

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1. Penelitian Terdahulu**

Beberapa penelitian terdahulu yang relevan dengan penelitian ini diantaranya adalah :

Esther Irawati Setiawan [6] melakukan penelitian dengan judul “Aplikasi *Mobile* Untuk Memantau *Body Mass Index* Dengan Metodologi Scrum”. Metode yang digunakan dalam pengembangan aplikasi adalah Scrum. Metode ini memiliki lima tahapan di antaranya adalah *Sprint Planning*, *Sprint Daily Meeting*, *Daily Scrum*, *Sprint Review*, dan *Retrospective*. Masalah yang terdapat dalam penelitian ini adalah untuk mencegah masyarakat terkena obesitas dikarenakan masa pandemi yang menyebabkan kurang bergerak dan meningkatkan berat tubuh. Maka dari itu penelitian ini merancang sebuah aplikasi berbasis *mobile* untuk dapat mengukur BMI sehingga dapat mengetahui apakah badan yang ia miliki termasuk dalam kategori ideal atau belum. Aplikasi ini mengimplementasikan dua metode perhitungan BMI, yaitu dengan input manual dan menggunakan kamera *smartphone*.

Sri Puji Utami [7] melakukan penelitian dengan judul “Rancang Bangun Aplikasi Edukasi Tuberkolosis Menggunakan Metode Scrum”. Scrum dipilih sebagai metode dalam melakukan pengembangan aplikasi. Metode ini dapat memiliki kerangka tim yang berisi *Product Owner*, *Scrum Master* dan *Team Scrum*, serta memiliki enam tahapan di antaranya adalah *User Stories*, *Product Backlog*, *Sprint Planning*, *Scrum Execution*, *Sprint Review* dan *Sprint Retrospective*. Adanya pembatasan sosial dimasa pandemi menjadi masalah di dalam penelitian untuk melakukan kegiatan penyuluhan kepada pasien maupun masyarakat mengenai TBC. Penelitian ini membuat sebuah aplikasi berbasis Android sebagai media untuk membantu para kader TBC memberikan penyuluhan. Aplikasi ini dibangun dengan fitur dan gambar yang sederhana agar mudah diakses dan dicerna.

Izmy Alwiah Musdar [8] melakukan penelitian dengan judul “Rancang Bangun Sistem Informasi Pariwisata Sulawesi Selatan Berbasis Android Dengan Menggunakan Metode *Prototyping*”. Metode yang digunakan adalah *prototyping* dalam mengembangkan perangkat lunak. Metode ini memiliki lima tahapan di antaranya adalah Penentuan Kebutuhan Sistem, Perancangan Sistem, Pembuatan *Prototype* Awal, Evaluasi *Prototype*, dan Modifikasi *Prototype*. Masalah yang terdapat pada penelitian ini adalah upaya dinas pariwisata Sulawesi Selatan dalam memperkenalkan objek-objek wisata. Penelitian ini mengembangkan sebuah sistem informasi berbasis *mobile* untuk membantu menyajikan informasi lengkap mengenai objek wisata yang ada di Sulawesi Selatan dengan lebih informatif seperti mengetahui *event* yang sedang terselenggara di wilayah tersebut

Helga Pramudita [9] melakukan penelitian dengan judul “Sosialisasi Perawatan Gigi Dan Mulut Pada Anak Berbasis Android”. *Multimedia Development Life Cycle (MDLC)* dipilih sebagai metode dalam melakukan pengembangan aplikasi. Metode ini memiliki enam tahapan di antaranya adalah *Concept, Design, Material Collecting, Assembly, Testing, dan Distribution*. Minimnya pengetahuan dan kesadaran masyarakat mengenai penyakit gigi dan mulut menjadi masalah di dalam penelitian ini. Penelitian ini membuat sebuah aplikasi kesehatan gigi dan mulut untuk anak-anak usia 6 – 10 tahun. Informasi di dalam aplikasi dikemas menarik dengan gambar dan video agar dapat memudahkan anak-anak lebih memahami mengenai kesehatan gigi dan mulut.

Fuad Husain Akbar [10] melakukan penelitian dengan judul “Rancang Bangun Aplikasi *Teledentistry* Berbasis *Appstore* Dan *Playstore* Pada Manajemen Praktik Gigi Selama Pandemi Covid-19 Dan Adapatasi Kebiasaan Baru”. Metode yang digunakan dalam analisis aplikasi ini adalah *Research and Development (R&D)*. Metode ini memiliki tiga tahapan di antaranya adalah Studi Pendahuluan, Pengembangan Model, dan Uji Model. Masalah yang terdapat dalam penelitian ini adalah selama pandemi Covid-19, banyak prosedur gigi rutin di seluruh dunia telah ditanggguhkan dan hanya prosedur dan operasi gigi darurat saja yang dilakukan. Penelitian ini melihat peluang untuk membuat sebuah aplikasi *Teledentistry*

berbasis Android yang meminimalkan kontak langsung antara pasien dan dokter. Aplikasi ini memudahkan proses pelayanan, membantu meningkatkan efektivitas waktu dalam proses pelayanan dan pencatatan klinik gigi secara akurat dan cepat, dan dapat membantu untuk meminimalisir terjadinya kesalahan pencatatan data pasien.

Nugrahani Putri [11] melakukan penelitian dengan judul “Implementasi Metode *Prototyping* Pada Perancangan Aplikasi *Electronic Ticket (E-Ticket)* Berbasis Android”. *Prototyping* dipilih sebagai metode dalam melakukan pengembangan aplikasi. Metode ini memiliki tiga tahapan di antaranya adalah Tahap Desain, Pembuatan Program, dan Tahap Evaluasi. Penyajian informasi ketersediaan tiket bus menjadi masalah dalam penelitian ini, dimana calon penumpang harus datang ke loket agen untuk mengetahui ketersediaan tiket dan melakukan pemesanan. Penelitian ini membuat sebuah aplikasi pemesanan tiket bus untuk memudahkan calon penumpang dalam melakukan pencarian bus dan pemesanan tiket dimana saja dan kapan saja tanpa terikat dengan waktu dan tempat.

Dewi Ratna [12] melakukan penelitian dengan judul “*A Mobile App (Smart Dental Alarm) On Improving Tooth Bruising Skills Among Early Childhood*”. Metode yang digunakan dalam pengembangan aplikasi ini adalah *Research and Development (R&D)*. Metode ini memiliki lima tahapan di antaranya adalah *Information Collection, Model Design, Expert Validation and Revision, Model Testing*, dan *Finished Model*. Masalah yang terdapat dalam penelitian ini adalah terkait dengan penyakit karies gigi pada anak-anak berumur 5 – 9 tahun yang mempunyai presentasi hingga 90.1%. Situasi ini disebabkan karena lemahnya keterampilan anak-anak dalam menggosok gigi. Penelitian ini membuat sebuah aplikasi “*Smart Dental Alarm*” sebagai media untuk meningkatkan kualitas kegiatan menggosok gigi untuk anak-anak. Aplikasi ini juga mengimplementasikan kamera untuk mengirimkan hasil gambar gigi untuk disimpan di server dan dapat diakses kembali.

Aditya Nurrochman [13] melakukan penelitian dengan judul “*Application ‘Senyum Gigiku’ Android Based Media Promotion As Prevention Caries Dental*

*Knowledge And Attitudes Toward Increasing The Mother In District Banyudono PKK*". *Waterfall* dipilih sebagai metode dalam melakukan pengembangan aplikasi. Karies merupakan salah satu penyakit gigi yang menjadi masalah dalam penelitian ini, penyakit ini akan menyebabkan rasa sakit sehingga membuat aktivitas harian terganggu dan mengurangi kualitas hidup dari penderita. Penelitian ini membuat sebuah aplikasi berbasis Android sebagai media kampanye untuk melakukan pencegahan terhadap karies gigi. Aplikasi ini akan berisi *Dental Health Education* (DHE) yang akan meningkatkan pengetahuan pengguna tentang kesehatan gigi.

Andhika Epriliyansyah [14] melakukan penelitian dengan judul "Perancangan Game Edukasi Pengenalan Perhitungan Untuk Anak Usia Dini Dengan Metode RAD Berbasis Android". Metode yang digunakan dalam pengembangan aplikasi ini adalah *Rapid Action Development (RAD)*. Metode ini memiliki tiga tahapan di antaranya adalah Perencanaan Syarat-Syarat, Workshop Desain RAD, dan Implementasi. Masalah yang terdapat dalam penelitian ini adalah banyaknya *game* kekerasan yang disenangi oleh anak-anak sehingga bisa berakibat buruk bagi mental anak-anak. Dibuatlah sebuah aplikasi game yang mendidik bagi anak-anak agar terhindar dari *game* kekerasan maupun *game* yang tidak mendidik lainnya.

Geetika Arora [15] melakukan penelitian dengan judul "*DeepTeeth : A Teeth-Photo Based Human Authentication System For Mobile And Hand-Held Devices*". *Agile* dipilih sebagai metode dalam melakukan pengembangan aplikasi. *Dental Biometrics* sering digunakan untuk mengidentifikasi korban dari bencana alam seperti gempa bumi, angin topan, dan sebagainya. Mengimplementasikan *Dental Biometrics* sebagai sistem autentifikasi ke dalam aplikasi *mobile* menjadi masalah dalam penelitian ini. Penelitian ini membuat *Dental Biometrics* menjadi sebuah modal *biometric* baru untuk melakukan autentifikasi *smartphone*. Sampel *biometric* didapatkan menggunakan kamera *smartphone* dengan bantuan aplikasi Android yang mempunyai *markers* untuk menyimpan area gigi. *Region of Interest (RoI)* kemudian di ekstrak menggunakan *markers* dan sampel yang telah didapatkan ditingkatkan menggunakan *CLACHE* untuk mendapatkan gambar yang lebih baik.

Aplikasi ini juga mengimplementasikan arsitektur *deep learning* dan skema *novel regulation* untuk mendapatkan ukuran *RoI* yang kecil.

Kanij Fatema Bushra [16] melakukan penelitian dengan judul “*Automated Detection Of COVID-19 Fron X-ray Images Using CNN And Android Mobile*”. Prevalensi dari pandemi *corona virus disease* 2019 (COVID-19) telah memberikan dampak yang sangat besar pada kesehatan dan ekonomi global. Penelitian ini membuat sebuah aplikasi AI (*artificial intelligence*) berbasis Android untuk mendeteksi COVID-19 menggunakan gambar X-ray. Model algoritma CNN (*convolutional neural network*) di kembangkan pada MATLAB kemudian dikonversi menjadi sebuah model TensorFlow Lite (TFLite) yang dapat dipasang pada Android. Dilakukan validasi menggunakan *5-fold cross-validation*, didapatkan hasil rata-rata akurasi sebesar 98,85%, kepekaan sebesar 98,49%, ketegasan sebesar 98,82%, ketelitian sebesar 98,81%, dan skor F1 sebesar 98,65%, yang didapatkan dari hasil deteksi COVID-19 menggunakan gambar X-ray.

H Artanto [17] melakukan penelitian dengan judul “*Predicting Potato Diseases Using Tensorflow In Mobile Apps Android*”. Kentang memiliki penyakit umum yaitu *early blight* dan *late blight*, kedua penyakit tersebut merupakan masalah pada bidang pertanian. Dalam mendeteksi penyakit tersebut para petani masih menggunakan cara sederhana yaitu dengan melihat gejala fisik dari kentang menggunakan penglihatan mereka. Penelitian ini membuat sebuah aplikasi ML (*machine learning*) berbasis Android untuk mendeteksi penyakit pada kentang. Dalam mengembangkan aplikasi menggunakan konfigurasi pelatihan SSD Mobilenet v1. ilakukan pengujian dengan hasil sebesar 90% menyatakan dapat mendeteksi dan mengenali tipe penyakit dari kentang.

Tabel 2.1 Penelitian Terdahulu

No	Judul	Penulis	Metode	Hasil
1.	Aplikasi Mobile Untuk Memantau Body Mass Index Dengan Metodologi Scrum [6]	E. Setiawan, H. Prakoso, T. Gunawan	Scrum	Peneliti berhasil mengimplementasikan kamera menggunakan <i>OpenCV</i> , penggunaan kamera ini sebagai salah satu cara untuk mengukur BMI, dengan memiliki tingkat akurasi sebesar 70%.
2.	Rancang Bangun Aplikasi Edukasi Tuberkolosis Menggunakan Metode Scrum [7]	S. Utami, K. Eviyanti, W. Sari	Scrum	Aplikasi dapat berjalan dengan baik dari Android 4.1 keatas. Hasil dari <i>black box testing</i> menunjukkan fungsi aplikasi valid 100%. Pengujian <i>UEQ (User Experience Questioner)</i> mendapatkan <i>feedback</i> positif dari pengguna dengan nilai > 0,8.
3.	Rancang Bangun Sistem Informasi Pariwisata Sulawesi Selatan Berbasis Android Dengan Menggunakan Metode Prototyping [8]	I. Mudar, H. Arfandy	Prototyping	Sistem dikembangkan dapat menyajikan informasi pariwisata mencakup 110 destinasi, 39 event, 45 kuliner, dan foto objek wisata dari 12 kabupaten kota. Hasil pengujian menggunakan <i>test case</i> menunjukkan bahwa fitur yang dikembangkan dapat berfungsi dengan baik dan berhasil menampilkan informasi pariwisata yang sesuai.

No	Judul	Penulis	Metode	Hasil
4.	Sosialisasi Perawatan Gigi dan Mulut pada Anak Berbasis Android [9]	H. Pramudita, A. Riyantomo	MDLC	Dirancang menggunakan <i>Adobe Flash Professional CS6</i> dan <i>CorelDraw X7</i> . Pengujian aplikasi dilakukan langsung ( <i>E2E</i> ) menggunakan <i>smartphone</i> Samsung Galaxy J2 Prime dan didapatkan hasil bahwa aplikasi berjalan dengan baik dan sesuai dengan apa yang diharapkan.
5.	Rancang Bangun Aplikasi <i>Teledentistry</i> berbasis <i>Appstore</i> dan <i>Playstore</i> Pada Manajemen Praktik Dokter Gigi Selama Pandemi Covid-19 Dan Adaptasi Kebiasaan Baru [10]	F. Akbar	Research and Development	Dibuat dengan menggunakan bahasa pemrograman <i>Kotlin</i> , <i>MySQL</i> sebagai <i>RDBMS</i> dan <i>Firebase</i> sebagai media penyimpanan data. Pengujian dilakukan pada setiap fitur yang telah dibuat dan didapatkan hasil bahwa seluruh fungsi dapat bekerja dengan baik.
6.	Implementasi Metode <i>Prototyping</i> Pada Perancangan Aplikasi <i>Electronic Ticket (E-Ticket)</i> Berbasis Android [11]	N. Putri, N. Agung Prabowo, R. Widyanto	Prototyping	Dirancang dengan menggunakan <i>MySQL</i> , <i>Adobe Dreamweaver</i> , dan <i>Android Studio</i> . Pengujian aplikasi menggunakan <i>Black Box Testing</i> dan dipatkan hasil bahwa sistem yang dibuat dapat berjalan dengan baik

No	Judul	Penulis	Metode	Hasil
				sehingga memudahkan calon penumpang dalam mencari informasi ketersediaan tiket bus.
7.	<i>A Mobile App (Smart Dental Alarm) on Improving Tooth Brushing Skills Among Early Childhood</i> [12]	D. Ratna, M. Djamil, T. Wiyatini	Research and Development	Aplikasi ini dianggap efektif untuk meningkatkan kualitas menggosok gigi anak-anak usia dini. Hasil tersebut didapatkan dari uji coba penggunaan aplikasi yang dilakukan selama 21 hari.
8.	<i>Application ‘Senyum Gigiku’ Android Based Media Promotion As Prevention Caries Dental Knowledge And Attitudes Toward Increasing The Mother In District Banyudono PKK</i> [13]	A. Nurrochman, M. Djamil, B. Santoso	Waterfall	Uji coba menggunakan <i>feasibility test</i> didapatkan hasil dengan presentase 85.2%, dan aspek material mendapatkan hasil sebesar 87 %.
9.	Perancangan Game Edukasi Pengenalan Perhitungan Untuk Anak Usia Dini Dengan Metode RAD Berbasis Android [14]	A. Epriliyansyah, W, Verina	Rapid Action Development	Aplikasi dapat berjalan dengan baik pada OS Jelly Bean 4.2 keatas. Hasil pengujian dengan <i>Black Box Testing</i> yang dilakukan oleh 10 <i>tester</i> menyatakan bahwa aplikasi berjalan cukup baik.



No	Judul	Penulis	Metode	Hasil
10.	<i>DeepTeeth: A Teeth-Photo Based Human Authentication System For Mobile And Hand-Held Devices</i> [15]	G. Arora, R. Bharadwaj, K. Tiwari	Agile	Berhasil mengimplementasikan <i>dental biometrics</i> sebagai autentifikasi perangkat <i>smartphone</i> . Diketahui bahwa semakin bersih gambar yang diambil maka akan semakin cepat juga untuk mengidentifikasinya. Model yang digunakan dapat bekerja dengan baik dengan ukuran yang kecil sehingga menambah efisiensi dan cocok untuk perangkat <i>mobile</i> .
11	<i>Automated Detection Of COVID-19 Fron X-ray Images Using CNN And Android Mobile</i> [16]	K. Bushra, M. Ahamed, M. Ahmad	-	Aplikasi Android dikembangkan menggunakan model TFLite untuk mendeteksi COVID-19 menggunakan gambar X-ray. Aplikasi dapat berjalan dengan baik dengan presentase hasil pengujian rata-rata diatas 95%.
12	<i>Predicting Potato Diseases Using Tensorflow In Mobile Apps Android</i> [17]	H. Artanto, F. Arifin	-	Memanfaatkan <i>machine learning</i> untuk mendeteksi sebuah penyakit pada kentang melalui gejala fisik dengan aplikasi Android. Pengujian ketepatan dalam mendeteksi penyakit memiliki presentase sebesar 90%.

Berdasarkan tabel penelitian terdahulu diatas, dapat disimpulkan bahwa penelitian yang akan dibuat ini memiliki perbedaan dengan penelitian sebelumnya, yaitu mengimplementasikan *Machine Learning* sebagai model untuk mendeteksi penyakit kerusakan gigi berbasis Android, dengan memanfaatkan *library* CameraX dan *library* TFLite. *Library* CameraX digunakan untuk membuat tampilan kamera yang dapat dikustomisasi sesuai dengan kebutuhan sistem, kamera disini akan memiliki sebuah *guideline* yang harus diikuti agar hasil dari gambar yang akan digunakan dapat dideteksi oleh aplikasi. *Library* TFLite dipilih sebagai model dari *machine learning* penyakit gigi. Beberapa penelitian terdahulu hanya mengimplementasikan aplikasi sebagai media edukasi penyakit gigi, dan belum mengimplementasikan model TFLite sebagai model dari penyakit gigi.

## 2.2. Landasan Teori

### 2.2.1. Penyakit Gigi

Pada aplikasi penelitian ini hanya dapat mengidentifikasi 2 macam penyakit saja, dikarenakan keterbatasan peneliti dalam mencari *dataset* penyakit gigi. Berikut adalah penyakit-penyakit yang dapat diidentifikasi :

#### a. *Periodontitis*

*Periodontitis* didefinisikan sebagai penyakit inflamasi dari jaringan pendukung gigi yang disebabkan oleh sebuah mikroorganisme spesifik atau sekumpulan mikroorganisme spesifik, sehingga menyebabkan terjadinya sebuah kerusakan progresif pada ligamen *periodontal* dan tulang aveolar dengan adanya pembentukan saku, resesi, atau bahkan keduanya [18].

#### b. *Teeth Discoloration*

*Teeth Discoloration* disebabkan karena adanya *stain* (noda) yang menempel pada permukaan gigi. *Stain* merupakan masalah gigi estetik dan tidak menyebabkan peradangan pada gigitnya [19]. *Stain* dapat diklasifikasikan berdasarkan letaknya, yaitu *Intrinsic Discoloration* yang disebabkan oleh *chromogen* yang tersimpan ke sebagian besar gigi, dan *Extrinsic Discoloration* yang terletak pada permukaan gigi [20]. Selain itu *stain* juga dapat disebabkan oleh faktor lain seperti faktor usia (*aged-related*) [21].

### 2.2.2. Tenaga Kesehatan

Tenaga kesehatan adalah setiap individu yang mengabdikan dirinya dalam bidang kesehatan serta memiliki keterampilan dan pengetahuan melalui pendidikan dalam bidang kesehatan atau jenis tertentu yang memerlukan kewenangan melakukan upaya kesehatan [22].

#### a. Dokter Gigi

Dokter Gigi adalah Dokter Spesialis, lulusan pendidikan kedokteran atau kedokteran gigi baik di dalam maupun di luar negeri yang diakui oleh Pemerintah Republik Indonesia sesuai

dengan peraturan perundang-undangan yang berlaku [23]. Ruang lingkup dari Kedokteran Gigi mencakup penyimpangan, perubahan atau keadaan tidak optimal dari fungsi stomatognatik, yang meliputi fungsi estetika, fungsi bicara, fungsi pengunyahan, dan fungsi persyarafan [24].

b. Terapis Gigi

Terapis Gigi merupakan salah satu tenaga kesehatan yang mempunyai kemampuan di bidang promotif, preventif dan kuratif serta mampu berkolaborasi dengan tenaga kesehatan lain dalam mengatasi permasalahan kesehatan gigi [25].

c. Teknisi Gigi

Teknisi Gigi adalah orang yang telah menyelesaikan Diploma III Teknik Gigi sesuai dengan peraturan perundang-undangan yang berlaku. Teknisi gigi mempunyai tugas pokok dalam melaksanakan pelayanan laboratorium, yang meliputi bidang pembuatan gigi tiruan cekat (*prothesa cekat*), gigi tiruan lepasan (*prothesa lepasan*), alat *orthodonti* dan alat rehabilitasi gigi (*protensa maxillofacial*) [26].

### 2.2.3. Kotlin

Kotlin merupakan bahasa pemrograman yang berbasis JVM (*Java Virtual Machine*) yang dibuat oleh *JetBrains*, sebuah perusahaan yang menciptakan *Intellij*, dan *development tools* lainnya seperti *WebStrom*, *PyCharm*, *ReSharper*, dan lainnya [27][28][29]. Proyek Kotlin dimulai pada tahun 2010 sebagai sebuah proyek *open source* dibawah lisensi Apache 2.0 [27].

Kotlin pertama kali dirilis pada tahun 2016 bulan Februari dengan versi 1.0 [29]. Saat penelitian ini dibuat sudah mencapai versi 1.7.20 per rilis 29 September 2022. Pada Google I/O yang diselenggarakan pada tahun 2017, Google menetapkan Kotlin masuk kedalam bahasa pemrograman resmi untuk

mengembangkan Android dan memberikan dukungan *first-class support* [27][28][30].

Kotlin memiliki fitur yang tidak dimiliki oleh bahasa pemrograman Java :

a. Expressiveness

Kotlin meminimalisir adanya *boilerplate* (penulisan suatu baris kode secara berulang di banyak tempat tanpa adanya perubahan) dikarenakan pola umum penulisan kode sudah dicakup seluruhnya oleh Kotlin itu sendiri [28].

b. Null Safety

Kotlin mampu melakukan pemeriksaan kesalahan pada waktu kompilasi, sehingga memungkinkan pengembang untuk mencegah kesalahan tersebut. Kotlin bahkan menghilangkan *NullPointerException*. Kotlin akan membedakan antara objek yang tidak/dapat bernilai *null* pada saat objek itu dibuat [28].

c. Extension Functions

Kotlin dapat memungkinkan menambahkan sebuah fungsi baru dari sebuah kelas ke kelas lainnya, tanpa harus melakukan pewarisan dari kelas tersebut [28].

d. Functional Support

Kotlin dapat membuat sebuah kelas yang disebut dengan *anonymous function* atau *function literal*. Disebut *anonymous* karena *lambda* tidak memiliki nama seperti fungsi pada umumnya, meskipun demikian *lambda* masih memiliki daftar parameter, *body*, dan *return type* [28].

#### 2.2.4. Android

Android adalah *open source mobile operating system*, yang berbasis Linux [31]. Android dibuat khusus untuk *smartphone* dan tablet. Berbagai produsen telah menggunakan Android sebagai sistem operasi *device* yang

mereka produksi. Android juga memiliki sebuah *store* dengan lebih dari 2.5 Milyar pengguna yang aktif perbulannya [31] [32].

Pemilihan Android sebagai basis pengembangan, dikarenakan Android membuat pengguna nyaman dengan fitur yang canggih dan tampilan yang menarik. Melihat dari sisi pengembang, Android memiliki banyak keunggulan diantaranya adalah :

a. Sistem Operasi Smartphone Terpopuler

Pada tahun 2013, Android berhasil menjadi *OS* terlaris pada *smartphone* dan tablet. Saat ini *market* Android sedikitnya 70% dari total penjualan *smartphone* di tingkat global [33]. Tercatat pada tahun 2022, Android *store* telah memiliki lebih dari 2.6 Juta aplikasi per Juni [34].

b. Android Store

Aplikasi Android bisa didistribusikan menggunakan *web*, *copy APK*, dan *store*. Android *store*, yaitu *Google Play Store*, merupakan cara termudah bagi para pengembang untuk mendistribusikan aplikasinya ke pasar dengan miliaran pengguna.

c. *Development Kit*

Android *Software Development Kit* (SDK) merupakan kit yang dapat digunakan oleh para pengembang untuk mengembangkan aplikasi berbasis Android. Di dalamnya, terdapat berbagai *tools* yang sangat membantu, seperti *debugger*, *software libraries*, *emulator*, *documentation*, *sample code*, dan *tutorial*.

### 2.2.5. Android Studio

*Integrated Development Environment* (IDE) merupakan salah satu *tools* yang merekat dengan pemrograman. Pada tahun 2014, Google mengeluarkan sebuah IDE khusus untuk mengembangkan Android yang bernama Android Studio berbasis IntelliJ IDEA.

### 2.2.6. Machine Learning

*Machine Learning* merupakan teknologi pengembangan algoritma komputer yang mampu untuk meniru atau mensimulasikan kecerdasan manusia. *Machine Learning* mengambil beberapa ide dari banyak disiplin ilmu, seperti kecerdasan buatan, probabilitas dan statistik, ilmu komputer, teori informasi, psikologi, teori kontrol, dan filsafat. Teknologi *Machine Learning* sudah diaplikasikan dalam berbagai bidang seperti pengenalan pola, visi komputer, astronautika, keuangan, hiburan, ekologi, biologi komputasi, dan biomedis dan aplikasi medis [35].

- Klasifikasi

Klasifikasi merupakan salah satu jenis tugas dari *Machine Learning* yang bertujuan untuk memprediksi kelas atau label output dari suatu data berdasarkan contoh-contoh pelatihan yang sudah memiliki label [36].

### 2.2.7. TensorFlow

TensorFlow merupakan *library open source* yang dikembangkan oleh Google Brain Team yang digunakan untuk berbagai macam tugas seperti *Machine Learning*, *Deep Learning*, dan *Data Science*. TensorFlow umumnya digunakan untuk mengembangkan dan menerapkan model *Machine Learning* [37]. TensorFlow memungkinkan para pengembang untuk membuat dan melatih model dengan beragam algoritma yang berbeda. Selain itu, TensorFlow menyediakan berbagai tools untuk memvisualisasikan dan me-debug proses pelatihan model, serta tools untuk men-deploy model yang telah dilatih. [38].

### 2.2.8. Android Library

*Library* dapat berisi komponen-komponen yang dibutuhkan untuk membuat sebuah aplikasi seperti *source code*, *classes*, *resource file*, dan *android manifest*. *Library* merupakan *tools* atau *source code* yang sudah dibuat oleh pihak lain (pihak ketiga) dan bisa dipakai [39]. *Library* utama yang akan digunakan pada penelitian ini adalah CameraX dan TFLite.

a. CameraX

CameraX merupakan bagian dari *Jetpack* yang terdapat pada kategori UI komponen Media. CameraX digunakan untuk membantu mempermudah pengembangan dalam melakukan kustomisasi tampilan kamera [40].

b. TFLite

TFLite merupakan *library* yang dapat digunakan untuk menjalankan model *machine learning* pada aplikasi Android [41].

### 2.2.9. Firebase

Firebase merupakan layanan *cloud* sebagai *Backend as a Service (BaaS)* yang mendukung untuk menyimpan data dan autentifikasi pengguna, sehingga memudahkan pengembang dalam mengembangkan suatu aplikasi (web dan *mobile*) dengan lebih cepat tanpa harus memikirkan infrastruktur [42][43].

Firebase dirilis pada tahun 2016, dengan tujuan untuk memberikan *tools* dan infrastruktur yang dibutuhkan untuk membangun sebuah aplikasi. Firebase pertama kali dibuat oleh perusahaan bernama Envolv pada tahun 2011 [43]. Saat ini Firebase sudah memiliki lebih banyak teknologi yang dapat digunakan baik secara gratis maupun berbayar, yang dapat disesuaikan dengan kebutuhan aplikasi, pada penelitian ini yang digunakan adalah :

a. Firebase Authentication

Firebase Authentication didesain sebagai SDK untuk membuat autentifikasi pada aplikasi dengan menggunakan *email* dan *password*, atau menggunakan *provider* seperti Google Mail, Facebook, Twitter, dan GitHub. Dapat dikombinasikan juga dengan autentifikasi yang telah dibuat pada aplikasi [43].

b. Cloud Firestore

Cloud Firestore merupakan *database* terbaru untuk pengembangan *mobile*. Dibangun atas kesuksesan *Realtime*



*Database*, dengan fitur, layanan dan data model yang lebih baru [44].

c. Firebase Cloud Storage

Cloud Storage merupakan penyimpanan yang digunakan untuk menyimpan dan menyajikan data konten seperti foto dan video [43].

### 2.2.10. Android Jetpack Component

Google memperkenalkan Android Jetpack pada tahun 2018 lebih tepatnya pada Google I/O 2018, Android Jetpack didesain untuk mempermudah pengembang dalam membuat aplikasi Android yang modern dan dapat diandalkan, Jetpack terdiri dari berbagai macam *tools*, *libraries*, dan *architectural guidelines* [45]. Komponen-komponen tersebut dibagi menjadi empat kategori, seperti pada Gambar 2.1.



Gambar 2.1 Android Jetpack Component [46]

a. Architecture

Architecture Component mempunyai kelas yang membantu untuk mengelola alur hidup (*lifecycle*) dari UI Component, menangani data, dan lainnya [47].

b. UI

UI Component membantu untuk membuat tampilan UI dari aplikasi agar lebih menarik sehingga meningkatkan *user experience* [47].

c. Behaviour

Behaviour Component membantu untuk membuat sebuah desain yang kokoh, *testable*, dan aplikasi yang dapat lebih dipelihara (*maintanable*) [47].

d. Foundation

Foundation Component menyediakan kemampuan *core system*, *extensions* Kotlin dan dukungan untuk *multidex* dan *automated testing* [47].

Dari keempat kategori diatas, pada penelitian ini akan menggunakan beberapa komponen *Jetpack*, diantaranya adalah :

a. Data Binding

Merupakan sebuah fitur yang memungkinkan untuk melakukan *binding* keseluruhan properti yang ada didalam *view XML*. *Library* ini secara otomatis akan memberikan akses ke semua komponen *view* yang telah dibuat pada *XML*. Dengan memanfaatkan *view binding* tidak perlu lagi untuk memanggil komponen *view* dengan *findViewById()* atau library lainnya [28].

b. LiveData

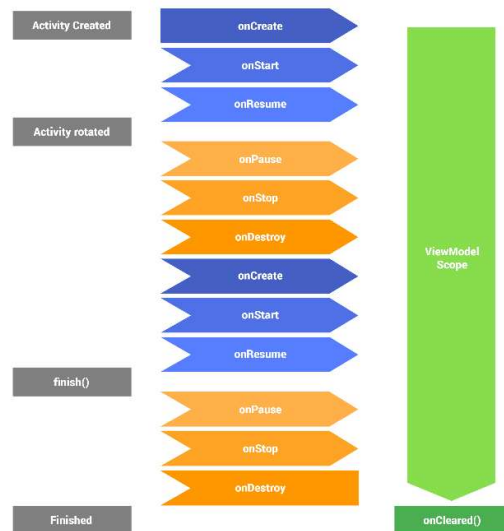
Observable class holder yang akan memberikan pemberitahuan jika terjadi adanya perubahan data. *LiveData* merupakan *Lifecycle-Aware*. Sehingga, dapat merespon perubahan pada *Lifecycle* (*Activity*, *Fragment*, atau *Service*). *LiveData* hanya akan memperbarui komponen *observers* ketika *Lifecycle* dari aplikasi dalam keadaan aktif [48].

### c. Navigation

Merupakan interaksi yang dapat memungkinkan pengguna untuk melihat, masuk, dan keluar dari berbagai konten dalam aplikasi. Komponen ini akan membantu menerapkan navigasi, mulai dari interaksi sederhana hingga interaksi yang kompleks. Komponen navigasi memastikan pengalaman pengguna yang konsisten dan dapat diprediksi [49].

### d. ViewModel

Kelas *ViewModel* akan membantu UI Controller untuk menyiapkan data yang ditampilkan ke UI. Objek *ViewModel* akan selalu dipertahankan selama ada *configuration changes*. Dengan begini data yang dimilikinya akan segera tersedia untuk *Activity* atau *Fragment* selanjutnya, yang tertera pada Gambar 2.2.



Gambar 2.2 *ViewModel* [50]

### e. Animation

Animasi pada Android digunakan untuk menangani perubahan pada tampilan, mempercantik tampilan, dan memberikan isyarat visual kepada pengguna. Dalam Android terdapat berbagai macam jenis animasi, diantaranya *Property Animation*, *Shared Element*, dan *Motion Layout*, dalam

pembuatan aplikasi penelitian ini akan menggunakan *Property Animation*.

f. Permission

*Permission* digunakan untuk mengizinkan aplikasi untuk mengakses data yang sensitif. Dengan memanfaatkan *permission* pengembang dapat membuat fungsi dari aplikasi lebih maksimal, seperti mengakses internet, lokasi peranti pengguna, mengakses kamera, dan sebagainya. Dengan memanfaatkan *permission* secara efisien, akan membantu pengguna untuk mengontrol, transparansi, dan minimalisasi data [51].

### 2.2.11. Android Architecture Pattern

Aplikasi Android akan terus berkembang dan bertambah ukuran. *Architecture* penting untuk memungkinkan aplikasi berkembang, meningkatkan kekokohan aplikasi, dan membuat aplikasi agar mudah di uji (*testing*). *Architecture* akan mendefinisikan batas-batas bagian dari aplikasi dan bertanggung jawab pada bagian-bagiannya [52]. *Architecture patterns* mengimplementasikan *separate the concerns* (memisahkan kode berdasarkan fungsinya masing-masing). Android *Architecture Patterns* yang paling sering digunakan di antaranya, MVC (*Model-View-Controller*), MVP (*Model-View-Presenter*), MVVM (*Model-View-ViewModel*) dan *Clean Architecture* [53]. Pada penelitian ini akan menggunakan MVVM.

- MVVM (Model-View-ViewModel)

Merupakan salah satu *architecture pattern* yang disarankan oleh Tim Android langsung. Dengan membagi menjadi tiga lapisan kode yang terpisah.

- Model

Bertanggung jawab untuk abstraksi sumber data. *Model* dan *ViewModel* bekerja sama untuk mendapatkan data dan menyimpan data.

- View  
Tujuannya untuk memberitahukan *ViewModel* mengenai aksi yang dilakukan oleh pengguna.
- ViewModel  
Memperlihatkan aliran data yang relevan dengan *View*.

#### 2.2.12. Low-Fidelity

Low-Fidelity (*Lo-Fi*), merupakan sebuah representasi yang masih berupa desain sederhana dan kasar dari produk yang akan dibuat. Desain ini tidak menunjukkan detail dari warna dan ilustrasi yang akan digunakan, namun hanya berupa beberapa simbol dan kerangka saja. Desain ini akan sangat cocok digunakan jika akan menuangkan ide dengan cepat ke dalam rancangan desain [54].

#### 2.2.13. High-Fidelity

High-Fidelity (*Hi-Fi*) merupakan desain lanjutan dari *Lo-Fi* yang sudah diperhalus dan lebih kompleks. Desain ini sudah menunjukkan pemilihan dari warna dan ilustrasi yang akan digunakan. Begitu juga dengan *font* teks pada setiap komponennya persis sebagaimana aplikasi yang akan dipakai oleh calon pengguna [54].

#### 2.2.14. Scrum

Scrum adalah sebuah proses untuk mengatur dan mengontrol pengembangan produk dan *software* [55]. Scrum merupakan perkembangan dari metodologi *Agile*, karena dasarnya mengimplementasikan filosofi *Agile*. Cara kerja dari Scrum adalah dengan membentuk sebuah tim untuk memberikan hasil kerja yang produktif, kreatif, memiliki nilai yang setinggi mungkin untuk dapat menyelesaikan masalah adaptif yang rumit. Dalam *Scrum* memiliki tiga pilar yang harus diterapkan untuk memperkokoh setiap implementasi dari proses empiris adalah : transparansi, inspeksi dan adaptasi [56].

a. Transparansi

Informasi yang relevan tentang tugas dan proses harus tersedia dan dapat diakses oleh Tim Scrum. Hal ini mencakup pemahaman yang jelas tentang tujuan, tugas yang harus dikerjakan, serta kemajuan dan kendala yang dihadapi [56].

b. Inspeksi

Tim secara teratur melakukan evaluasi terhadap pekerjaan yang dilakukan untuk mengidentifikasi permasalahan, kesalahan, peluang perbaikan. Tujuan dari inspeksi untuk meningkatkan kualitas dan efektivitas kerja [56].

c. Adaptasi

Tim mengambil tindakan perbaikan berdasarkan hasil inspeksi. Jika ada masalah atau perubahan, tim melakukan penyesuaian dan perubahan diperlukan untuk mencapai hasil yang diinginkan [56].

Pada Scrum juga mempunyai *role* yang memiliki peran dan tugasnya masing-masing.

a. Product Owner

Merupakan orang yang bertanggung jawab atas produk yang dikembangkan, mengetahui produk yang dibuat bukan hanya dari sisi teknis, dapat menempatkan diri dari berbagai sudut pandang untuk memberikan *feedback* kepada tim, dan mempunyai kemampuan yang lebih tinggi dari *Scrum Master*, karena *Product Owner* akan bertanggung jawab untuk memaksimalkan hasil dari produk yang dibuat oleh tim [55].

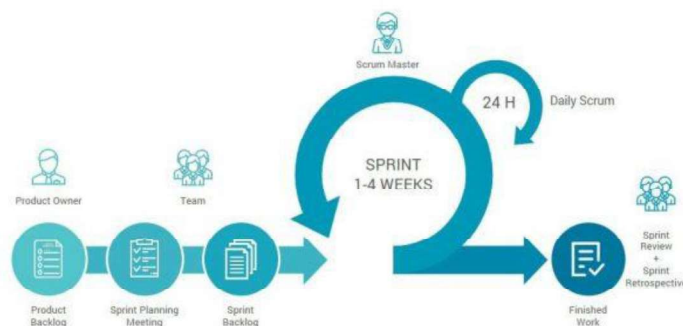
b. Scrum Master

Merupakan orang yang bertanggung jawab untuk memfasilitasi seluruh rapat, memastikan transparansi, dan yang paling penting adalah membantu tim untuk menemukan apa masalah yang menjadi kendala saat ini atau kedepannya [55].

### c. Development Team

Berisi individu yang akan menjalankan proyek dari produk / *software* yang akan dibuat, setiap individu bertanggung jawab terhadap *to-do list* mereka masing-masing untuk mengerjakan dan menyelesaikan *Product Backlog* [55].

Sebelum memulai sprint, tim akan menentukan *Product Backlog*, yang merupakan sebuah daftar dari seluruh kebutuhan yang harus dibangun dan diselesaikan. *Product Backlog* ini nantinya akan dibahas oleh tim pada tahap awal dari pengerjaan *sprint* [55], seperti yang tertera pada Gambar 2.3.



Gambar 2.3 Alur kerja metodologi *SCRUM* [57]

#### a. Sprint Planning

Tahapan awal dari *Scrum* dimana seluruh tim melakukan rapat perencanaan yang membahas hal yang perlu dilakukan pada *sprint* dan menyiapkan *sprint* untuk kedepannya [55].

#### b. Daily Stand-up

Tahapan yang menjadi inti dari *Scrum*, dimana tim melakukan rapat setiap hari yang membahas mengenai kemajuan yang telah dilakukan dari masing-masing anggota tim [55].

#### c. Sprint Demo

Pada tahapan ini masing-masing anggota tim menunjukkan hasil dari *backlog* yang telah diselesaikan pada saat pengerjaan *sprint* [55].

#### d. Sprint Retrospective

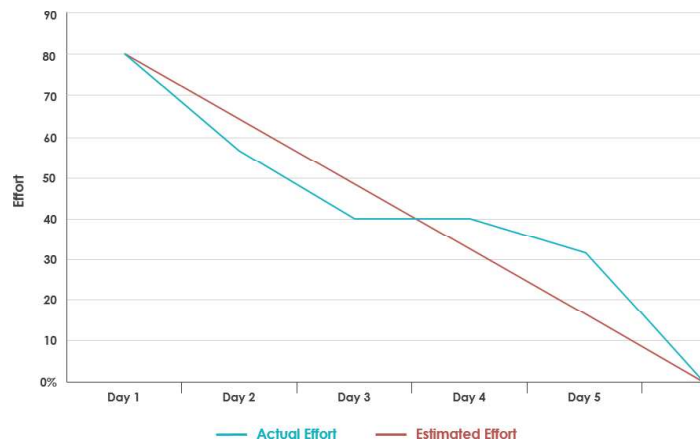
Setelah keseluruhan *sprint* selesai, diakhiri dengan melakukan retrospektif. Masing-masing anggota tim akan mengevaluasi *sprint* yang telah selesai dilakukan sebelumnya, dan memberikan *feedback* untuk mendapatkan kinerja yang lebih baik untuk kedepannya [55].

### 2.2.15. Report Chart

*Report Chart* merupakan gambaran dalam bentuk grafis yang digunakan didalam kerangka kerja *Scrum* untuk membantu tim memahami kemajuan dari proyek. *Report Chart* ini membantu tim untuk melihat pekerjaan yang sudah selesai, pekerjaan yang belum selesai, pekerjaan yang harus dilakukan dan kecepatan pengerjaan.

- Burndown Chart

Burndown Chart merupakan sebuah gambaran grafis diagram yang menunjukkan penyelesaian tugas per-harinya berdasarkan tugas yang telah direncanakan. Tujuannya adalah agar proyek yang sedang dikerjakan berada pada jalur dan selesai sesuai dengan jadwal yang telah direncanakan [58].



Gambar 2.4 Burndown Chart [59]

Burndown Chart terdiri dari dua parameter yaitu sumbu x yang merepresentasikan lama pengerjaan dari sprint, dan sumbu y yang merepresentasikan *story points* [60].



*Story point*, sebuah unit penilaian yang digunakan untuk mengestimasi waktu pengerjaan tugas, pemberian nilai *story point* pada *product backlog* akan didasarkan oleh pengalaman tim sebelumnya dalam mengerjakan suatu proyek [60].

Grafis Burdown Chart memiliki dua garis, garis ideal dan garis aktual. *Ideal line* menunjukkan jumlah tugas yang seharusnya sudah selesai pada setiap titik waktu dalam *sprint*, sementara *actual line* menunjukkan jumlah tugas yang benar-benar sudah selesai pada setiap titik waktu dalam *sprint* [60].

Dengan menggunakan Burdown Chart, tim dapat memantau apakah pengerjaan tugas sudah mengikuti jadwal yang telah direncanakan atau belum. Jika garis *actual line* lebih tinggi dari *ideal line*, itu menunjukkan bahwa tim sedang mengalami keterlambatan dalam menyelesaikan tugas. Sebaliknya jika garis *actual line* lebih rendah dari *ideal line*, itu menunjukkan bahwa tim bekerja dengan cepat [60].

#### **2.2.16. Testing**

*Testing* atau pengujian merupakan bagian terpenting dalam mengembangkan sebuah aplikasi, *testing* sendiri digunakan untuk mengukur kode yang telah dibuat, dan agar dapat mudah di *maintanance* kemudian hari. *Testing* memiliki tujuan untuk mendapatkan informasi mengenai kualitas dari aplikasi, dan memastikan bahwa aplikasi yang telah dibuat sudah sesuai dengan kebutuhan sebelum dapat digunakan [61]. Dalam penelitian ini akan menggunakan UI Test, Blackbox Testing dan Compatibility Test.

##### **a. UI Test**

UI Test merupakan mekanisme di mana aplikasi akan diuji sesuai dengan kondisi pengguna ketika berinteraksi pada sebuah aplikasi untuk memastikan apakah fungsi dari tampilan aplikasi sudah sesuai. Pengujian ini dapat dilakukan dengan dua cara,

dengan manual atau secara otomatis dengan memanfaatkan beberapa *frameworks* [62].

b. Black Box Testing

Black Box Testing merupakan sebuah pengujian yang hanya berdasarkan dengan kebutuhan sistem dan tidak perlu melakukan pemeriksaan kode pada *black box testing*. Pengujian ini dilakukan berdasarkan sudut pandang pengguna. *Black box testing* hanya dilakukan ketika produk sudah benar-benar jadi [63].

c. Compatibility Test

Compatibility Test merupakan pengujian untuk menguji aplikasi dengan berbagai macam versi Android, terutama Android versi terbaru [64].

d. System Usability Scale

System Usability Scale (SUS) adalah kuesioner standar yang banyak digunakan untuk menilai persepsi tentang kemudahan penggunaan suatu sistem [65].

### 2.2.17. Unified Modeling Language





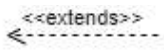
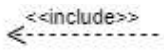
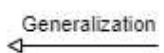
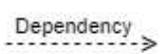
Unified Modeling Language (UML) merupakan sebuah bahasa standar untuk menulis kerangka dari perangkat lunak. UML digunakan untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan komponen-komponen dari sistem perangkat lunak [66].

UML sangat cocok digunakan untuk memodelkan sistem mulai dari sistem informasi perusahaan hingga aplikasi berbasis web terdistribusi dan bahkan dapat digunakan untuk sistem tertanam. UML merupakan bahasan yang sangat ekspresif, menangani semua pandangan yang dibutuhkan untuk mengembangkan dan meluncurkan sistem/*software* [66]. Berikut adalah alat bantu yang umum digunakan dalam perancangan berbasis objek menggunakan UML :

a. Use Case Diagram

Use Case Diagram merupakan pemodelan yang menggambarkan hubungan antara perilaku (*behaviour*) sistem dan aktor. Use Case Diagram sangat penting digunakan untuk mengatur dan mengilustrasikan perilaku dari sistem [66]. Berikut adalah representasi visual dari Use Case Diagram, yang dapat dilihat pada Tabel 2.2 :

Tabel 2.2 Use Case Diagram




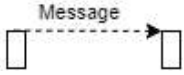
No	Gambar	Keterangan
1	 Actor	Peran dari <i>user</i> yang akan berinteraksi dengan <i>use case</i> . Aktor merepresentasikan peran manusia, perangkat keras, atau bahkan sistem.
2	 Use Case	Sebuah fungsi dari sistem yang akan dibuat dan dihubungkan dengan aktor.
3	 Association	Untuk menghubungkan antara satu objek dengan objek lainnya.
4	 System	Menspesifikasikan sebuah paket dari suatu sistem untuk membatasi <i>use case</i> berinteraksi dengan luar sistem.
5	 <<extends>>	Menspesifikasikan target dari suatu <i>use case</i> untuk menambahkan perilaku.
6	 <<include>>	Menspesifikasikan sumber dari <i>use case</i> berasal secara eksplisit.
7	 Generalization	Berupa hubungan <i>child</i> yang mewarisi implementasi dari <i>parent</i> .
8	 Dependency	Hubungan yang memengaruhi objek yang terhubung dengan

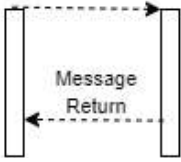

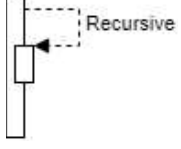
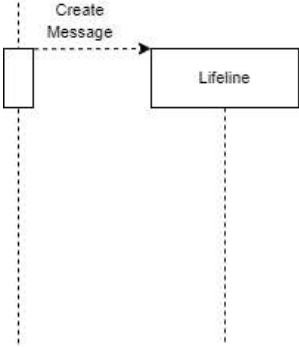
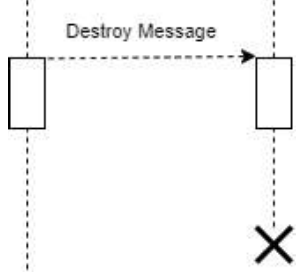
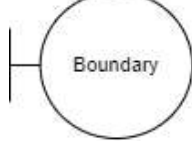

No	Gambar	Keterangan
		suatu objek lain yang mengalami perubahan.

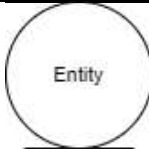
b. Sequence Diagram

Sequence Diagram merupakan sebuah diagram yang berupa interaksi atau komunikasi. Sequence Diagram menunjukkan sekumpulan objek dan pesan yang dikirim dan diterima oleh objek tersebut [66]. Berikut adalah representasi visual dari Sequence Diagram, yang dapat dilihat pada Tabel 2.3 :

Tabel 2.3 Sequence Diagram

No	Gambar	Keterangan
1		Peran dari sebuah entitas yang akan berinteraksi dengan subjek.
2		Sebuah garis putus vertikal yang merepresentasikan objek yang berinteraksi.
3		Merepresentasikan sebuah waktu dari suatu objek saat melakukan eksekusi
4		Menggambarkan sebuah komunikasi antara <i>lifeline</i> yang saling berinteraksi.



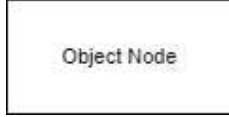

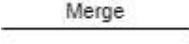
No	Gambar	Keterangan
5		Pesan yang didapatkan dari hasil komunikasi antara <i>lifeline</i> yang saling berinteraksi.
6		Sebuah pesan yang mewakili pesan dari <i>lifeline</i> yang sama.
7		Pesan yang dikirimkan untuk <i>lifeline</i> itu sendiri. Menargetkan aktivasi di atas aktivasi dari pesan itu berasal.
8		Pesan yang merepresentasikan <i>instance</i> dari <i>lifeline</i> tujuan.
9		Pesan yang dihapus oleh alur hidup <i>lifeline</i> yang dituju.
10		Menggambarkan <i>activity</i> yang akan dilakukan.
11		Merepresentasikan penghubung antara <i>boundary</i> dengan tabel

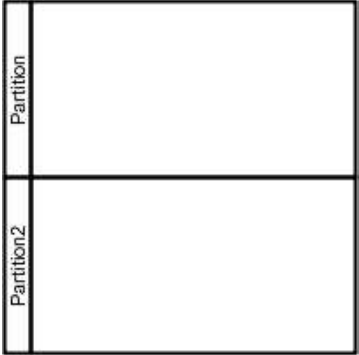
No	Gambar	Keterangan
12		Menghubungkan <i>activity</i> yang akan dilakukan.

c. Activity Diagram

Activity Diagram menunjukkan alur dari *activity* ke *activity* lain yang berada di dalam sistem [66]. Berikut adalah representasi visual dari Activity Diagram, yang dapat dilihat pada Tabel 2.4 :

Tabel 2.4 Activity Diagram

No	Gambar	Keterangan
1		<i>Initial Node</i> , Menggambarkan titik awal dari aksi atau <i>activity</i> .
2		<i>Final Node</i> , Menggambarkan titik akhir dari aksi atau <i>activity</i> .
3		Menggambarkan objek yang terhubung dengan alur <i>activity</i> .
4		Percabangan pemilihan dari <i>activity</i> yang lebih dari satu.
5		Penggabungan dari <i>activity</i> yang berbeda menjadi satu <i>activity</i> .

No	Gambar	Keterangan
6	 The diagram shows a large rectangle divided into two horizontal sections. The top section is labeled 'Partition' and the bottom section is labeled 'Partition2'. The labels are oriented vertically along the left side of their respective sections.	<i>Swirlane</i> memisahkan logika yang berbeda dari masing-masing <i>activity</i> .