

## **BAB 2**

### **DASAR TEORI DAN TINJAUAN PUSTAKA**

#### **2.1. Dasar Teori**

##### **2.1.1. Android Studio**

Android Studio adalah lingkungan pengembangan terintegrasi (*IDE*) untuk pengembangan *platform Android*. Dulu diumumkan pada 16 Mei 2013 di konferensi Google I/O oleh Manajer Produk Google, Katherine Chou. (Thamizharasi, 2016) Beberapa fitur-fitur yang ada di *Android Studio* antara lain:

- *Build* berbasis *Gradle* yang fleksibel
- Alat yang terpadu untuk mengembangkan aplikasi untuk semua perangkat *Android*.
- Mengedit secara langsung untuk mengupdate *composable* di emulator dan perangkat fisik secara *real time* yang terhubung.
- *Framework* dan *dependency* yang lengkap.
- Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*.

##### **2.1.2. Android**

*Android* adalah sistem operasi seluler yang didasarkan pada versi modifikasi dari Linux. Awalnya dikembangkan oleh *startup* dengan nama yang sama, *Android, Inc.* Pada tahun 2005 (Thamizharasi, 2016). *Android* dibangun sepenuhnya oleh Google Inc. dan bersifat terbuka, sehingga pengembang dapat menggunakannya tanpa membeli lisensi dari Google dan membangunnya tanpa batasan. Untuk memulai pengembangan aplikasi yang menggunakan bahasa pemrograman *Java* pada *platform Android*, Kit Pengembangan *Software Android (SDK)* menyediakan alat dan *Interface Pembelajaran Aplikasi (API)*. *Android*, yang dikenalkan sejak tahun 2008, terus mempertahankan keunggulannya di pasar teknologi. Menurut *International Data Corporation (IDC)*, *Android* memiliki jumlah pengguna yang lebih besar daripada *iOS*. *Android* menawarkan pendekatan yang menyeluruh dalam pengembangan aplikasi. Artinya, satu aplikasi *Android* yang dibangun dapat berjalan diberbagai

perangkat yang menggunakan sistem operasi *Android* baik itu *smartphone*, *smartwatch*, *tablet*, dan perangkat lainnya.(Irvansyah et al, 2020)

Tabel 2.1 Versi *Android*

<b>Android Version</b>	<b>Nama</b>	<b>Rilis</b>
1.0	Android One	23 September 2008
1.1	Petit Four	9 Februari 2009
1.5	Cupcake	27 April 2009
1.6	Donut	15 September 2009
2.1	Eclair	11 Januari 2010
2.2.3	Froyo	20 Mei 2010
2.3.7	Gingerbread	9 Februari 2011
3.2.6	Honeycomb	15 Juli 2011
4.0.4	Ice Cream Sandwich	16 Desember 2011
4.3.1	Jelly Bean	24 Juli 2013
4.4W.2	Key Lime Pie	25 Juni 2014
5.1.1	Lemon Meringue Pie	2 Maret 2014
6.0.1	Macadamia Nut Cookie	2 Oktober 2015
7.1.2	New York Cheesecake	4 Oktober 2016
8.1	Oatmeal Cookie	5 Desember 2017
9	Pistachio Ice Cream	6 Agustus 2018
10	Quince Tart	3 September 2019
11	Red Velvet Cake	8 September 2020
12	Snow Cone	4 Oktober 2021
12.1	Snow Cone v2	7 Maret 2022
13	Tiramisu	15 Agustus 2022
14	Upside Down Cake	8 Februari 2023

### 2.1.3. Bahasa Pemrograman Kotlin

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine (JVM)* yang dikembangkan oleh JetBrains. *Kotlin* merupakan bahasa pemrograman yang pragmatis untuk *Android* yang mengkombinasikan *object oriented (OO)* dan pemrograman fungsional. *Kotlin* juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu *project* dengan bahasa pemrograman *Java*. Bahasa pemrograman ini juga dapat digunakan untuk

pengembangan aplikasi berbasis *desktop*, *web* dan bahkan untuk *backend*(Sibarani et al, 2018). Sehingga bahasa pemrograman *kotlin* merupakan Bahasa pemrograman yang dapat mengembangkan aplikasi yang *multiplatform* secara efektif.

#### **2.1.4. Manual Testing**

*Manual Testing* adalah teknik pengujian di mana teknisi pengujian menyiapkan kasus pengujian secara manual dan mengeksekusinya untuk mengidentifikasi *bug/error* pada perangkat lunak. Ini adalah metode pengujian perangkat lunak yang paling ketat dan lama. *Manual Testing* atau Pengujian manual adalah aktivitas yang melelahkan yang mengharuskan penguji mengharuskan untuk menjadi sabar, jeli, spekulatif, kreatif, inovatif, berpikiran terbuka, dan terampil. Pengujian manual berulang mungkin sulit dilakukan pada aplikasi perangkat lunak besar atau aplikasi yang memiliki kumpulan data yang sangat besar. (Sharma, 2014)

Seperti manajemen sebuah Langkah-langkah terkait pembuatan fitur yang mengharuskan dari teknisi atau *Quality Assurance* mengetahui standar yang digunakan dalam pembuatan aplikasi, sehingga fitur-fitur pada aplikasi pengujian dapat dilakukan. Biasanya dalam manual *testing* terdapat masalah seperti pengujian tidak dapat di program sehingga rawan akan kesalahan oleh teknisinya sendiri.

#### **2.1.5. Unified Modelling Language (UML)**

*UML* merupakan suatu cara dalam menunjukkan proses bisnis yang terjadi pada suatu aplikasi dengan mewakilinya dalam suatu aktor dan juga fungsional seperti apa saja yang bisa dilakukan. Biasanya *UML* digunakan dalam pengembangan aplikasi berorientasi objek. *UML* bisa digunakan untuk melakukan visualisasi, merancang dokumentasi di level fitur program dan juga menjadi *blueprint* dalam pengembangan *software* (Soetrisno et al, 2022).

#### **2.1.6. Extensible Markup Language (XML)**

Pada pengembangan aplikasi *testing* untuk pembuatan *User Interface* atau *layout* tampilan menggunakan *XML*. *XML (Extensible Markup Language)* di manfaatkan dalam mendefinisikan dokumen dengan format standar dimana yang dapat dibaca dan di dukung oleh aplikasi-aplikasi *xml* yang kompatibel. Bahasa format *xml* bisa digunakan dengan halaman *html*, akan tetapi *xml* itu sendiri bukan bahasa

*markup*. Sebaliknya, *xml* itu merupakan “metabahasa” yang dapat di pakai dalam membuat bahasa *markup* untuk aplikasi khusus.(Owandity et al, 2022)

Semua elemen yang terdapat pada *layout* dibuat dengan menggunakan hierarki objek *View* dan *View Group*. *View* biasanya berisikan tentang elemen yang terlihat dan pengguna dapat melakukan interaksi dengan elemen tersebut. Sedangkan, *View Group* merupakan *container* tak terlihat yang menentukan struktur tata letak bagi *View* dan objek *View Group*. Objek *View* yang biasanya disebut “*widget*” dapat berupa salah satu dari banyak *subclass* seperti *Button*, atau *TextView*. Objek *ViewGroup* yang biasanya disebut “tata letak” dapat berupa salah satu dari banyak jenis yang menyediakan berbagai struktur tata letak.

#### **2.1.7. Web Service**

*Web service* adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin. Sistem-sistem lainnya berinteraksi dengan *web service* menggunakan pesan *SOAP* yang umumnya dikirim melalui *HTTP* dalam bentuk *XML*.

Definisi diatas diberikan oleh *World Wide Web Consortium(W3C)* yang merupakan badan yang menciptakan dan mengembangkan standar *web service*. Tetapi secara umum, *web service* tidak terbatas hanya pada standar *SOAP* saja. Salah satu pustaka yang mengulas lengkap tentang *web service* menyebutkan definisi yang lebih umum: *web service* adalah aplikasi yang diakses melalui internet menggunakan protokol standar internet dan menggunakan *XML* sebagai format pesannya.(Sibagariang, 2016)

*Representational State Transfer (REST)* adalah arsitektur perangkat lunak yang digunakan untuk pendistribusian sistem layanan *web* atau *web services*, sedangkan *REST* adalah aplikasi yang berfungsi untuk melakukan komunikasi dengan layanan *web* menggunakan *HTTP* untuk melakukan perintah *GET*, *PUT*, *POST*, dan *DELETE* data.

### **2.1.8. Logic Activity**

*Activity* adalah komponen utama dalam pengembangan aplikasi *Android*. Setiap layar yang terlihat oleh pengguna dalam aplikasi *Android* yang direpresentasikan oleh satu atau lebih *Activity*. *Activity* mengatur tampilan *UI*, logika interaksi, dan mengelola alur logika aplikasi pada suatu tahap. Logika *Activity* melibatkan implementasi alur kerja dan interaksi dalam aplikasi. Ini mencakup pengelolaan tampilan, pengolahan *input* pengguna, pemrosesan data, dan navigasi antar layar.

## **2.2. Tinjauan Pustaka**

Perangkat lunak aplikasi banyak digunakan di kalangan masyarakat, sehingga banyak mahasiswa meneliti terkait perangkat lunak aplikasi dalam berbagai versi, terutama yang berbasis *Android*.

Dalam jurnal ilmiah dari (Irvansyah et al, 2020) dengan judul “Aplikasi Pemesanan Jasa Cukur Rambut Berbasis *Android*” menjelaskan bahwa pemanfaatan teknologi informasi terkini dapat digunakan untuk membangun sistem yang dapat menunjang dalam pemesanan jasa cukur rambut agar lebih mudah. Perancangan sistem berbasis *Android* dirasa perlu untuk mempermudah pemesanan jasa cukur rambut. Dalam penelitian ini dikembangkan sebuah aplikasi pemesanan jasa cukur rambut berbasis *Android* untuk mengurangi antrian pada penyedia jasa cukur rambut. Berdasarkan hasil pengujian sistem menggunakan pengujian ISO 25010 mendapatkan nilai sangat baik atau sistem dapat diterima dan dijalankan dengan baik. Sehingga dapat di simpulkan perangkat lunak aplikasi berbasis *Android* dapat menjangkau masyarakat dalam hal teknologi.

Kemudian dari penelitian sebelumnya dari (Lian Min et al, 2020) dengan judul “Evaluasi Penggunaan Manual Dan *Automated Software Testing* Pada Pelaksanaan *End-To-End Testing*” menjelaskan masa *development cycle* dari sebuah *Software* yang semakin singkat, maka diperlukan pengerjaan *testing* (secara manual maupun *automated testing*) untuk melihat bagaimana karakteristik dan teknis pengujian baik secara manual atau pun secara otomatis. Maka dengan adanya *testing Software* maka perangkat lunak aplikasi dapat berjalan sesuai kebutuhan.

Pada jurnal ilmiah (Alviansyah & Ramadhani, 2021) dengan judul “*Implementasi Dynamic Application Security Testing* pada Aplikasi Berbasis *Android*” menyatakan *DAST* mampu melakukan pengujian secara otomatis pada setiap *Activity* yang ada dibuktikan dengan hasil kerentanan pada *Activity Insecure Data Storage* yang memiliki celah *XSS* dan *Database Breach*. Maka dengan adanya *testing*/pengujian sangat dibutuhkan demi kenyamanan dan efektifitas penggunaan sebuah aplikasi.