berfungsi untuk melihat data grid jadwal mata pelajaran selama satu minggu.
6	Show Schedule Detail	<i>Use case Show Schedule Detail</i> berfungsi untuk melihat detail jadwal berupa jam pelajaran, nama mata pelajaran, guru pengampu, ruangan kelas.
7	Show Task	<i>Use case Show Task</i> berfungsi untuk melihat data grid tugas pelajaran yang telah diinput.

No	Use Case	Deskripsi
8	Show task detail	<i>Use case Show task detail</i> berfungsi untuk menampilkan detail tugas berupa nama mata pelajaran, keterangan tugas, waktu tugas.
9	Edit Task	<i>Use case Edit Task</i> berfungsi untuk mengubah data tugas yang telah diinput agar dapat diedit sesuai keinginan user.
10	Delete Task	<i>Use case Delete Task</i> berfungsi untuk menghapus data tugas yang sudah diinput ketika sudah selesai atau jika terjadi kesalahan input.
11	Add Task	<i>Use case Add Task</i> berfungsi untuk melakukan tambah tugas di dalam aplikasi yang dilakukan oleh user
12	Setting Notification	<i>Use case Setting Notification</i> berfungsi untuk melihat data list set alarm notifikasi.
13	Set Alarm day before	<i>Use case Set Alarm day before</i> berfungsi untuk mengatur alarm notifikasi sehari sebelum jadwal pelajaran berlangsung
14	Set alarm minutes before	<i>Use case Set alarm minutes before</i> berfungsi untuk mengatur alarm notifikasi yang muncul sebelum suatu mata pelajaran berlangsung
15	About	<i>Use case About</i> berfungsi untuk melihat tentang informasi aplikasi.
16	Sign out	<i>Use case Sign out</i> berfungsi untuk mengeluarkan user dari aplikasi.

3) Skenario *Use case*

Skenario *use case* merupakan alur jalannya bagaimana proses *use case* dari sisi aktor menjalankan suatu fungsi tertentu dan bagaimana sistem merespon setiap aktivikasi yang dilakukan. Berikut adalah skenario use case dari aplikasi notifikasi jadwal sekolah:

a) Skenario membuka aplikasi

Tabel 6. Skenario membuka aplikasi

Aksi aktor	Reaksi sistem
Membuka aplikasi	
	Menampilkan halaman intro slider
Menggeser halaman intro	
	Menampilkan halaman intro slider
Menggeser halaman intro	
	Menampilkan halaman intro dan tombol masuk ke aplikasi
Menekan tombol “Mulai”	
	Menampilkan halaman sign in

b) Skenario *Sign in*

Tabel 7. Skenario *Sign in*

Aksi aktor	Reaksi sistem
	Menampilkan halaman sign in
Mengisi Nis dan Password	
	Record data Nis dan password
Menekan tombol “Masuk”	
	Mengirim informasi username dan password, dan mengecek di dalam database nis dan password sudah benar dan terdaftar atau belum.

c) Skenario Sign up

Tabel 8. Skenario Sign up

Aksi aktor	Reaksi sistem
Menekan tombol “Daftar”	
	Menampilkan halaman sign up

Aksi aktor	Reaksi sistem
Mengisi Nis, nama siswa, alamat, kelas, password, konfirmasi password	
	Record data Nis dan password
Menekan tombol “Daftar”	
	Mengirim informasi Nis, nama siswa, alamat, kelas, password, dan mengecek di dalam database sudah benar dan terdaftar atau belum.

d) Skenario *Show Profile*

Tabel 9. Skenario *Show Profile*

Aksi aktor	Reaksi sistem
Menekan tombol menu utama	
	Menampilkan daftar menu utama
Menekan menu “Profil”	
	Memunculkan data profil siswa

e) Skenario *Edit password*

Tabel 10. Skenario *Edit Password*

Aksi aktor	Reaksi sistem
Menekan icon pada daftar password	
	Menampilkan halaman password
Mengisi password lama dan password baru	
	Record data password lama dan password baru
Menekan tombol “Simpan”	
	Mengirim informasi password lama dan baru ke database dan mengecek apakah

Aksi aktor	Reaksi sistem
	password yang dimasukkan benar atau salah.

f) Skenario *Show Schedule*

Tabel 11. Skenario *Show Schedule*

Aksi aktor	Reaksi sistem
Menekan tombol menu utama	
	Menampilkan daftar menu utama
Menekan menu “Jadwal”	
	Memunculkan halaman berupa data jadwal mata pelajaran

g) Skenario *Show Schedule Detail*

Tabel 12. Skenario *Show Schedule Detail*

Aksi aktor	Reaksi sistem
Menekan salah satu mata pelajaran	
	Menampilkan halaman detail mata pelajaran berupa nama mata pelajaran, guru pengampu, ruang kelas, waktu pelajaran

h) Skenario *Show Task*

Tabel 13. Skenario *Show Task*

Aksi aktor	Reaksi sistem
Menekan tombol menu utama	
	Menampilkan daftar menu utama
Menekan menu “Tugas”	

	Memunculkan halaman berupa daftar tugas
--	---

i) Skenario *Show Task Detail*

Tabel 14. Skenario *Show Task Detail*

Aksi aktor	Reaksi sistem
Menekan salah satu tugas	
	Menampilkan halaman detail tugas berupa nama mata pelajaran, catatan tugas, waktu tugas

j) Skenario *Edit Task*

Tabel 15. Skenario *Edit Task*

Aksi aktor	Reaksi sistem
Menekan icon edit pada halaman detail tugas	
	Menampilkan halaman detail tugas berupa nama mata pelajaran, catatan tugas, tanggal tugas, waktu alarm
Mengubah data tugas	
	Record data baru
Menekan tombol “Simpan”	
	Mengirim data baru yang sudah diubah

k) Skenario *Delete task*

Tabel 16. Skenario *Delete task*

Aksi aktor	Reaksi sistem
Menekan icon hapus pada halaman detail tugas	

Aksi aktor	Reaksi sistem
	Menampilkan halaman peringatan hapus
Menekan tombol “Ya”	
	Menghapus data tugas

l) Skenario *Add Task*

Tabel 17. Skenario *Add Task*

Aksi aktor	Reaksi sistem
Menekan <i>icon plus</i> pada halaman tugas	
	Menampilkan halaman tambah tugas.
Mengisi nama mata pelajaran, catatan tugas, tanggal tugas, waktu alarm	
	Record data nama mata pelajaran, catatan tugas, tanggal tugas, waktu alarm.
Menekan tombol simpan	
	Mengirim data baru.

m) Skenario *Setting Notification*

Tabel 18. Skenario *Setting Notification*

Aksi aktor	Reaksi sistem
Menekan tombol menu utama	
	Menampilkan daftar menu utama
Menekan menu “Notifikasi”	
	Memunculkan halaman berupa daftar Alarm.

n) Skenario *Set alarm day before*

Tabel 19. Skenario *Set alarm day before*

Aksi aktor	Reaksi sistem
Menekan tombol Hari Sebelum	
	Menampilkan daftar hari
Memilih daftar hari	
	Menampilkan menu atur waktu alarm
Menekan tombol “On”	
	Mengaktivasi waktu alarm untuk bisa di ubah
Mengubah waktu alarm	
	Record data waktu alarm
Menekan tombol “Simpan	
	Mengirim data baru yang sudah dibuat.

o) Skenario *Set alarm minutes before*

Tabel 20. Skenario *Set alarm minutes before*

Aksi aktor	Reaksi sistem
Menekan tombol Menit Sebelum	
	Menampilkan daftar menit
Memilih waktu alarm	
	Record data waktu alarm
Menekan tombol “Simpan	
	Mengirim data baru yang sudah dibuat.

p) Skenario *About*

Tabel 21. Skenario *About*

Aksi aktor	Reaksi sistem
Menekan tombol menu utama	

Aksi aktor	Reaksi sistem
	Menampilkan daftar menu utama
Menekan menu “Tentang”	
	Menampilkan halaman tentang aplikasi.

q) Skenario Log out

Tabel 22. Skenario Log out

Aksi aktor	Reaksi sistem
Menekan tombol menu utama	
	Menampilkan daftar menu utama
Menekan menu “Keluar”	
	Keluar dari menu aplikasi dan kembali ke halaman sign in.

b. Desain *Sequence Diagram*

Setelah hasil skenario *use case* dibuat, maka kemudian dapat dihasilkan *sequence diagram*. Berikut merupakan daftar *sequence diagram* yang dibuat untuk pengembangan aplikasi notifikasi jadwal sekolah terdapat pada Tabel 29:

Tabel 23. Daftar Sequence diagram

No	Sequence Diagram	Skenario
1	Sign in	Sign in
2	Sign up	Sign up
3	Show profile	Show profile
4	Edit password	Edit password
5	Show schedule	Show schedule
6	Show schedule detail	Show schedule detail
7	Show task	Show task
8	Show task detail	Show task detail
9	Edit task	Edit task
10	Delete task	Delete task

No	Sequence Diagram	Skenario
11	Add task	Add task
12	Setting notification	Setting notification
13	Set alarm day before	Set alarm day before
14	Set alarm minutes before	Set alarm minutes before
15	About	About
16	Sign out	Sign out

Adapun *Sequence diagram* yang digunakan dalam pengembangan aplikasi Notifikasi Jadwal Sekolah terdapat pada Lampiran 8.

c. **Desain Activity Diagram**

Setelah interaksi dari suatu sistem digambarkan, maka perilaku dinamis dari sebuah sistem dapat dituang kedalam *Activity diagram*. Berikut daftar *activity diagram* yang dibuat untuk pengembangan aplikasi notifikasi jadwal sekolah pada Tabel 30:

Tabel 24. Daftar Activity Diagram

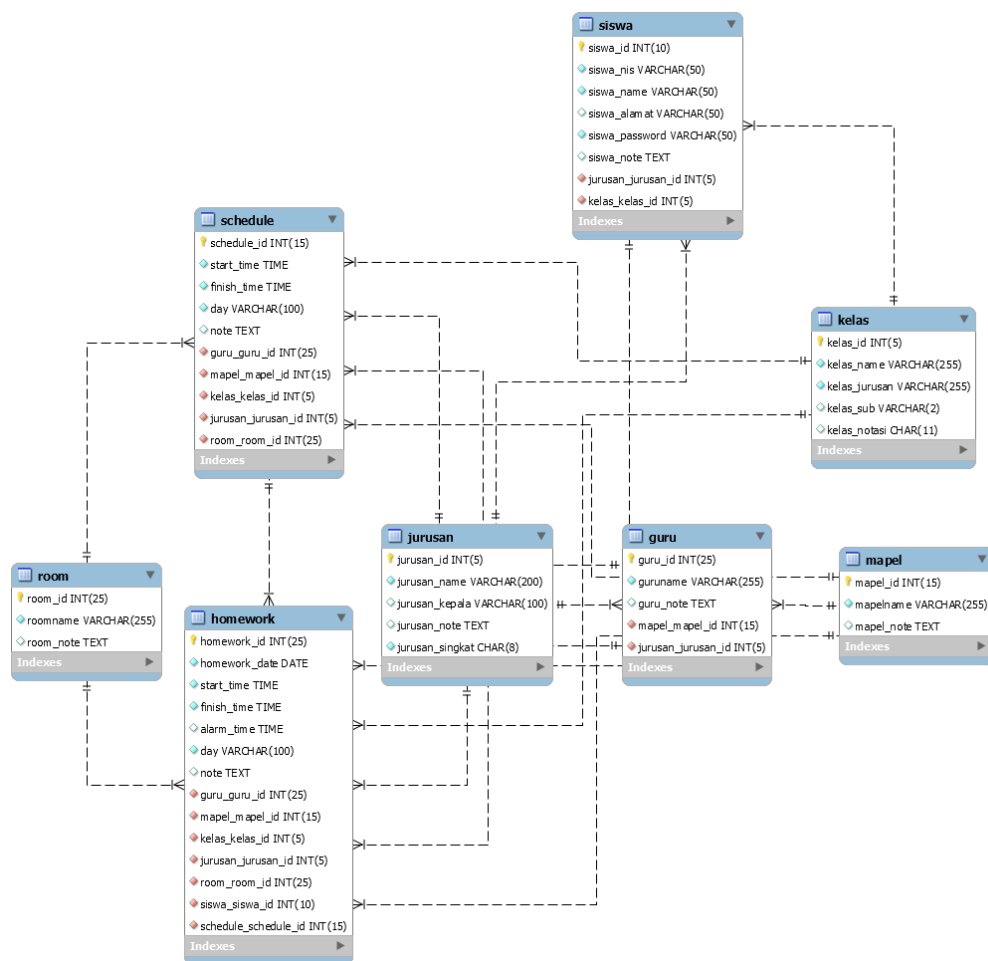
No	Sequence Diagram	Skenario
1	<i>Sign in</i>	<i>Sign in</i>
2	<i>Sign up</i>	<i>Sign up</i>
3	<i>Show profile</i>	<i>Show profile</i>
4	<i>Edit password</i>	<i>Edit password</i>
5	<i>Show schedule</i>	<i>Show schedule</i>
6	<i>Show schedule detail</i>	<i>Show schedule detail</i>
7	<i>Show task</i>	<i>Show task</i>
8	<i>Show task detail</i>	<i>Show task detail</i>
9	<i>Edit task</i>	<i>Edit task</i>
10	<i>Delete task</i>	<i>Delete task</i>
11	<i>Add task</i>	<i>Add task</i>
12	<i>Setting notification</i>	<i>Setting notification</i>
13	<i>Set alarm day before</i>	<i>Set alarm day before</i>
14	<i>Set alarm minutes before</i>	<i>Set alarm minutes before</i>

No	Sequence Diagram	Skenario
15	About	About
16	Sign out	Sign out

Adapun *Activity diagram* yang digunakan dalam pengembangan aplikasi Notifikasi Jadwal Sekolah terdapat pada Lampiran 9.

d. Desain Basis Data

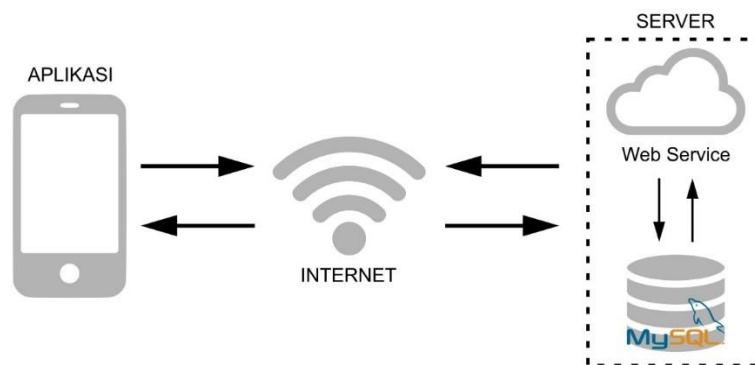
Pengembangan aplikasi notifikasi jadwal sekolah menggunakan sebuah basis data di dalam web server. Basis data dirancang menggunakan suatu alat yang bernama *MySQL Workbench*. Adapun hasil dari desain basis data yang digunakan untuk pengembangan aplikasi ini dapat dilihat pada Gambar 11:



Gambar 2. Desain Basis Data Aplikasi

e. Desain Sistem

Aplikasi Notifikasi Jadwal Sekolah di desain supaya dapat memperbarui data secara otomatis di dalam aplikasinya. Proses sinkronasi dilakukan dengan basis data server melalui internet untuk mendapatkan data yang dibutuhkan. Hal ini menyebabkan dibutuhkan interaksi antar aplikasi dengan server. Adapun desain sistem aplikasi Notifikasi Jadwal Sekolah dapat dilihat pada Gambar 12:



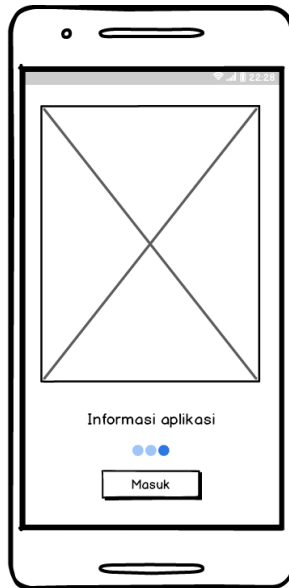
Gambar 3. Desain Sistem

f. Desain *User Interface (UI)*

Desain antarmuka dari aplikasi notifikasi jadwal sekolah masih berbentuk kerangka kasar. Adapun tampilan antarmuka aplikasi notifikasi jadwal sekolah adalah sebagai berikut:

1) Halaman *Highlight feature and benefit* Aplikasi

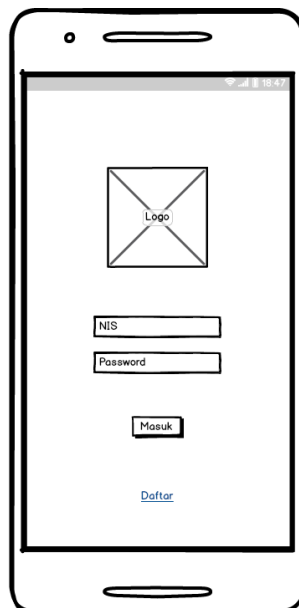
Halaman saat aplikasi pertama kali digunakan akan muncul gambar yang berisi informasi mengenai aplikasi. Halaman ini berbentuk *image slider* yang diakhiri dengan sebuah tombol untuk masuk ke menu utama aplikasi. Desain tampilan *Highlight feature and benefit* dapat dilihat pada Gambar 13:



Gambar 4. Desain *Interface Highlite and feature*

2) Halaman *Sign in* Aplikasi

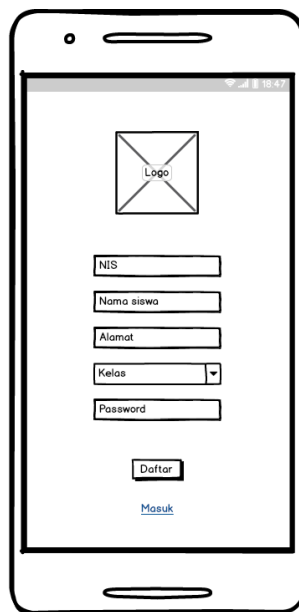
Halaman setelah *highlite and feature* yaitu halaman *Sign in* yang terdiri dari logo aplikasi, *text input* untuk memasukkan Nomor Induk Siswa (NIS), password, dan tombol login, serta terdapat link untuk mendaftar aplikasi apabila siswa belum terdaftar. Desain tampilan *Sign in* dapat dilihat pada Gambar 14:



Gambar 5. Desain *Interface Sign in* aplikasi

3) Halaman *Sign Up* Aplikasi

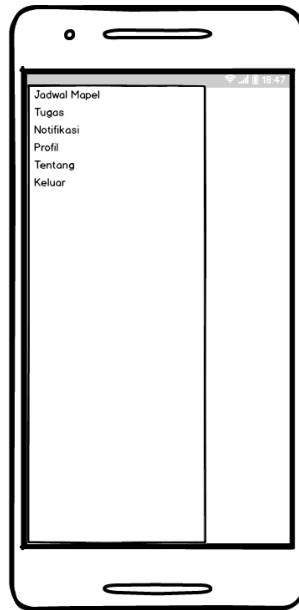
Halaman saat siswa meng-klik link daftar yaitu halaman *Sign up* yang terdiri dari logo aplikasi, *text input* untuk memasukkan Nomor Induk Siswa (NIS), nama siswa, alamat siswa, *combobox* kelas siswa, password, dan tombol daftar, serta terdapat link untuk *Sign in* aplikasi apabila siswa sudah terdaftar. Desain tampilan *Sign up* dapat dilihat pada Gambar 15:



Gambar 6. Desain *Interface Sign up* aplikasi

4) Menu utama aplikasi dengan *Side bar*

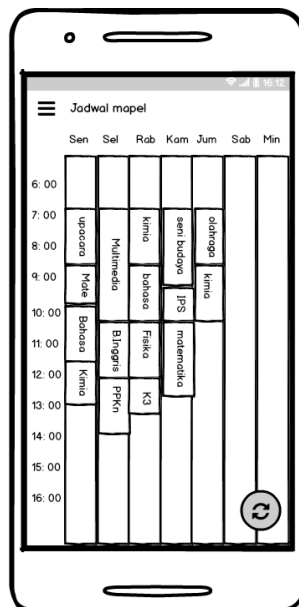
Saat aplikasi berada pada *side bar* maka menampilkan menu utama aplikasi. Halaman ini berisikan menu jadwal, tugas, notifikasi, profil, tentang dan keluar. Tampilan *side bar* yang menampilkan menu utama dapat dilihat pada Gambar 16:



Gambar 7. Desain *interface* menu utama aplikasi

5) Halaman Jadwal Mata Pelajaran

Halaman jadwal berisikan informasi jadwal mata pelajaran yang berbentuk datagrid selama satu minggu dimulai hari senin hingga hari minggu, halaman ini juga terdapat tombol refresh untuk mengambil data jadwal apabila terjadi perubahan jadwal dari basis data server. Tampilan jadwal dapat dilihat pada Gambar 17:



Gambar 8. Desain *Interface* Jadwal

6) Halaman Detail jadwal mata pelajaran

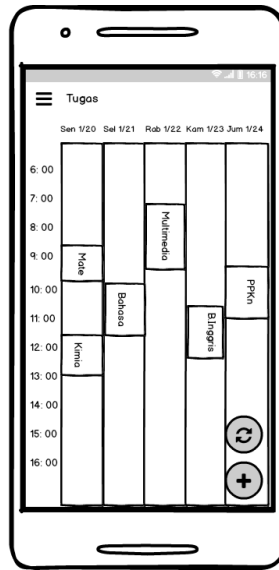
Halaman detail jadwal berisikan informasi jadwal mata pelajaran yang menampilkan informasi lengkap dari suatu jadwal, yaitu terdapat informasi nama mata pelajaran, guru pengampu, waktu, dan ruangan. Pada halaman ini juga terdapat tombol tambah untuk menambahkan tugas apabila siswa memiliki tugas dari guru pengampu. Tampilan detail jadwal dapat dilihat pada Gambar 18:



Gambar 9. Desain *Interface* Detail Jadwal

7) Halaman Tugas

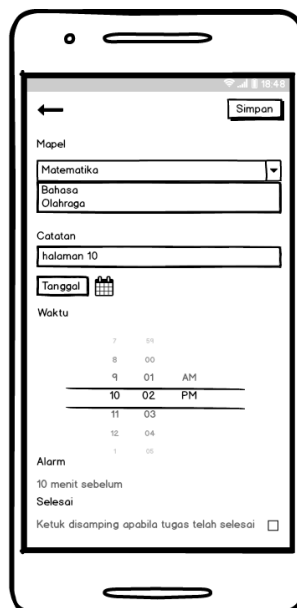
Halaman tugas berisikan informasi tugas yang dimiliki oleh siswa, halaman ini menampilkan informasi tugas, waktu *deadline* yang berbentuk datagrid dan tombol refresh untuk mengambil data tugas dari server, serta tombol tambah apabila siswa ingin menambahkan tugas. Tampilan tugas dapat dilihat pada Gambar 19:



Gambar 10. Desain *Interface* Halaman Tugas

8) Halaman Edit dan Tambah tugas

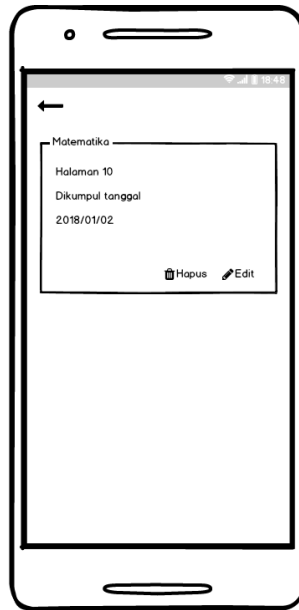
Halaman edit dan tambah tugas pada aplikasi sama peris, yaitu berisikan *combobox* mata pelajaran, *text input* catatan, waktu *deadline* tugas dengan model inputan *datechooser* dan *time picker*, alarm dengan inputan *radiobutton*, *checkbox* apabila tugas telah selesai. Adapun tampilan untuk halaman tambah tugas dapat dilihat pada Gambar 20:



Gambar 11. Desain *Interface* halaman Tambah Tugas

9) Halaman Detail Tugas

Halaman detail tugas pada aplikasi berisikan informasi tugas yang berupa mata pelajaran tugas, waktu *deadline* tugas, tombol edit dan tombol hapus tugas. Adapun tampilan untuk halaman detail tugas dapat dilihat pada Gambar 21:



Gambar 12. Desain *Interface* halaman Detail Tugas

10) Halaman hapus tugas

Pada halaman hapus tugas, berisikan warning alert yang mengingatkan kembali apakah tugas akan benar-benar dihapus, terdapat dua pilihan ya dan tidak pada tampilan tersebut. Adapun tampilan pada hapus tugas dapat dilihat pada Gambar 22:



Gambar 13. Desain *Interface* halaman Hapus Tugas

11) Halaman Notifikasi

Pada halaman Notifikasi, berisikan pengingat jadwal mata pelajaran yang dapat di ubah sesuai kebutuhan siswa. Adapun alarm notifikasi dapat di atur satu hari sebelum jadwal terlaksana dan menit sebelum mata pelajaran berlangsung. Tampilan pada halaman notifikasi dapat dilihat pada Gambar 23:



Gambar 14. Desain *Interface* halaman Notifikasi

12) Halaman *Set alarm day before*

Pada halaman *Set alarm day before*, terdapat daftar hari dari senin hingga sabtu dimana jadwal berlangsung. Siswa dapat memilih mengaktifkan notifikasi atau tidak dengan mengklik salah satu hari, apabila notifikasi diaktifkan maka akan dapat memilih waktu alarm yang akan muncul 1 hari sebelumnya menggunakan *time picker*. Tampilan pada halaman *Set alarm day before* dapat dilihat pada Gambar 24:



Gambar 15. Desain *Interface* halaman *Set alarm day before*

13) Halaman *Set alarm minutes before*

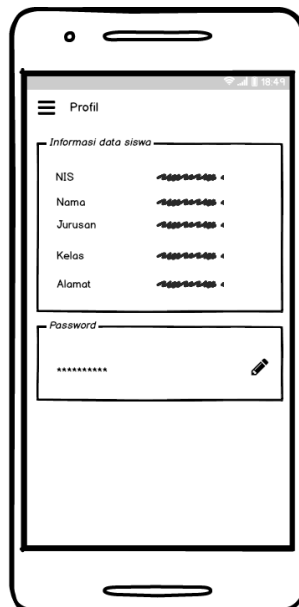
Pada halaman *Set alarm minutes before*, terdapat *radiobutton* dengan pilihan alarm didalamnya yaitu alarm untuk memunculkan notifikasi beberapa menit sebelum mata pelajaran berlangsung. Tampilan pada halaman *Set alarm minutes before* dapat dilihat pada Gambar 25:



Gambar 16. Desain *Interface* halaman *Set alarm minutes before*

14) Halaman Profil

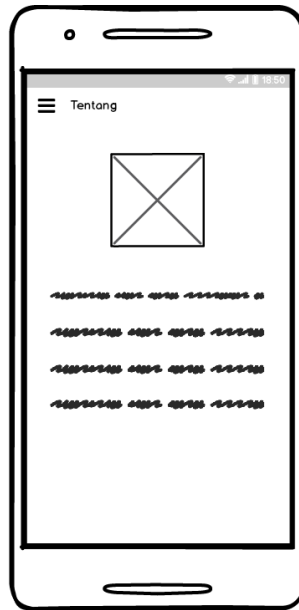
Pada halaman *Profile*, terdapat informasi lengkap siswa yaitu NIS, nama, jurusan, kelas, alamat, dan password, serta tombol edit password. Tampilan pada halaman *Profile* dapat dilihat pada Gambar 26:



Gambar 17. Desain *Interface* halaman *Profile*

15) Halaman Edit *Password*

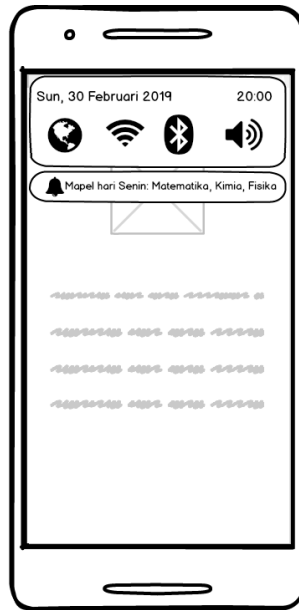
Pada halaman Edit *Password*, terdapat *text input* untuk *password* lama dan *password* baru serta tombol simpan dan tombol untuk keluar. Tampilan pada halaman edit *password* dapat dilihat pada Gambar 27:



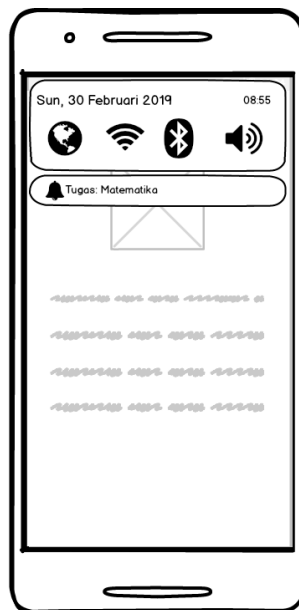
Gambar 18. Desain *Interface* halaman Edit *Password*

16) Tampilan untuk alarm notifikasi jadwal dan tugas

Pada tampilan ini notifikasi alarm akan muncul pada *notification bar* di perangkat yang digunakan oleh siswa. Sehingga akan membawa masuk ke dalam aplikasi. Adapun tampilan untuk notifikasi pada *notification bar* dapat dilihat pada Gambar 28 dan Gambar 29:



Gambar 19. Tampilan notifikasi jadwal pada *notification bar*



Gambar 20. Tampilan notifikasi tugas pada *notification bar*

4. Konstruksi (*Construction*)

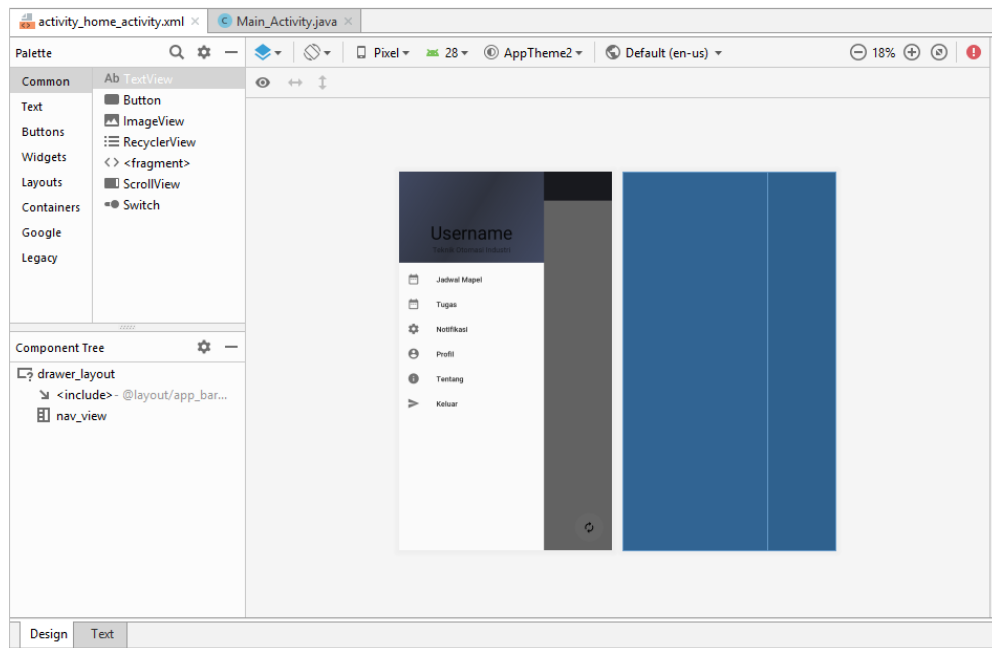
Tahap kontruksi (*construction*) adalah tahap yang dilakukan setelah tahap pemodelan telah selesai dibuat. Pada tahap ini, model yang telah dibuat direalisasikan ke dalam bentuk nyata suatu sistem perangkat lunak. Tahap ini terdiri dari

implementasi *layout* aplikasi, implementasi pemrograman dan implementasi basis data.

a. Membangun perangkat lunak

1) Implementasi *layout* aplikasi

Proses implementasi *layout* aplikasi adalah mewujudkan desain yang telah dibuat ke dalam bentuk UI/UX aplikasi. Dalam pengembangan aplikasi notifikasi jadwal sekolah, *layout* dibuat dengan bahasa XML dengan menggunakan *designer tool* yang telah disediakan oleh Android Studio. *Design view* yang dikembangkan dapat dilihat pada Gambar 30:



Gambar 21. Design View

Adapun daftar layout aplikasi notifikasi jadwal sekolah terdapat pada Tabel 31:

Tabel 25. Daftar layout aplikasi

No.	Layout	Fungsi
1	Activity_onboard_.xml	Pada layout ini halaman <i>highlite features and benefit</i> ditampilkan

No.	Layout	Fungsi
2	Activity_main.xml	Pada layout ini halaman Sign In ke dalam aplikasi ditampilkan.
3	Activity_sign_up_user.xml	Pada layout ini halaman Sign Up ke dalam aplikasi ditampilkan.
4	Activity_home_activity.xml	Pada layout ini halaman jadwal mapel ditampilkan.
5	Activity_detail_schedule.xml	Pada layout ini halaman detail jadwal mapel ditampilkan.
6	Activity_home_work.xml	Pada layout ini untuk menampilkan halaman tugas ditampilkan.
7	Activity_detail_home_work_activity.xml	Pada layout ini halaman tugas ditampilkan
8	Activity_add_home_work.xml	Pada layout ini halaman tambah tugas ditampilkan.
9	Activity_edit_homework.xml	Pada layout ini halaman edit tugas ditampilkan.
10	Activity_profile_activity.xml	Pada layout ini halaman data profil siswa ditampilkan.
11	Activity_setting.xml	Pada layout ini halaman <i>setting</i> notifikasi aplikasi ditampilkan.
12	Activity_day_before.xml	Pada layout ini halaman notifikasi alarm hari sebelum pada aplikasi ditampilkan.
13	Popup_minut.xml	Pada layout ini halaman notifikasi alarm menit sebelum pada aplikasi ditampilkan.
14	Activity_about.xml	Pada layout ini halaman tentang aplikasi ditampilkan.
15	Activity_home_activity_drawer.xml	Pada layout ini halaman menu pada <i>side bar</i> ditampilkan.

Kemudian hasil implementasi model pada tahap pemodelan ke dalam *interface* di Android Studio dapat dilihat pada Lampiran 10.

2) Implementasi pemrograman

Setelah selesai membuat layout aplikasi, maka setiap layout yang sudah dibuat dikontrol dengan menggunakan file java sehingga layout tersebut dapat menjadi layout yang dinamis. Pada Gambar 31 adalah potongan kode program yang dikembangkan:

```

67
68
69 @Override
70 protected void onCreate(Bundle savedInstanceState) {
71     super.onCreate(savedInstanceState);
72     setContentView(R.layout.activity_main);
73     ButterKnife.bind(target: this);
74     baseApiService = utilsApi.getApiServices();
75     sessionManager = new SessionManager(context: Main_Activity.this);
76     sessionDetailsSchedule = new SessionDetailsSchedule(context: this);
77     if (sessionManager.getSpSessionlogin()) {
78         startActivity(new Intent(context: Main_Activity.this, Home_activity.class).
79             addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK));
80         finish();
81     }
82     handler.postDelayed(runnable, delayMillis: 2000); //2000 is the timeout for the splash
83
84     showPassword.setOnCheckedChangeListener((compoundButton, value) -> {
85         if (value) edPassword.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
86         else edPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());
87     });
88
89 }
90
91
92
93 @OnClick(R.id.btn_login)
94 void btnLogin() { loginRequest(); }
95
96
97 @OnClick(R.id.btn_SignUp)
98 void btnSignUp() {
99     // Toast.makeText(Main_Activity.this, "Daftar", Toast.LENGTH_SHORT).show();
100     startActivity(new Intent(context: Main_Activity.this, SignUpUser_Activity.class).
101         addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK));
102 }
103
104
Main_Activity > loginRequest()

```

Gambar 22. Potongan kode program

File java yang dibuat dan deskripsi fungsionalitasnya yang terdapat pada aplikasi notifikasi jadwal sekolah dapat dilihat:

Tabel 26. Daftar file java aplikasi

No	File Java	Fungsi
1	Onboard_Activity.java	Aktivitas utama yang menangani halaman <i>highlite feature and benefit</i> .
2	Main_Activity.java	Aktivitas utama yang menangani halaman <i>sign in</i> .

No	File Java	Fungsi
3	SignUpUser_Activity.java	Aktivitas utama yang menangani halaman <i>sign up</i> .
4	Home_Activity.java	Aktivitas utama yang menangani halaman jadwal.
5	DetailSchedule_Activity.java	Aktivitas utama yang menangani halaman detail jadwal.
6	Homework_Activity.java	Aktivitas utama yang menangani halaman tugas.
7	DetailHomework_Activity.java	Aktivitas utama yang menangani halaman detail tugas.
8	AddHomework_Activity.java	Aktivitas utama yang menangani halaman tambah tugas.
9	EditHomework_Activity.java	Aktivitas utama yang menangani halaman edit tugas.
10	Setting_Activity.java	Aktivitas utama yang menangani halaman <i>Set</i> Notifikasi.
11	DayBefore_Activity.java	Aktivitas utama yang menangani halaman set alarm hari sebelum.
12	Profile_Activity.java	Aktivitas utama yang menangani halaman profil siswa.
13	About_Activity.java	Aktivitas utama yang menangani halaman tentang aplikasi.
14	AlertReceiverDayBefore2.java	Aktivitas utama yang menangani notifikasi hari sebelum.
15	AlertReceiverMinuteBefore2.java	Aktivitas utama yang menangani notifikasi menit sebelum.

3) Implementasi desain basis data

Implementasi basis data pada aplikasi notifikasi jadwal sekolah berbasis Android terdapat pada server. Pendefinisian tabel dan relasi yang digunakan dalam aplikasi

dilakukan melalui sebuah kelas *java* yang bernama *baseApiServices*. Pendefinisian tabel dan relasi antar tabel dapat dilihat pada Gambar 32.

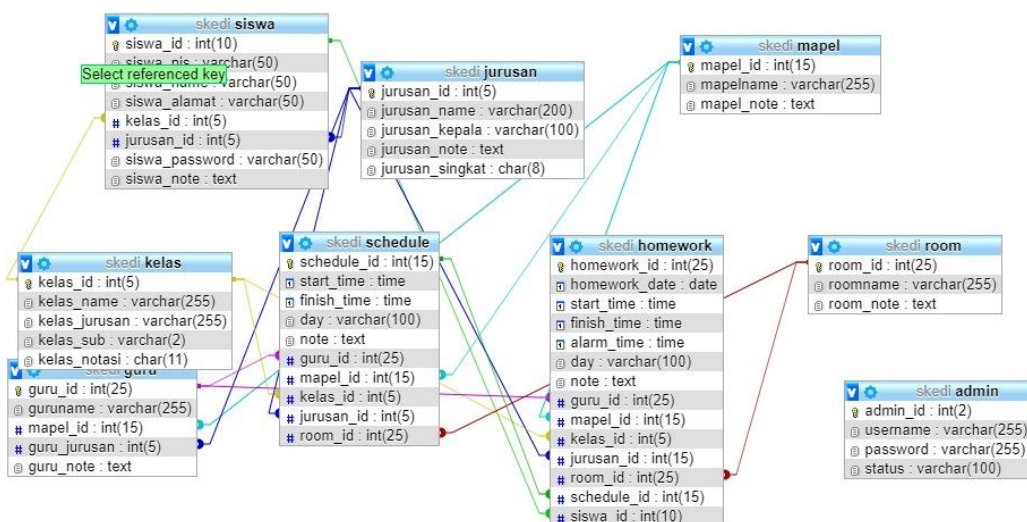
```

1 package com.starcode.skedi.apiHolder;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 public interface baseApiService {
20     @FormUrlEncoded
21     @POST("Users/LoginPost")
22     Call<LoginUserResponse> loginRequest(@Field("NIS") String nik,
23                                         @Field("Password") String password);
24
25     @POST("Users/GetProfile")
26     Call<DataProfileResponse> getAllProfile(@Header("Content-Type") String contentType,
27                                           @Header("Authorization") String authToken);
28
29     @FormUrlEncoded
30     @POST("Users/EditProfile")
31     Call<EditUserProfile> editUserProfile(@Field("siswa_id") String siswa_id,
32                                         @Field("siswa_nis") String siswa_nis,
33                                         @Field("siswa_name") String siswa_name,
34                                         @Field("siswa_alamat") String siswa_alamat,
35                                         @Field("kelas_id") String kelas_id,
36                                         @Field("jurusan_id") String jurusan_id,
37                                         @Field("siswa_password") String siswa_password,
38                                         @Field("siswa_note") String siswa_note
39                                         );
40
41
42
43
44
45
46
47
48
49
50

```

Gambar 23. Implementasi basis data aplikasi

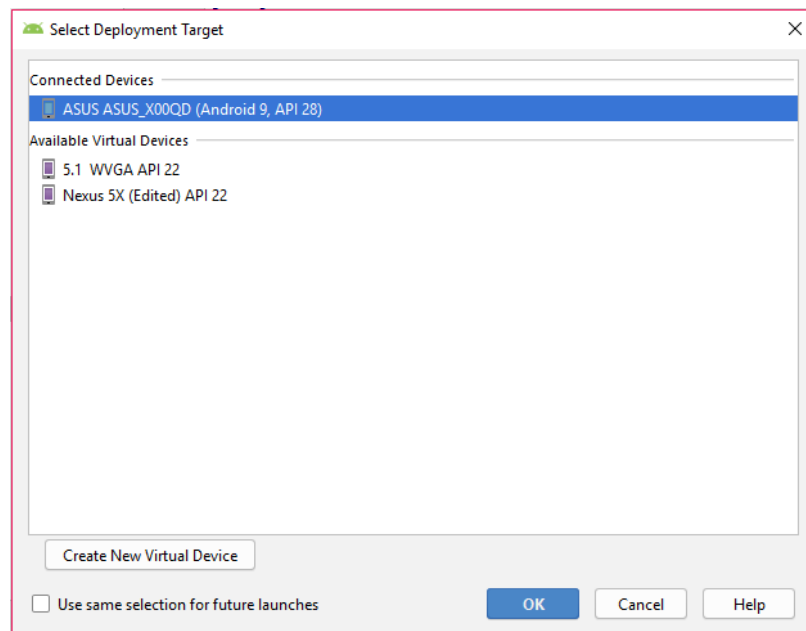
Sedangkan basis data di dalam *server* menggunakan MySQL. Implementasi basis data pada *server* dapat dilihat pada Gambar 33.



Gambar 24. Implementasi basis data server

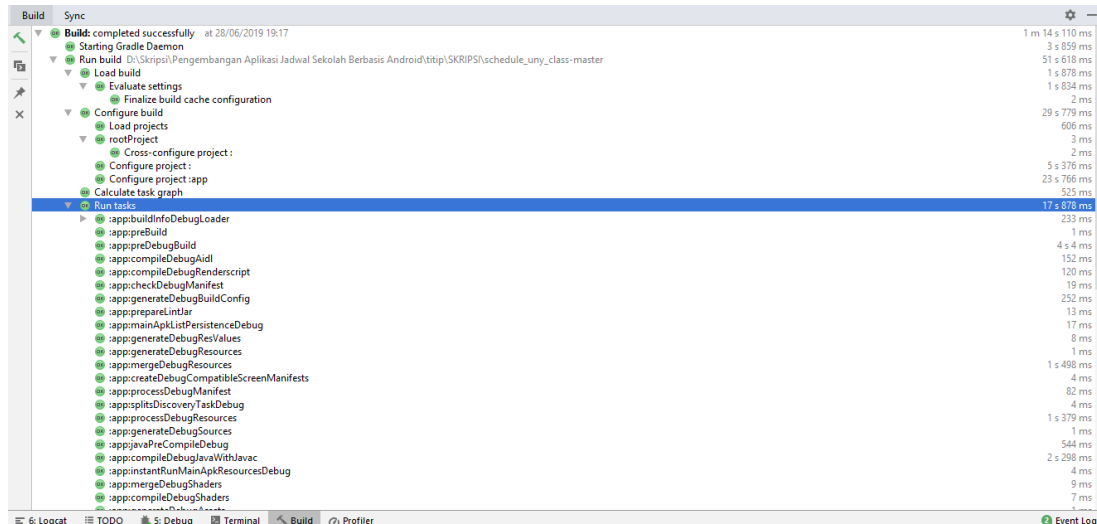
4) *Debugging* aplikasi

Proses *debugging* aplikasi pada penelitian ini menggunakan *device* Android secara langsung dengan kabel data untuk mengkomunikasikan datanya. Hasil dari proses *debugging* tertera pada Gambar 34.



Gambar 25. Memilih deployment target perangkat Android

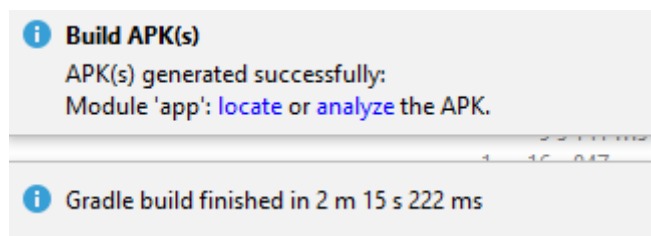
Proses *debugging* terus dilakukan hingga aplikasi muncul di dalam perangkat Android yang dipilih. Kemudian dilakukan pengecekan fungsi yang terdapat pada aplikasi. Monitoring performa aplikasi pada perangkat yang digunakan untuk proses *debugging* dapat dilihat pada Gambar 35.



Gambar 26. Monitoring performa aplikasi pada perangkat yang digunakan proses debugging

5) *Build release* Aplikasi

Build release dilakukan sebelum tahap pengujian agar aplikasi dapat dijalankan pada perangkat lain. Ini merupakan tahap akhir untuk mengubah sekumpulan kode menjadi file berbentuk APK. Gambar 36 menunjukkan *build* APK telah selesai.



Gambar 27. *Build* APK sukses

b. Pengujian perangkat lunak

Pengujian perangkat lunak dilakukan dengan beberapa karakteristik pada standar ISO/IEC 25010. Adapun terdapat 4 karakteristik yang diuji yaitu aspek *functional suitability*, *compatibility*, *usability*, dan *performance efficiency*. Berikut ini adalah hasil pengujian yang telah dilakukan:

1) Hasil pengujian *Functional Suitability*

Pengujian *Functional Suitability* dilakukan oleh tiga orang ahli dari profesi yang memahami tentang proses pengembangan perangkat lunak. Tabel 33 adalah ketiga nama orang ahli tersebut.

Tabel 27. Penguji *Functional Suitability*

No	Nama	Profesi	Instansi
1	Avidah Amalia Zahro, S. Pd.	Guru	SMA Negeri 1 Mertoyudan
2	Arya Wicaksana, S.T.	Web Developers	PT. Pojok Celebes Mandiri
3	Antonius Heru Setiawan, S.T.	Ahli IT	PT. Yoofix

Pengujian dilakukan untuk mengecek fitur-fitur yang terdapat dalam aplikasi apakah dapat berjalan dengan baik atau tidak. Hasil pengujian *functional suitability* yang dilakukan oleh tiga orang ahli terdapat pada Tabel 34.

Tabel 28. Hasil pengujian *Functional Suitability*

No	Fungsi	Skor yang diperoleh Responden			Jumlah	Skor Maksimal
		1	2	3		
1	<i>Highlight feature and benefit</i>	1	1	1	3	3
2	Masuk	1	1	1	3	3
3	Daftar	1	1	1	3	3
4	Lihat Jadwal	1	1	1	3	3
5	Detail Jadwal	1	1	1	3	3
6	Lihat Tugas	1	1	1	3	3
7	Tambah Tugas	1	1	1	3	3
8	Ubah Tugas	1	1	1	3	3
9	Hapus Tugas	1	1	1	3	3
10	Akun	1	1	1	3	3

No	Fungsi	Skor yang diperoleh Responden			Jumlah	Skor Maksimal
		1	2	3		
11	Ubah Password	1	1	1	3	3
12	Tentang	1	1	1	3	3
	Total	12	12	12	36	36

Tabel hasil pengujian *functional suitability* di atas dilakukan oleh tiga orang ahli. Berdasarkan hasil pengujian *Functional Suitability* tersebut maka dapat diperoleh presentase kelayakan sebagai berikut:

$$\text{Presentase Kelayakan(\%)} = \frac{\text{jumlah skor yang didapatkan}}{\text{jumlah skor yang diharapkan}} \times 100\%$$

$$= \frac{36}{36} \times 100\%$$

$$= 100\%$$

Hasil perhitungan presentase kelayakan pengujian *functional suitability* adalah 100%, sehingga dapat disimpulkan bahwa seluruh fitur yang terdapat dalam aplikasi 100% dapat berjalan dengan baik. Berdasarkan hasil perhitungan presentase kelayakan tersebut, kualitas aplikasi dari sisi *functional suitability* memiliki nilai “**Sangat Layak**”.

2) Hasil pengujian *usability*

Pengujian *usability* dilakukan kepada 31 responden yaitu siswa-siswi kelas XI Sistem Informatika dan Jaringan di SMK Negeri 2 Depok. Siswa mencoba langsung aplikasi notifikasi jadwal sekolah pada masing-masing perangkat *smartphone* Android yang mereka miliki, setelahnya para responden mengisi kuisioner yang sudah

dibagikan. Sampel hasil pengujian usability aplikasi notifikasi jadwal sekolah oleh siswa terdapat pada Lampiran 11.

Adapun hasil pengujian usability dengan total skor paling tinggi adalah butir pernyataan nomor 14 dengan pernyataan “Tidak kesulitan menggunakan sistem ini” dengan total skor adalah 145 dari total skor yang diharapkan adalah 160. Hasil pengujian dengan total skor pada tiap-tiap butir pernyataan terdapat pada Lampiran 17. Sedangkan ringkasan hasil pengujian aspek *usability* dapat dilihat pada Tabel 35.

Tabel 29. Ringkasan hasil pengujian aspek *usability*

No	Jawaban	Jumlah	Skor	Jumlah Skor
1.	Sangat Setuju (5)	286	5	1430
2.	Setuju (4)	546	4	2184
3.	Ragu-ragu (3)	120	3	360
4.	Kurang Setuju (2)	7	2	14
5.	Tidak Setuju (1)	1	1	1
Skor Total				3989
Skor Maksimal				4800

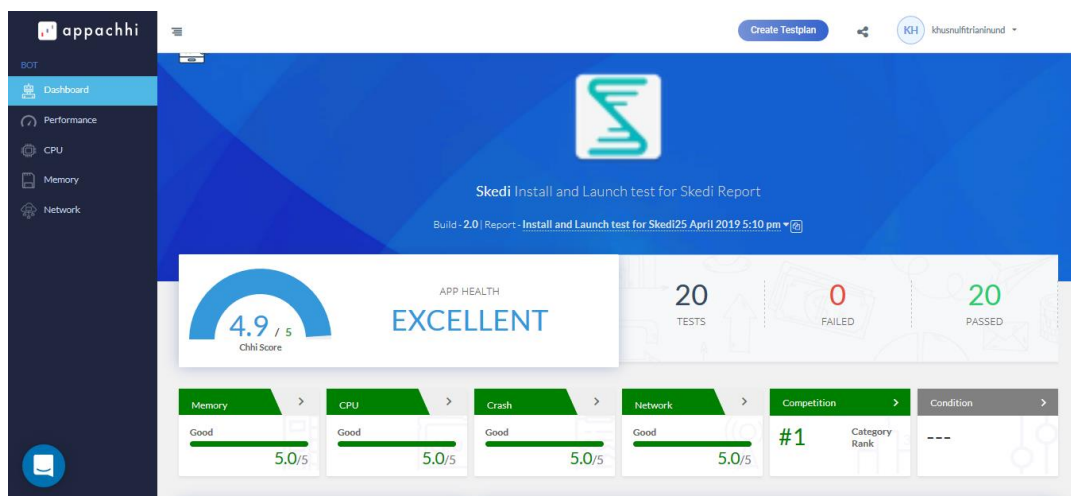
Berdasarkan ringkasan hasil pengujian usability pada Tabel 35, dapat diperoleh presentase kelayakan sebagai berikut:

$$\begin{aligned}
 P_{\text{skor}} &= \frac{\text{skor total}}{i \times r \times 5} \times 100\% \\
 &= \frac{3989}{30 \times 32 \times 5} \times 100\% \\
 &= \frac{3989}{4800} \times 100\% \\
 &= 83,10 \%
 \end{aligned}$$

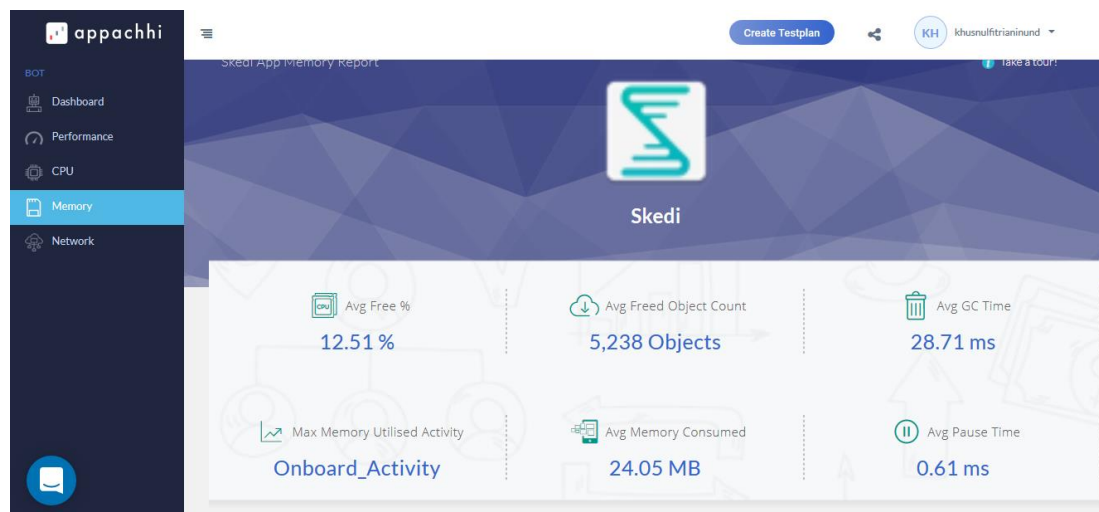
Presentase kelayakan dari pengujian *usability* 83,10 %, sehingga dapat disimpulkan bahwa aplikasi notifikasi jadwal sekolah memiliki nilai “**Sangat Layak**” pada aspek *usability*.

3) Hasil pengujian *performance efficiency*

Pengujian *performance efficiency* menggunakan tools pengujian dari appachhi. Pengujian ini menggunakan 5 perangkat virtual yang disediakan oleh appachhi. Hasil dari pengujian *performance efficiency* terdapat pada Gambar 37 dan Gambar 38.



Gambar 28. Hasil pengujian *performance efficiency*



Gambar 29. Hasil pengujian *performance efficiency* memory

a) Hasil pengujian *Time Behaviour*

Time behaviour dihitung dengan membagi 1 per *thread* tiap detiknya. Berdasarkan hasil pengujian dari tools Appachhi.com aplikasi notifikasi jadwal sekolah berbasis Android dapat dihitung *time behaviour* dari aplikasi pada tabel berikut.

Tabel 30. Pengujian *time behaviour*

No	Nama Perangkat	<i>Thread</i> (/s)
1	LENOVO-Lenovo A7010a48	30
2	Micromax-Micromax AQ4502	31
3	Motorola-MotoE2(4G-LTE)	30
4	PANASONIC-P55 Novo 4G	31
Rata-rata		30,5

$$\begin{aligned}
 \textit{Time behaviour} &= \frac{1}{\textit{Thread per Second}} \\
 &= \frac{1}{30,5} \\
 &= 0,032 \textit{ second/thread}
 \end{aligned}$$

b) Hasil pengujian *Resource Utilization* pada CPU

Berdasarkan hasil pengujian dari tools Appachhi aplikasi notifikasi jadwal sekolah berbasis Android memiliki rata- rata penggunaan CPU sebesar 12,51%.

c) Hasil pengujian *Resource Utilization* pada Memory

Berdasarkan hasil pengujian dari tools Appachhi aplikasi jadwal sekolah berbasis Android memiliki rata-rata penggunaan memori sebesar 24,5 MB.

Berdasarkan seluruh pengujian *performance efficiency* yang telah dilakukan, aplikasi notifikasi jadwal sekolah berbasis Android dapat berjalan dengan baik tanpa mengalami *memory leak* yang mengakibatkan *error* atau *force close*. Oleh karena itu,

berdasarkan hasil pengujian pada aspek *performance efficiency* aplikasi notifikasi jadwal sekolah berbasis Android dapat dikatakan “**Layak**”.

4) Hasil pengujian *compatibility*

Pengujian *compatibility* dilakukan dengan dengan cara menginstal aplikasi Notifikasi Jadwal Sekolah pada perangkat *smartphone* Android kemudian menjalankannya. Pengujian dilakukan dengan menggunakan berbagai macam test:

a) *Coexistence*

Pengujian dilakukan menjalankan aplikasi Notifikasi Jadwal Sekolah bersama dengan aplikasi lain sehingga dapat diketahui aplikasi Notifikasi Jadwal Sekolah dapat berjalan bersama dengan aplikasi lain dengan baik atau tidak.

Tabel 31. Hasil Uji Coexistence

No	Pernyataan	Berhasil	Gagal
1	Aplikasi google play store dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
2	Aplikasi kamera dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
3	Aplikasi whatsapp dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
4	Aplikasi gmail dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
5	Aplikasi google maps dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
6	Aplikasi spotify dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
7	Aplikasi youtube dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
8	Aplikasi gojek dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0

No	Pernyataan	Berhasil	Gagal
9	Aplikasi shopee dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
10	Aplikasi instagram dan notifikasi jadwal sekolah dapat berjalan bersamaan dengan baik	1	0
Total		10	0

Dari Tabel 37 uji *coexistence* menunjukkan hasil pengujian aplikasi yang dijalankan bersamaan dengan 10 aplikasi lain dalam suatu perangkat. Berdasarkan uji *coexistence* tersebut maka dapat diperoleh presentase kelayakan sebagai berikut:

$$\begin{aligned}
 \text{Presentase Kelayakan (\%)} &= \frac{\text{skor yang didapatkan}}{\text{skor maksimal}} \times 100\% \\
 &= \frac{10}{10} \times 100\% \\
 &= 100\%
 \end{aligned}$$

Hasil perhitungan presentase kelayakan adalah 100%, oleh karena itu, aplikasi Notifikasi Jadwal Sekolah memenuhi standar *coexistence* dengan “**Sangat Layak**”.

b) Pengujian pada berbagai merek perangkat *mobile* dan versi Android

Uji pada berbagai sistem operasi, tipe perangkat dan berbagai tipe layar dilakukan secara manual yaitu menginstal aplikasi Notifikasi Jadwal Sekolah ke dalam *smartphone* Android. *Smartphone* Android yang digunakan berjumlah 10 dengan spesifikasi minimal untuk melakukan penginstalan aplikasi Notifikasi Jadwal Sekolah. Daftar *smartphone* Android yang digunakan dalam pengujian *compatibility* dapat dilihat dalam Tabel 38.

Tabel 32. Uji *compatibility* dengan *smartphone* yang berbeda

No	Perangkat	Versi OS	CPU	RAM	Ukuran Layar
1	Redmi 5+	8.1	1.8 Ghz	3 GB	5.7”
2	Redmi 4A	7.0	1.4 Ghz	2 GB	5”

No	Perangkat	Versi OS	CPU	RAM	Ukuran Layar
3	Redmi 6	8.1	2.0 Ghz	4 GB	5.45"
4	Oppo A57	6.0	1.4 Ghz	3 GB	5.2"
5	Oppo F7	8.1	2.0 Ghz	4 GB	6.23"
6	Asus Zenfone max pro (m2)	9.0	1.8 Ghz	3 GB	6.3"
7	Asus Zenfone 5 ZE620KL	9.0	2.0 Ghz	4 GB	6.2"
8	Redmi 4X	9.0	1.4 Ghz	2 GB	5"
9	Samsung Galaxy S10+	9.0	2.8 Ghz	8 GB	6.1"
10	Redmi Note 6 Pro	9.0	1.8 Ghz	4 GB	6.26"

Adapun setelah dilakukan pengujian, didapat hasil uji pada 10 perangkat *smartphone* Android. Ringkasan hasil pengujian aspek compatibility dapat dilihat pada Tabel 39.

Tabel 33. Hasil Uji dengan 10 *smartphone* Android

No	Pengujian	Jumlah skor	Berhasil	Gagal
1	Instalasi aplikasi pada perangkat	10	10	0
2	Menjalankan aplikasi pada perangkat	10	10	0
Total		20	20	0

Setelah mendapatkan hasil pengujian, dilakukan perhitungan presentase kelayakan sebagai berikut:

$$\begin{aligned}
 \text{Presentase Kelayakan (\%)} &= \frac{\text{skor yang didapatkan}}{\text{skor maksimal}} \times 100\% \\
 &= \frac{20}{20} \times 100\% \\
 &= 100\%
 \end{aligned}$$

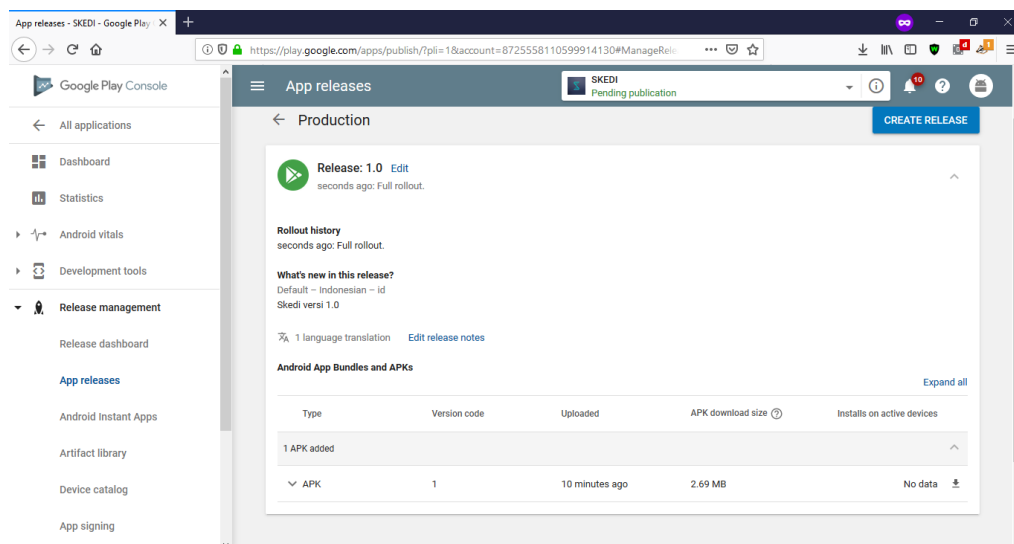
Hasil perhitungan presentase kelayakan dari pengujian aspek *compatibility* uji berbagai *smartphone* Android adalah 100%, sehingga dapat disimpulkan aplikasi Notifikasi Jadwal Sekolah memiliki nilai “**Sangat Layak**”.

Berdasarkan hasil seluruh uji *compatibility*, dapat disimpulkan bahwa aplikasi notifikasi jadwal sekolah memiliki *compatibility* yang “**Sangat Layak**” karena:

- a) Dari sisi *coexistence*, aplikasi Notifikasi Jadwal Sekolah dapat dijalankan bersamaan dengan aplikasi yang lain tanpa ada yang dirugikan.
- b) Dari sisi sistem operasi dan tipe perangkat aplikasi Notifikasi Jadwal Sekolah dapat dijalankan tanpa ada permasalahan

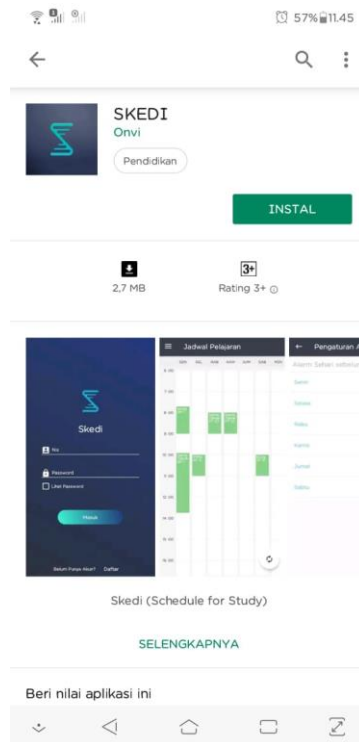
5. Penyerahan perangkat kepada pengguna (*Deployment*)

Proses *deployment* dilakukan menggunakan layanan *Google Play Store*. Aplikasi Notifikasi Jadwal Sekolah di *publish* di *Google Play Store* dalam kategori pendidikan. Publikasi aplikasi di *Google Play Store* bertujuan untuk membuat pengguna lebih mudah dalam mendapatkan aplikasi tersebut. Gambar 39 menunjukkan proses *publish* aplikasi ke *Google Play Store*.



Gambar 30. Proses publish aplikasi ke *Google Play Store*

Setelah diunggah, *play store* membutuhkan waktu untuk menampilkan Aplikasi Notifikasi Jadwal Sekolah. Gambar 40 merupakan tampilan halaman Google Play Store yang menampilkan aplikasi Notifikasi Jadwal Sekolah.



Gambar 31. Aplikasi Notifikasi Jadwal Sekolah pada Google Play Store

B. Pembahasan

Aplikasi Notifikasi Jadwal Sekolah adalah aplikasi berbasis Android yang memberikan notifikasi dan informasi jadwal mata pelajaran secara detail. Aplikasi ini telah melalui tahap pengembangan yaitu *communication*, *planning*, *modeling*, *construction*, dan *deployment*.

Tahap komunikasi dilakukan dengan siswa-siswi SMK Negeri 2 Depok, dimulai dari menganalisis permasalahan yang ada hingga melakukan wawancara, yang kemudian menghasilkan spesifikasi produk serta daftar kebutuhan perangkat lunak untuk digunakan pada tahap selanjutnya yaitu tahap desain.

Tahap kedua adalah perencanaan, berupa penyusunan jadwal pengembangan aplikasi. Tahap ketiga adalah pemodelan, desain terdiri dari UI (*User Interface*) dan UX (*User Experience*) dengan menggunakan UML (*Unified Modeling Language*), desain basisdata, dan desain sistem.

Tahap keempat adalah konstruksi, dari desain UI dan UX tersebut dibangun menjadi sebuah aplikasi Android menggunakan Android Studio. Setelah menjadi aplikasi, selanjutnya dilakukan pengujian berdasarkan ISO 25010 dengan teori Ben David mencakup aspek *usability*, *functional suitability*, *performance efficiency*, dan *compatibility*.

Pengujian *usability* dilakukan oleh pengguna aplikasi dengan menggunakan instrumen A.M Lund. Hasil yang diperoleh dari pengujian tersebut yaitu 83,10% sehingga dapat dikatakan “**Sangat Layak**” dalam memenuhi aspek *usability*. Pengujian *functional suitability* dilakukan oleh 3 ahli di berbagai bidang IT. Hasil pengujian menunjukkan bahwa seluruh fungsi berhasil sehingga dapat dikatakan “**Sangat Layak**” dalam memenuhi aspek *functional suitability*. Pengujian pada aspek *performance efficiency* mendapatkan hasil yaitu rata-rata penggunaan CPU sebesar 12,51%, sedangkan rata-rata penggunaan memory 24,5 MB, dan penggunaan time behaviour rata-rata adalah sebesar 0,032 detik. Sehingga dapat dikategorikan sebagai “**Layak**” dalam memenuhi aspek *performance efficiency*. Pengujian pada aspek *compatibility* diuji melalui uji *coexistence* dan uji pada perangkat Android. Pada uji *coexistence*, aplikasi Notifikasi Jadwal Sekolah dijalankan bersamaan dengan 10 aplikasi lain, hasilnya semua dapat berjalan dengan baik. Kemudian aplikasi juga di uji pada 10 perangkat Android, dan hasilnya dapat berjalan di semua perangkat tanpa

mengalami error atau *force close*. Sehingga dapat disimpulkan aplikasi ini “**Sangat Layak**” dalam memenuhi aspek *Compatibility*.

C. Keterbatasan Penelitian

Pengembangan aplikasi notifikasi jadwal sekolah di SMK N 2 Depok memiliki keterbatasan, diantaranya belum dapat menambah foto profil guru, belum dapat digunakan oleh guru hanya dapat digunakan untuk siswa, serta belum terdapat *pop-up notification* di aplikasi notifikasi jadwal sekolah. Penelitian ini juga masih terbatas dengan hanya dapat digunakan pada perangkat dengan sistem operasi Android.