

APLIKASI VIRTUAL ASSISTANT BERBASIS ANDROID *ANDROID BASED VIRTUAL ASSISTANT*

Satya Anggrahita¹, Siska Komala Sari, S.T., M.T.¹, Wahyu Hidayat, S.T., M.T.²

¹Jurusan Manajemen Informatika, Fakultas Ilmu Terapan, Telkom University, Bandung.
satya.anggrahita24@gmail.com, siska@tass.telkomuniversity.ac.id,
wahyuhidayat@tass.telkomuniversity.ac.id.

ABSTRAK

Aplikasi Virtual Assistant Berbasis Android adalah aplikasi yang digunakan untuk membantu mahasiswa mengetahui kegiatan yang akan dilakukan. Kebanyakan saat ini mahasiswa memiliki kesulitan dalam mengetahui kegiatan kesehariannya. Bukan hanya itu, ada beberapa mahasiswa yang telah mengetahui kegiatannya akan tetapi sering lupa karena tidak ada yang mengingatkan. Aplikasi berbasis android ini dibangun menggunakan metode *prototype*. Pembuatan aplikasi ini juga menggunakan Bahasa pemogramJava yang diterapkan pada perangkat lunak android studio sebagai media untuk membangun aplikasi. Sedangkan untuk perancangan programnya menggunakan *storyboard* dan alur navigasi. Dengan dibangunnya aplikasi ini, diharapkan dapat membantu mahasiswa untuk mengetahui kegiatan secara akurat, juga dilengkapi dengaadanya fitur pemberitahuan (*notifikasi*) dapat mempermudah dalam pemberitahuan informasi tentang kegiatan yang akan dilakukan.

Kata Kunci: Kegiatan, Android Studio, Prototype

ABSTRACT

The Android-Based Application for Virtual Assistant is an application used to help student know the activities. Mostly student still have difficulty in knowing the daily activities. Not only that, some student who have known the activities but also forgotten because is no one reminding. This android-based built using the protorype method. The making of the application also uses the Java programming language are applied on android studio software as a media to build the application. For the design of program using storyboards and flow in navigation. With the building of this application, is expected to help student to known the activities accurate, also equipped with notification feature for knowing the activities.

Keywords: Activities, Android Studio, Prototype

Universitas
Telkom

1. Pendahuluan

1.1 Latar Belakang

Dalam era mobilitas tinggi seperti sekarang, banyak sekali orang yang melakukan banyak aktivitas dalam satu hari. Semisal bekerja, kemudian berbelanja, mencuci mobil dan lain sebagainya. Ketepatan waktu dan keruntutan apa yang akan dilakukan tentu menjadi kunci tidak beraturannya *daily schedule activity*. Aktifitas sehari-hari itu tidak hanya satu atau dua, tetapi banyak, sehingga supaya tepat waktu, bila harus mengingat satu persatu tentu akan merepotkan.

Masyarakat saat ini banyak sekali melakukan suatu aktifitas, dimana aktifitas tersebut tidaklah sedikit, apalagi bagi para mahasiswa, mahasiswa memiliki banyak kegiatan di kampus. Dari sekian banyak kegiatan, mahasiswa sering lupa akan kegiatan kesehariannya. Akibatnya, banyak kegiatan mahasiswa yang tidak dilakukan.

Selain itu, mahasiswa juga harus mengerjakan *deadline* tugas, mengajar sebagai asisten praktikum, mengikuti kegiatan daerah, mengikuti organisasi dan sebagainya. Dengan aktifitas sebanyak itu, tidak sedikit mahasiswa yang lupa akan aktifitasnya. Sehingga mereka harus mengingat satu persatu. Akibatnya, mereka lebih banyak memikirkan daripada melaksanakan kegiatan tersebut.

Hasil survey membuktikan bahwa 12 mahasiswa dari 16 yang berkuliah saat ini memerlukan notifikasi untuk mempermudah para mahasiswa dalam mengingat kegiatan-kegiatan mahasiswa tersebut. Kebanyakan mahasiswa merasa terbebani ketika mengatur jadwal sehari-hari.

Dengan seiringnya perkembangan zaman, dan semakin mudahnya teknologi yang dapat dilakukan oleh semua mahasiswa, sehingga pada saat ini kebanyakan mahasiswa cenderung tidak lepas dari yang namanya *smartphone*. Berdasarkan masalah tersebut, maka akan dibuatlah sebuah “Aplikasi Virtual Assistant Berbasis Android” yang dapat memberikan kemudahan bagi para mahasiswa untuk mengetahui apakah aktifitas yang akan dilakukan oleh para mahasiswa, dan juga dilengkapi dengan fitur pemberitahuan

(*notifikasi*) untuk mempermudah dalam pemberitahuan informasi aktifitas mahasiswa. Sehingga nantinya diharapkan dapat membantu para mahasiswa menjalankan aktifitas kesehariannya.

1.2 Rumusan Masalah

Berdasarkan paparan pada sub bab Latar Belakang, dalam Proyek Akhir ini masalah yang diangkat adalah bagaimana mengurangi resiko tidak ingat supaya mahasiswa tidak lupa terhadap jadwal atau kegiatan yang sudah ditetapkan.

1.3 Tujuan

Untuk memberikan solusi bagi masalah-masalah yang telah dipaparkan pada sub bab Rumusan Masalah, maka dapat dirumuskan bahwa tujuan dari Proyek Akhir ini adalah membangun sebuah Aplikasi Virtual Assistant yang memiliki fitur “create, notifikasi” untuk meringankan beban mencatat jadwal sehari-hari.

1.4 Batasan Masalah

Agar pembahasan dalam Proyek Akhir ini tidak melebar, maka ditetapkan batasan-batasan sebagai berikut:

1. Aplikasi ini akan dibangun di aplikasi Android Studio
2. Fitur catatan hanya dalam bentuk teks
3. Aplikasi ini tidak menyediakan fitur share

1.5 Definisi Operasional

Aplikasi Virtual Assistant adalah sebuah aplikasi berbasis android yang digunakan untuk mengingatkan pengguna dalam mengatur kegiatan sehari-hari. Aplikasi ini mencakup pembuatan jadwal sehari-hari.

Dengan adanya Aplikasi Virtual Assistant, maka pengguna dapat mengetahui jadwal-jadwal yang akan dilakukan. Hal ini dapat mengurangi resiko lupa akan jadwal yang akan dikerjakan, janji yang sudah dibuat. Dengan Aplikasi ini pengguna dapat mengoptimalkan waktu yang ada.

2. Tinjauan Pustaka

2.1 Virtual Assistant

Sebuah perangkat lunak yang dapat melakukan tugas atau layanan untuk seorang individu. Beberapa asisten virtual mampu menafsirkan ucapan manusia dan merespons melalui suara yang disintesis. Pengguna dapat mengajukan pertanyaan asisten mereka, mengontrol perangkat otomatisasi rumah dan pemutaran media melalui suara, dan mengelola tugas-tugas dasar lainnya seperti email, daftar yang harus dilakukan, dan kalender dengan perintah verbal [1].

2.2 Android

Android merupakan sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya Google Inc. membeli Android Inc. yang merupakan pendatang baru untuk membuat perangkat lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [2].

2.3 Unified Modelling Language

UML (Unified Modelling Language) merupakan standar bahasa yang banyak di dunia industri bertujuan mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek[3]. Pada perkembangan teknik pemrograman berorientasi objek, munculah standarisasi suatu bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu Unified Modelling Language (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi

mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan.

2.4 Usecase Diagram

Fungsionalitas *Use Case* direpresentasikan dengan aliran – aliran peristiwa. *Use Case* menggambarkan bagaimana realisasi dari setiap *use case* yang ada pada model[4]. *Use case* diagram merupakan suatu aktivitas yang menggambarkan urutan interaksi antar satu atau lebih aktor dan sistem. *Use case* yang akan dirancang yaitu *use case* diagram untuk pengaksesan melalui perangkat android. *Use case* atau diagram *use case* merupakan pemodelan untuk melakukan sistem informasi yang akan dibuat. *Use case* didefinisikan sebagai interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat[5].

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “Apa” yang diperbuat sistem, dan bukan “Bagaimana”. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya Login ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

2.5 Storyboard

Storyboard mempunyai peranan yang sangat penting dalam pengembangan multimedia. *Storyboard* digunakan sebagai alat bantu pada tahapan perancangan multimedia. *Storyboard* merupakan pengorganisasian grafik, contohnya adalah sederetan ilustrasi atau gambar yang ditampilkan berurutan untuk keperluan visualisasi awal dari suatu file, animasi, atau urutan media interaktif, termasuk interaktivitas di web. Winanto dalam membagi *storyboard* menjadi dua yakni *storyboard* ringkas dan *storyboard* lengkap[8].

2.6 Blackbox Testing

Blackbox testing adalah pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus pada *output* yang dihasilkan yang merespon *input* yang telah dipilih dengan kondisi eksekusi tertentu. Pengujian dilakukan untuk mengevaluasi pemenuhan sistem dengan kebutuhan fungsionalitas tertentu. Pengujian tidak pernah memeriksa kode program dan informasi lebih lanjut mengenai program serta spesifikasinya[6].

Blackbox Testing adalah strategi pengujian dimana hanya memperlihatkan atau memfokuskan kepada faktor fungsionalitas dan spesifikasi perangkat lunak. Uji coba *input* yang akan melatih seluruh syarat fungsionalitas suatu program.

2.7 SQLite

SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi SQL *interface* dan penggunaan memory sangat sedikit dengan kecepatan yang sangat cepat. SQLite di android termasuk dalam android runtime, sehingga setiap versi dari android dapat membuat database dengan SQLite.

Dalam sistem android memiliki beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut:

- a. *Shared Preferences* yaitu menyimpan data beberapa nilai (*value*) dalam bentuk groups key yang dikenal dengan *preferences*.
- b. *Files* yaitu penyimpanan data dalam *file*, dapat berupa menulis ke *file* atau membaca dari file.
- c. *SQLite Databases*, yaitu menyimpan data dalam bentuk *Databases*.
- d. *Content Provider*, yaitu menyimpan data dalam bentuk *content provider Service*.

Tidak ada database yang otomatis disediakan oleh android, jika kita menggunakan SQLite, kita harus mengcreate database sendiri, mendefinisikan tabelnya,

index serta datanya. Untuk membuat dan membuka database yang paling baik adalah menggunakan libraries `import android.database.sqlite.SQLiteOpenHelper` yang menyediakan tiga metode yaitu:

a. *Constructor*, menyediakan representasi versi dari *database* dan skema database yang kita gunakan.

b. *onCreate()*, menyediakan `SQLiteDatabase` object yang kita gunakan dalam definisi tabel dan inialisasi data.

c. *onUpgrade()*, menyediakan fasilitas konversi database dari database versi lama ke database versi yang baru atau sebaliknya [7].

3. Analisis

Pada sub bab ini akan dilakukan perbandingan antara aplikasi yang akan dibangun dan aplikasi lainnya yang sejenis, maka dari itu akan di bagi dalam 5 bagian yaitu Analisis perbandingan aplikasi sejenis, analisis system berjalan, analisis system usulan, kebutuhan perangkat lunak dan perangkat keras serta kebutuhan fungsionalitas.

3.1.1 Analisis Keetuhan Produk

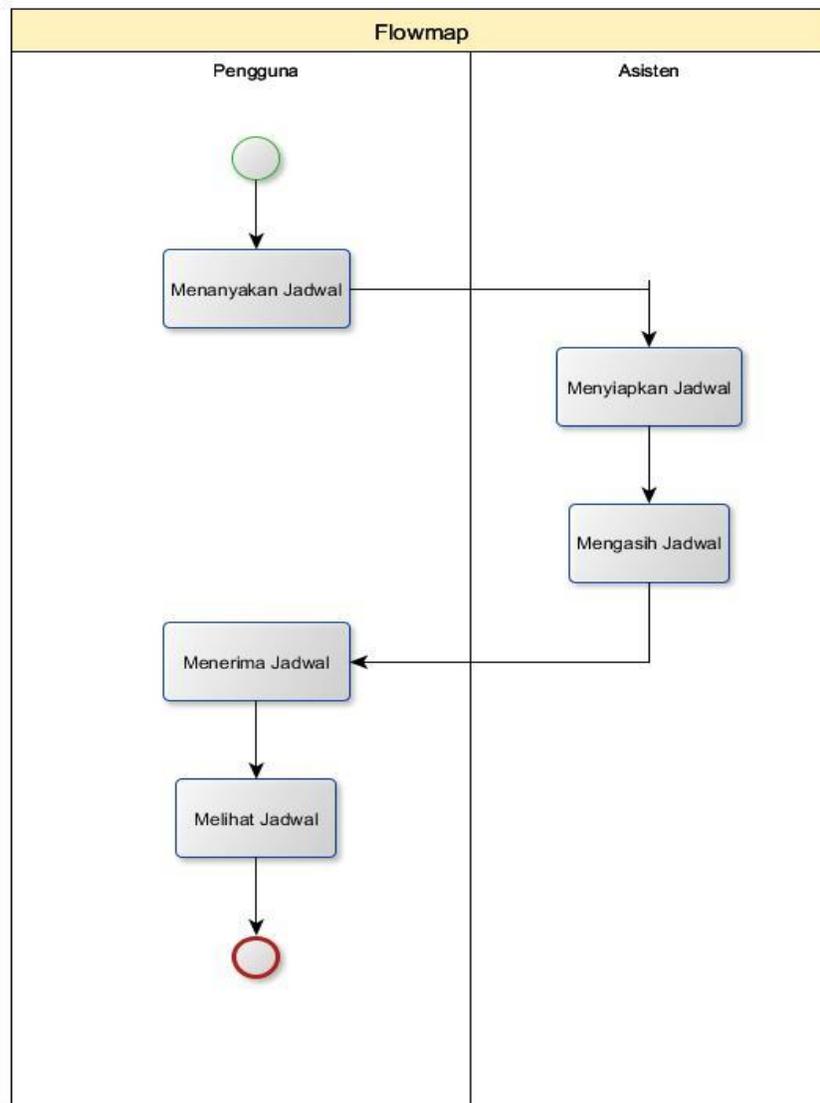
Berikut merupakan fitur-fitur yang ada pada Aplikasi Virtual Assistant

1. Menyajikan fitur membuat, dan melihat untuk mencatat kegiatan yang akan dilakukan
2. Menyajikan notifikasi kegiatan

3.1.2 Analisis Sistem Berjalan

Pada proses berjalan saat ini kegiatan melihat jadwal masih berjalan secara manual. Pertama pengguna atau masyarakat terlebih dahulu menanyakan jadwal kepada asisten yang akan dikerjakan. Lalu, asisten menyiapkan jadwal yang diminta oleh pengguna. Kemudian, pengguna menerima jadwal lalu pengguna baru bisa melihat jadwal tersebut.

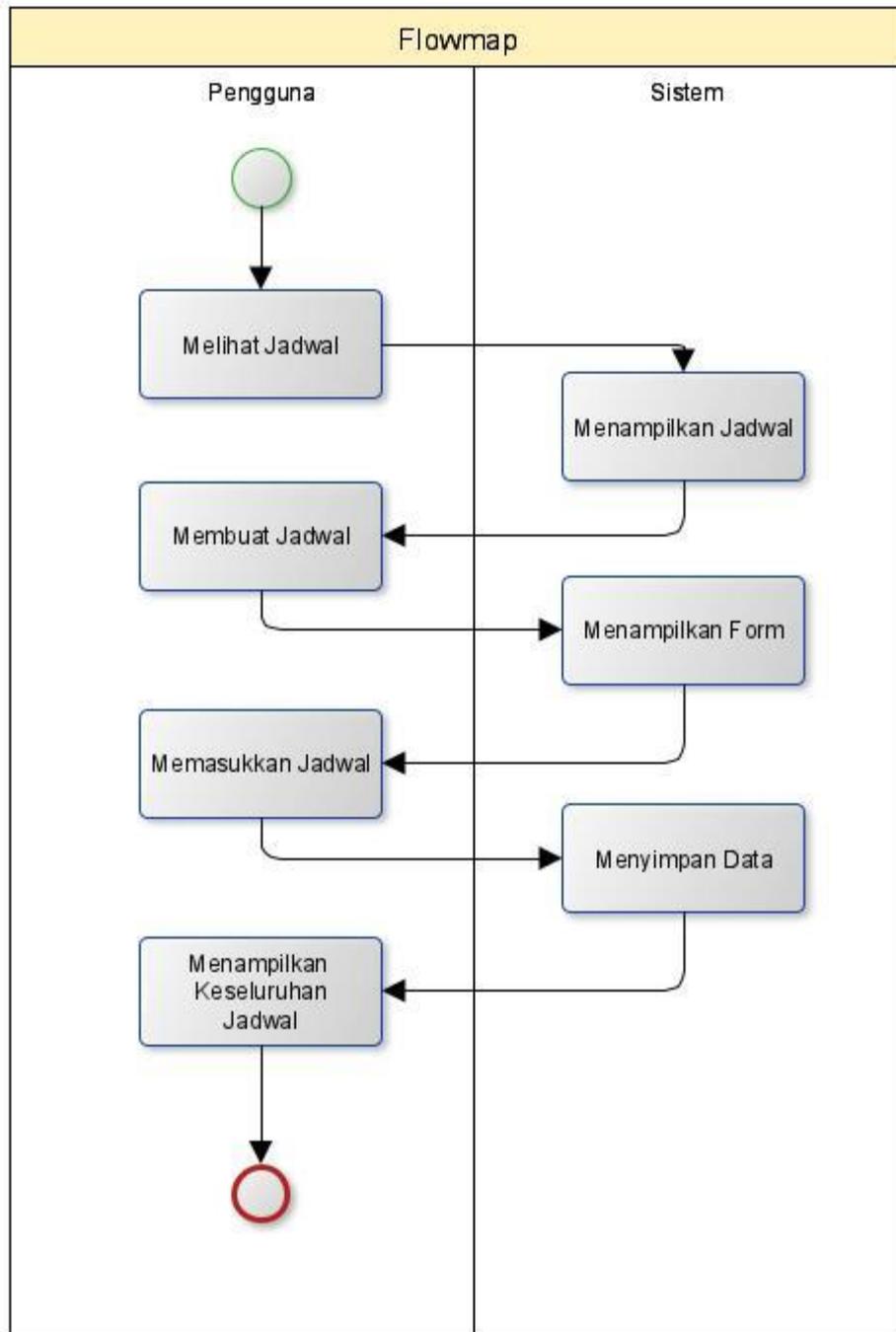
Dengan sistem seperti ini, terdapat kelemahan yang terjadi yaitu pengguna harus menunggu asisten terlebih dahulu, kemudian, pengguna baru bisa melihat jadwal tersebut.



3.1.3 Analisis Sistem Usulan

Aplikasi Virtual Assistant ini memiliki satu user, adapun usernya adalah masyarakat atau mahasiswa.





3.1.4 Analisis Perbandingan Sistem Sejenis

Dalam pembuatan aplikasi ini penulis menganalisis perbandingan aplikasi yang sejenis untuk mendapatkan kekurangan atau menemukan fitur yang belum ada pada aplikasi lain sehingga dapat diimplementasikan pada aplikasi yang akan di buat ini. Berikut tabel perbandingan aplikasi sejenis tersebut.

No	Faktor Pemanding	Aplikasi A	Aplikasi B	Aplikasi yang akan dibuat
1	Nama Aplikasi	ToDo List Task Manager[9]	To-do List[10]	Virtual Assistant
2	Fitur Membuat Kegiatan	Ada	Ada	Ada
3	Fitur Notifikasi	Ada	Ada	Ada
4	Ringkas	Tidak	Tidak	Ada

Dari hasil table di atas terlihat bahwa beberapa fitur dari aplikasi ToDo List Manager dan To-Do List memiliki fitur membuat kegiatan, fitur notifikasi, tetapi aplikasi tersebut tidak ringkas untuk digunakan.

3.1.5 Analisis Kebutuhan Perangkat Lunak

Adapun kebutuhan perangkat lunak untuk membuat aplikasi ini, yaitu:

No	Jenis Software	Spesifikasi Minimum Perangkat Lunak	Spesifikasai Perangkat Lunak yang Digunakan
1	<i>Operating System</i>	Windows 7	Windows 10
2	<i>Software Aplikasi</i>	Android Studio	Android Studio
3	<i>Database</i>	SQLite	SQLite
No	Jenis Software	Spesifikasi Minimum Perangkat Lunak	Spesifikasai Perangkat Lunak yang Digunakan

4	<i>Editing</i>	Adobe Photoshop CS5	Adobe Photoshop CC 18
5	<i>Dokumentasi</i>	Microsoft Word 2006	Microsoft Word 2016

3.1.6 Analisis Kebutuhan Perangkat Keras

Adapun kebutuhan perangkat keras untuk membuat aplikasi ini, yaitu:

No	Jenish Hardware	Spesifikasi Minimum Perangkat Keras	Spesifikasai Perangkat Keras yang Digunakan
1	<i>Processor</i>	I3	I5
2	RAM	4 GB	8GB
3	<i>Hard disk</i>	100 GB	150 GB
4	<i>Operating System Android</i>	Kitkat	Marshmallow

3.1.7 Kebutuhan Fungsionalitas

Analisis sistem merupakan sebuah Teknik pemecah masalah yang menguraikan sebuah sistem menjadi bagian-bagian komponen dengan tujuan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka. Analisis sistem dilakukan sebelum tahap desain sistem dan merupakan tahap-tahap kritis dan sangat penting, karena kesalahan di tahap selanjutnya.

3.1.7.1 Diagram Usecase

Pada bagian ini terdapat diagram usecase perancangan aplikasi yang ingin dibangun. Dimana fungsionalitasnya ada tiga yaitu melihat jadwal, membuat jadwal, menghapus jadwal.

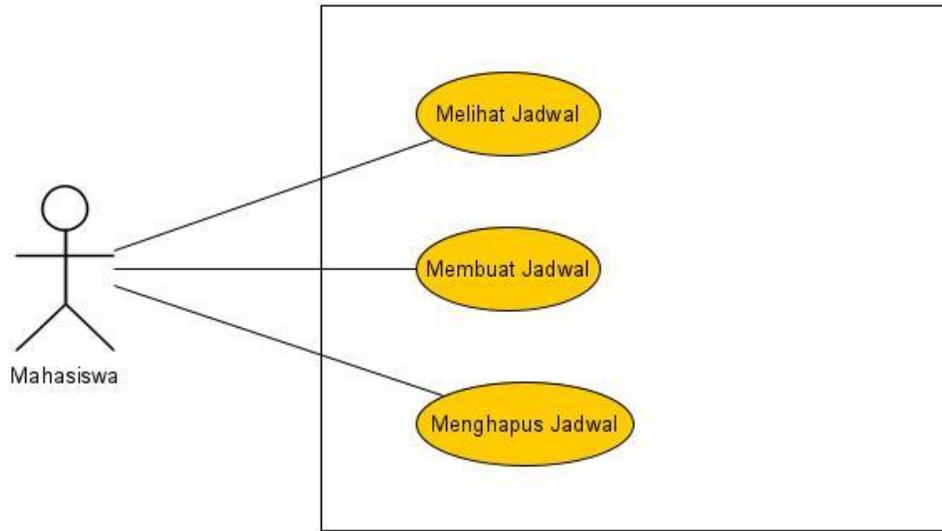


Diagram diatas memperlihatkan bagaimana interaksi antar komponen di dalam aplikasi yang dapat dilalui oleh pengguna yang menggunakan aplikasi virtual assistant. *Usecase diagram* juga menunjukkan menu yang dapat dipilih pengguna Ketika menggunakan aplikasi ini.

3.1.7.2 Skenario Usecase

Berikut adalah scenario *use case* yang akan menjelaskan *use case diagram* diatas:

1. Skenario Usecase Melihat Jadwal

Nama Usecase : Melihat Jadwal

Aktor : Mahasiswa

Deskripsi : mahasiswa dapat melihat catatan kegiatan yang sudah diinputkan pengguna ke dalam aplikasi

Pre-Condition : mahasiswa sudah mengisi form jadwal kegiatan

Post-Condition : mahasiswa berhasil melihat jadwal kegiatan

Mahasiswa	Reaksi Sistem
1. Masuk ke aplikasi	
	2. Menampilkan halaman awal yang berisi jadwal kegiatan mahasiswa
3. Melihat jadwal kegiatan	

2. Skenario Usecase Membuat Jadwal

Nama Usecase : Membuat Jadwal

Aktor : Mahasiswa

Deskripsi : Mahasiswa dapat membuat jadwal kegiatan

Pre-Condition : Mahasiswa sudah mengisi form jadwal kegiatan

Post-Condition : Mahasiswa berhasil membuat jadwal kegiatan

Mahasiswa	Reaksi Sistem
1. Masuk ke aplikasi	
	2. Menampilkan halaman awal
3. Klik tombol tambah	
	4. Menampilkan form tambah jadwal
5. Mengisi form	
	6. Menampilkan jadwal di halaman awal

3. Scenario Usecase Menghapus Jadwal

Nama Usecase : Menghapus Jadwal

Aktor : Mahasiswa

Deskripsi : Mahasiswa dapat menghapus jadwal

Pre-Condition : Mahasiswa sudah mengisi form jadwal kegiatan

Post-Condition : Mahasiswa berhasil menghapus jadwal kegiatan

Mahasiswa	Reaksi Sistem
1. Masuk ke aplikasi	
	2. Menampilkan halaman awal
3. Klik tombol hapus	
	4. Menghapus jadwal kegiatan
	5. Menampilkan jadwal kegiatan di halaman awal

3.1.7.3 Deskripsi Usecase

Adapun deskripsi actor yang akan berkaitan dengan aplikasi ini adalah

Tabel 1 Deskripsi Use Case

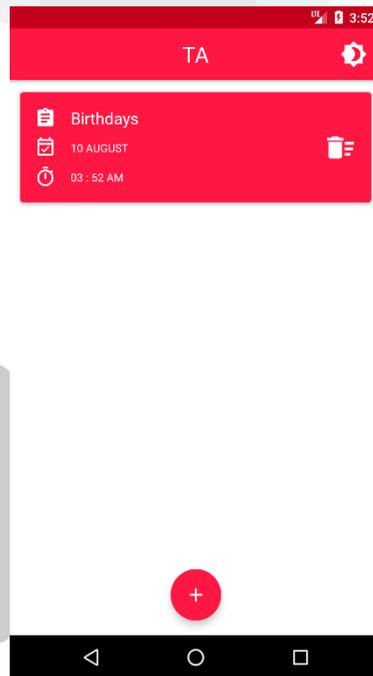
Use Case	Deskripsi
Lihat Jadwal	Aktivitas yang digunakan oleh mahasiswa untuk melihat jadwal keseluruhan
Membuat Jadwal	Aktivitas yang dilakukan oleh mahasiswa untuk membuat jadwal
Menghapus jadwal	Aktivitas yang dilakukan mahasiswa untuk menghapus jadwal yang sudah selesai

3.2 Perancangan Antarmuka

Perancangan antarmuka dilakukan dengan menggunakan *balsamiq wireframe* untuk menggambarkan aplikasi yang akan dibangun. Berikut adalah tampilan antarmuka setiap fungsionalitas:

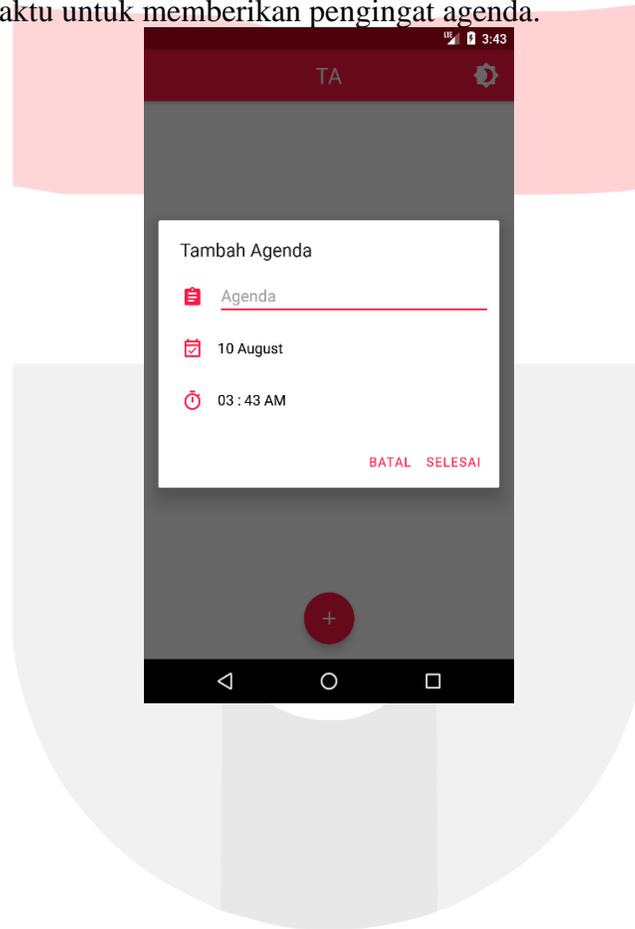
3.2.1 Halaman Awal

Halaman awal berisi informasi jadwal atau kegiatan pengguna sehari-hari.



3.2.2 Halaman Tambah Jadwal Kegiatan

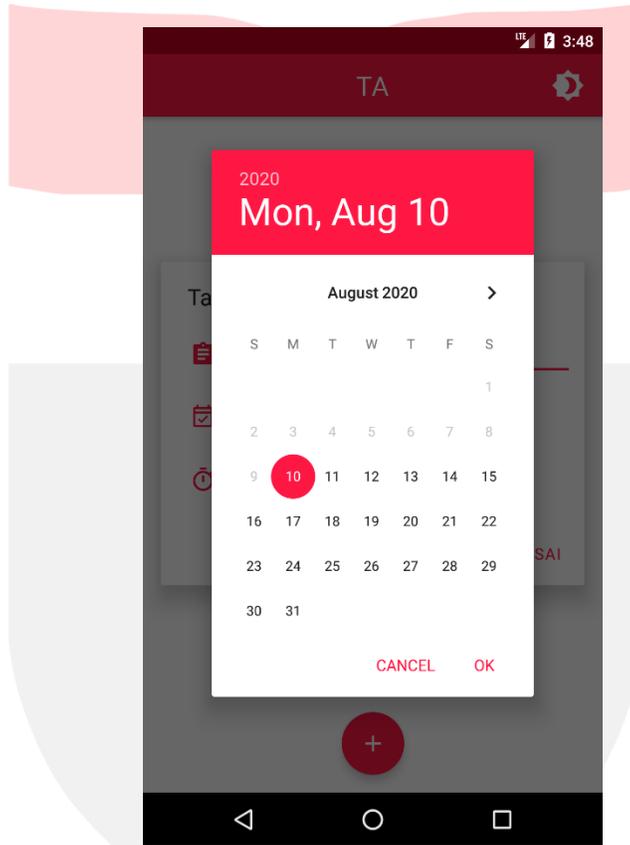
Halaman tambah jadwal kegiatan berisi baris nama agenda, tanggal agenda untuk menetapkan agenda, waktu untuk memberikan pengingat agenda.



Universitas
Telkom

3.2.3 Halaman Pemilihan Waktu

Halaman ini berisikan penetapan waktu untuk menentukan notifikasi muncul



4. Implementasi dan Pengujian

Pada tahap ini akan dijelaskan mengenai pengujian yang telah dilakukan terhadap system. Pengujian dilakukan oleh target *user* yang akan menggunakan system dengan menguji semua fungsionalitas yang ada pada aplikasi ini. Hasil pengujiannya adalah sebagai berikut.

Perangkat Lunak	Aplikasi Virtual Assistant
Deskripsi	Aplikasi yang digunakan oleh <i>user</i> yang memiliki segala hak akses atas aplikasi tersebut
Function	
<i>Function 1</i>	<i>Input data kegiatan</i>

No	Function	Case	Test Case Description (Event)	Test Data (Input)	Expected Result
1	Function 1	1	Melakukan Input data kegiatan <i>User</i> memilih Input data kegiatan		
		1.1	Entry data kegiatan <i>User</i> memasukkan data berupa: Nama agenda, tanggal agenda, Jam agenda, kemudian klik selesai	Birthdays, 10 agustus, 03.52	Input data kegiatan berhasil, data tersimpan.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan tujuan dan hasil pengujian, maka dapat ditarik kesimpulan bahwa beberapa mahasiswa merasa terbebani dalam mengatur jadwal kegiatan sehari-hari. Dengan terbebannya mengatur jadwal keseharian, maka dibuatlah aplikasi pendukung dengan fitur membuat jadwal dan notifikasi jadwal kegiatan sehari-hari

5.2 Saran

Berikut adalah saran yang dapat dipertimbangkan untuk pengembang aplikasi di masa akan datang adalah menambahkan fitur kategori berdasarkan kegiatan.

Referensi

- [1] Hoy, Matthew B. (2018). "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants". Medical Reference Services Quarterly
- [2] Sfaat, Nazruddin, Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Bandung: Informatika Bandung, 2015.
- [3] David M. Kroenke, *Database Processing Jilid 1 edisi 9*, halaman 60. Erlangga.
- [4] Ariani, Rosa, Rekayasa Perangkat Lunak, Bandung: Modula, 2011.
- [5] Ivon, Horton, Ivor Horton's Beginning Java 2, JDK 5 Edition, Indianapolis: Wiley Publishing, 2005.
- [6] Putra, Hendra, Jaminan Mutu Sistem Informasi, Bandung: PT. Remaja Rosdakarya, 2009.
- [7] Allen, Grant; Owens, Mike (November 5, 2010).
- [8] Binanto, Iwan, Multimedia Digital Dasar Teori dan Pengembangan, Yogyakarta: C.V ANDI OFFSET, 2010.
- [9] <https://play.google.com/store/apps/details?id=com.mikesandroidworkshop.android.tasklist>
- [10] <https://play.google.com/store/apps/details?id=io.github.neomsoft.todo>

Universitas
Telkom