

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Tinjauan pustaka merupakan hasil dari beberapa penelitian sebelumnya yang sudah pernah dilakukan dan digunakan sebagai acuan penelitian selanjutnya. Berikut ini adalah beberapa penelitian yang berkaitan dengan penelitian ini:

**Tabel 2.1 Tinjauan Pustaka**

Tahun	Judul	Penulis	Permasalahan	Hasil
2020	Perancangan Aplikasi Reminder Dan Monitoring Jadwal Servis Sepeda Motor Berbasis Android	Holis Muchlis Sugianto dan Asti Herliana	Sering terjadinya kecelakaan akibat terjadinya malfungsi kendaraan karena kelalaian pemilik kendaraan mengatur jadwal <i>service</i> kendaraan	Sistem ini memiliki beberapa fitur seperti menjadwalkan agenda, pengingat, dan perhitungan batas maksimal <i>service</i> kendaraan.
2020	Perancangan Aplikasi Notifikasi Jadwal Kuliah Berbasis Android (Studi Kasus : Prodi Teknik Informatika-Uniks)	Rila Efra Ningsih	Memperbaiki kualitas penyebaran informasi dan mempermudah penyebaran informasi mengenai jadwal perkuliahan melalui perangkat android	Adanya aplikasi perangkat android berbasis mobile yang memiliki notifikasi jadwal perkuliahan untuk mahasiswa.

**Tabel 2.2 Tinjauan Pustaka (Lanjutan)**

2021	Pembuatan Aplikasi Reminder Jadwal Perkuliahan di Jurusan Teknik Eletronika Berbasis Android	Genta Primadana Edde, Khairi Budayawan	Mempermudah mengingatkan mahasiswa dalam mengatur agenda atau jadwal kuliah dan kegiatan lain secara efisien dan efektif antara akademik dan kegiatan berorganisasi.	Aplikasi yang dibuat dapat membantu mahasiswa dengan fitur-fitur yang berfungsi untuk menunjang kegiatan mahasiswa.
2021	Aplikasi Pengingat Kegiatan Akademik Berbasis Android	Azlin, Abdul Rasyid Sabirin, I Kadek Deni Mardikayasa	Untuk membantu pendistribusian perubahan informasi akademik secara <i>realtime</i> dan mengingatkan dosen maupun mahasiswa untuk melaksanakan kegiatan akademik sesuai jadwal.	Menghasilkan aplikasi yang dapat membantu mahasiswa dan dosen mengatur perkuliahan secara <i>realtime</i> dalam melaksanakan kegiatan akademik.
2022	Aplikasi Reminder Jadwal Kegiatan Berbasis Mobile	Veri Ihadi, Danil Ardiansyah, dan Muthmainnah	Dibutuhkan suatu aplikasi untuk membantu mengingat melakukan kegiatan sesuai jadwal yang membantu untuk mengingatkan jadwal sehingga terhindar dari kelalaian.	Menghasilkan suatu aplikasi pengingat kegiatan yang memiliki fitur notifikasi reminder sebagai pengingat jadwal.

Dari penjelasan beberapa literatur di atas, dapat disimpulkan bahwa aplikasi penjadwalan kuliah bagi mahasiswa sudah memiliki beberapa fitur-fitur yang dapat digunakan mahasiswa untuk meningkatkan produktivitas dalam bidang akademik maupun kegiatan organisasi. Dari beberapa literatur tersebut fitur yang digunakan oleh aplikasi yang ada yaitu pengingat berupa *reminder* yang merupakan fitur utama dalam aplikasi penjadwalan. Berdasarkan tinjauan tersebut, pada penelitian ini penulis akan meningkatkan penggunaan fitur *reminder* dan menambahkan beberapa fitur pendukung lain seperti akses penyimpanan file, berupa dokumen dan foto yang dapat dipergunakan mahasiswa untuk menunjang kegiatan perkuliahan.

## **2.2 Pengingat**

Menurut Suwarti dan Catriwati, pengingat merupakan suatu petunjuk atau pengaturan dengan spesifik waktu yang berfungsi untuk mengalokasikan kegiatan, informasi, yang ada dengan tujuan untuk diselesaikan sesuai dengan rencana yang ada (Suwarti & Catriwati, 2022).

Sedangkan menurut Dwi, pengingat diartikan suatu fungsi pengambilan keputusan berdasarkan dengan teori dan prinsip yang membuktikan suatu kegiatan yang ada dalam manajemen waktu (Dwi et al., 2021; Yos & Sudarso, 2021).

Berdasarkan uraian beberapa peneliti diatas, penulis dapat menyimpulkan bahwa pengingat adalah suatu cara atau metode pengumpulan informasi dengan tujuan untuk membantu seseorang mengingat rencana yang sudah dibuat.

## **2.3 Reminder**

Menurut Desta, *reminder* adalah sebuah pesan yang membantu seseorang untuk mengingat sesuatu. *Reminder* dapat dikatakan bermanfaat jika menyajikan informasi pada waktu yang tepat sesuai dengan yang ditentukan. *Reminder* digunakan untuk manajemen waktu yang ditampilkan sebagai alarm berupa peringatan dan pemberitahuan (Yos & Sudarso, 2021).

## **2.4 Android**

*Android* merupakan sebuah sistem operasi berbasis *linux* pada perangkat *smartphone*. Sistem operasi dikatakan sebagai jembatan yang menghubungkan

antara *device* dan penggunanya, sehingga pengguna dapat menggunakan aplikasi tersebut pada *device* yang dipakai sebagai platform terbuka.

Menurut Holis, *android* adalah perangkat yang memiliki kelebihan seperti bersifat *open source* alias gratis, cepat dan *responsive*, *user friendly* dan memiliki produk yang beragam (Holis & Herliana, 2020).

Sedangkan menurut Desta pada jurnalnya, aplikasi *Android* saat ini memiliki kemajuan pesat yang memiliki beberapa kategori yang menjadi ruang lingkup dari pengembangan aplikasi *Android* diantaranya yaitu dalam bidang pendidikan, komunikasi, informasi, permainan, jejaring sosial dan Kesehatan. Sehingga aplikasi *Android* memiliki banyak manfaat membantu pengguna dalam kehidupan sehari-hari (Yos & Sudarso, 2021).

## **2.5 Perangkat Lunak Pengembangan Sistem**

Dalam mengembangkan suatu aplikasi berbasis *android*, dibutuhkan beberapa perangkat lunak dalam pembangunan sebuah aplikasi sebagai berikut :

### **2.5.1 *Android Studio***

*Android studio* merupakan *Integrated Development Environment (IDE)* atau dalam pengertian lain yaitu sebuah *device* untuk mengembangkan sistem operasi *Google Android*. Aplikasi ini dapat digunakan di berbagai sistem operasi yang ada, yaitu *Windows*, *Linux*, dan *MacOS*. Pada aplikasi ini banyak fitur canggih yang dapat digunakan untuk meningkatkan produktifitas dalam proses pengembangan suatu aplikasi, sehingga *android studio* dilirik oleh banyak *developer* dalam pembuatan aplikasi (Niesviantika & Marcos, n.d.)

Sebagai Pengembangan dari Eclipse *Android Studio* mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan ADT, *Android Studio* Menggunakan *gradle* sebagai build environment. Fitur-fitur lainnya antara lain; menggunakan *Gradle-based build system* yang fleksibel, dapat mem-build multiple APK, *template support* untuk *Google Service* dan berbagai macam tipe perangkat dan layout editor yang lebih bagus (Cki, 2022).

### 2.5.2 Java

*Java* adalah bahasa pemrograman yang memiliki banyak kegunaan. Bahasa ini dapat digunakan untuk membuat suatu program, mendukung sumber daya internet atau yang lebih dikenal sebagai *World Wide Web* atau sering disebut dengan *web*.

Sedangkan menurut Desta, *Java* adalah suatu bahasa pemrograman *multi-platform*. Beberapa *IDE* yang *support* ke dalam bahasa *Java*, misalnya *Netbeans*, *Eclipse*, *TextPad*, dan lain-lain (Yos & Sudarso, 2021).

### 2.5.3 Firebase

*Firebase* adalah salah satu layanan yang digunakan untuk pengembang aplikasi dapat melakukan penyimpanan data. Fitur yang terdapat pada *firebase* antara lain yaitu *firebase remote config* dan *firebase realtime database*. Data pada *firebase* disimpan dalam bentuk objek *JSON tree*. Sedangkan untuk pengembangan aplikasi digunakan *firebase storage* yang memungkinkan pengembang mengunggah atau melakukan pengunduhan berkas (Sukmansyah et al., 2021).

Beberapa layanan yang tersedia pada *firebase* antara lain:

#### 1. Analisis *Firebase*

Layanan ini merupakan bagian dari aplikasi berbayar yang memungkinkan pengembang memahami cara mendapatkan data dan wawasan dalam penggunaan aplikasi.

#### 2. *Firebase Cloud Messaging (FCM)*

Layanan ini dikenal sebagai *Google Clouds Messaging (GCM)* yang memberikan pesan pemberitahuan pada aplikasi *Android*, *website* maupun *IOS*.

#### 3. *Firebase Auth*

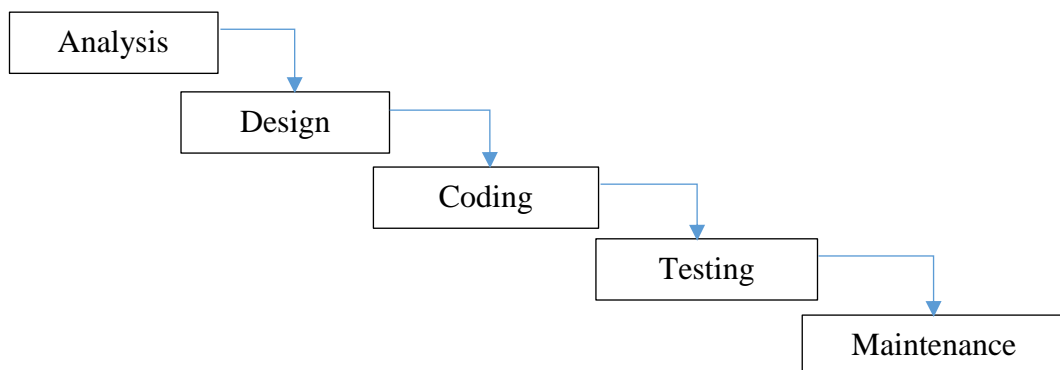
Layanan ini mendukung penyedia *login* pada media social seperti pada *Facebook*, *Google*, dan *Twitter*. Layanan ini akan mengautentikasi pengguna yang *login* dengan menggunakan sisi kode dimana pengembang akan mengaktifkan otentikasi pengguna dengan email dan *password* dan disimpan pada *Firebase*.

#### 4. *Firebase Storage*

Memfasilitasi transfer *file* yang mudah dan aman. Layanan ini dapat digunakan untuk menyimpan gambar, audio, video, maupun konten lain yang dibuat oleh pengguna.

## 2.6 Metode Pengembangan Sistem

Dalam proses mengembangkan sebuah sistem, diperlukan metode yang dapat mempermudah pembuatan sebuah aplikasi android. Pada penelitian ini penulis menggunakan metode *Waterfall*. Dalam jurnal yang ditulis oleh Veri Ilhadi, model *waterfall* dapat dianalogikan mirip seperti air terjun karena masing-masing tahap dikerjakan secara berurutan mulai dari atas hingga ke bawah. Mulai dari analisa kebutuhan, merancang sistem, hingga membangun sistem sesuai kebutuhan (Ilhadi & Ardiansyah, n.d.). Sedangkan menurut Anna dalam jurnal penelitiannya, model *waterfall* dikatakan adalah metode yang paling cocok digunakan untuk pengembangan sistem yang identifikasi kebutuhannya sudah direncanakan dari awal, dan metode ini sesuai untuk tugas akhir atau skripsi yang bertujuan untuk membuat sebuah sistem (Sukmansyah et al., 2021).



**Gambar 2.1 Metode Waterfall**

Dalam metode ini memiliki beberapa tahapan yaitu: *analysis*, *design*, *coding*, *testing*, dan *maintenance*. Pada tahap awal penulis akan melakukan identifikasi masalah yang ada. Kemudian akan dilanjutkan dengan tahapan mendesain sistem sesuai dengan kebutuhan. Kemudian dilakukan pembuatan program dengan desain yang ada. Setelah itu dilakukan tahapan pengujian sistem. Pada tahapan pengujian ini akan dilakukan uji coba keseluruhan fitur sehingga dapat menemukan kesalahan atau kekeliruan *software* yang tidak sinkron dengan kebutuhan analisis pada awal

penelitian (Niesviantika & Marcos, n.d.). Setelah selesai akan dilakukan pemeliharaan untuk memperbaiki kesalahan pada sistem.

## 2.7 Unified Modeling Language





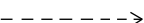
Menurut Prihandoyo, *Unified Modeling Language* (UML) merupakan sebuah model perancangan sistem yang bertujuan memberikan kemudahan pada *developer* sistem untuk merancang sistem (Yos & Sudarso, 2021). UML juga merupakan sebuah standar penulisan atau *blue print* yang memiliki bisnis proses dengan bahasa yang spesifik.

Dalam UML terdapat beberapa diagram yang sering digunakan dalam pengembangan sistem, yaitu:

### 2.7.1 Use Case

*Use case* merupakan gambaran fungsionalitas yang memiliki aktor dan sistem. Pada *use case* terdapat aktor yang merupakan simbol manusia atau sistem yang melakukan pekerjaan. Sekumpulan *use case* dikatakan dapat menggambarkan sebuah sistem mengikuti sebanyaknya ketentuan-ketentuan yang ada pada apa yang dilakukan oleh user pada sistem tersebut. Sehingga *use case* dikatakan memiliki peranan penting dalam pembuatan sebuah aplikasi (Suwarti & Catriwati, 2022).






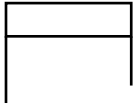
**Tabel 2.3 Simbol dan Keterangan Use Case**

Simbol	Keterangan
Aktor 	Mewakili orang atau sistem saat melakukan komunikasi dengan <i>use case</i> .
<i>Use case</i> 	Interaksi antara sistem dan aktor.
Generalisasi 	Menunjukkan partisipasi aktor dalam <i>use case</i> .
<i>Extend</i> 	Menunjukkan perluasan perilaku dari sumber pada informasi lain dalam sistem.
<i>Include</i> 	Menunjukkan bagian fungsionalitas dari <i>use case</i> lain.

### 2.7.2 Activity Diagram

Merupakan gambaran alur aktivitas yang sedang berjalan di dalam sebuah sistem yang dikembangkan. *Activity diagram* memiliki peranan cukup penting dalam pembuatan sebuah aplikasi. Sebuah aplikasi dapat dikatakan berhasil jika membuat aktivitas yang dijelaskan secara rinci. Dalam pembuatan suatu aplikasi diperlukan penjelasan aktivitas yang ringkas dan digambarkan melalui diagram. *Activity diagram* ini menggambarkan alur aktivitas dan menjelaskan proses masing-masing dari awal hingga akhir dari proses suatu sistem. Simbol dan keterangan mengenai *activity diagram* seperti pada tabel 2.4.

**Tabel 2.4 Simbol dan Keterangan Activity Diagram**

Simbol	Keterangan
Status Awal 	Status awal menggambarkan sebuah diagram aktivitas.
Perancangan 	Asosiasi percabangan dengan kondisi jika ada pilihan aktivitas lebih dari satu.
Status Akhir 	Status akhir menggambarkan sebuah diagram aktivitas yang memiliki akhir.
Action 	Menunjukkan antar kelas yang saling berinteraksi.
Fork Node 	Satu aliran yang berubah menjadi beberapa aliran.
Swimlane 	Memisahkan organisasi bisnis pada saat aktivitas yang terjadi.

### 2.8 Flowchart


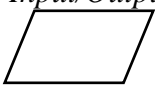
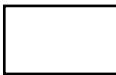
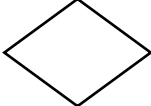

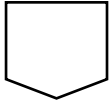
Perancangan sistem memiliki peranan penting dalam pembuatan aplikasi. Salah satunya adalah pembuatan *flowchart* yang dapat menjelaskan secara rinci






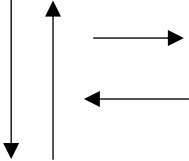
pembuatan aplikasi. *Flowchart* adalah urutan atau alir penyelesaian masalah yang terdiri dari simbol-simbol yang menggambarkan alur program.

Skema pembuatan aplikasi dan jalan agar aplikasi tersebut dapat menemukan penyelesaian adalah dengan menggunakan *flowchart*. Oleh karena itu setiap pembuatan aplikasi diperlukan pembuatan *flowchart* agar aplikasi dapat terbentuk sesuai dengan rancangan awal dan menghasilkan aplikasi yang sesuai dengan permintaan atau kebutuhan *user*. *Flowchart* bertujuan untuk menjelaskan tahapan penyelesaian masalah secara sederhana dan terurai. Simbol dan Keterangan *flowchart* terdapat pada tabel 2.5.

**Tabel 2.5 Simbol dan Keterangan Flowchart**

Simbol	Keterangan
<p>Terminal</p> 	Menggambarkan permulaan atau akhir suatu program.
<p>Input/Output</p> 	Menggambarkan proses <i>input</i> atau <i>output</i> suatu program.
<p>Process</p> 	Menggambarkan tindakan atau proses yang dilakukan.
<p>Decision</p> 	Menggambarkan kondisi tertentu dengan kemungkinan ya atau tidak.
<p>Connector</p> 	Menyatakan sambungan dari satu proses ke proses lainnya dalam satu halaman.
<p>Offline Connector</p> 	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman yang berbeda.

**Tabel 2.6 Simbol dan Keterangan Flowchart (Lanjutan)**

<p><i>Predefined Process</i></p> 	<p>Menyatakan penyediaan tempat penyimpanan suatu pengolahan.</p>
<p><i>Punched Card</i></p> 	<p>Menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.</p>
<p><i>Document</i></p> 	<p>Mencetak keluaran dalam bentuk dokumen melalui printer.</p>
<p><i>Flow</i></p> 	<p>Menyatakan jalannya alur sebuah proses.</p>

## 2.9 Metode Pengujian Sistem

Pengujian pada perangkat lunak sangat penting karena berhubungan dengan kualitas sistem yang dihasilkan. Pada penelitian ini digunakan pengujian menggunakan pengujian *functional* dan *compatibility*. Pengujian merupakan pengujian yang berfokus pada spesifikasi fungsional pada perangkat lunak.