# BAB II TINJAUAN PUSTAKAN

## 2.1 Penelitian Terdahulu

Berikut ini adalah refrensi yang diambil untuk menunjang penelitian ini

dari penelitian yang telah dilakukan sebelumnya.

### 2.1.1 Penelitian Pertama

Penelitian pertama berjudul “Prototipe Aplikasi Layanan Pengaduan

Masyarakat Berbsis Android dan Web Service” yang ditulis oleh Andi Jumardi, dan

Achmad Solichin dari Universitas Budi Luhur yang di publikasi pada tahun 2016.

Penelitian ini berdasarkan peningkatan limbah yang terjadi setiap tahun di Kota Makassar. Dalam aplikasi tersebut menggunakan Strees Tool Versi 8 dengan 100 user dengan waktu rata rata 819ms.

Kelebihan dari Penelitian ini adalah pengujian dilakukan oleh 100 user

secara seksama, sehingga didapatkan hasil yang akurat. Namun pada penelitian ini tidak dijelaskan *web service* yang digunakan.

### 2.1.2 Penelitian Kedua

Penelitian kedua berjudul “Perancangan *User Experience* Pelaporan Keluhan Masyarakat Menggunakan Metode *Human-Centered Design”* yang oleh Aryun Nadaa A., Herman Tolle, Hanifa Muslimah A. dari Universitas Brawijaya yang di publikasi pada tahun 2018. Penelitian ini merombak suatu *User Experience* (UX) dari aplikasi keluhan masyarakat. Dengan menggunakan UX yang telah ditawarkan oleh penelitian ini *usability* dari aplikasi tersebut meningkat hingga

10

86%. Peningkatan persentasi yang terjadi oleh penelitian ini dapat dengan mudah di implementasikan kepada aplikasi. Akan tetapi rancangan tersebut hanya diberikan bagian *homepage.* Akan lebih baik jika penelitian tersebut memberikan detail pada *page-page* lain untuk ditinjau lebih lanjut.

### 2.1.3 Penelitian Ketiga

Penelitian ketiga berjudul “Pengembangan Sistem Informasi Pengaduan Masyarakat Kabupaten Jepara Berbasis Web” oleh Siti Rohmatun, Ida Widiastuti,

Muhammad Khosyi’in dari Universitas Islam Sultan Agung yang di publikasi tahun 2017. Penelitian ini memberikan sarana masyarakat kota Jepara untuk mengeluh tentang masalah di dalam kota maupun pemerintahan serta tanggapan yang akan diberikan oleh pemerintah kota tersebut.

Pada penelitian ini, aplikasi yang dirancang dapat menerima gambar

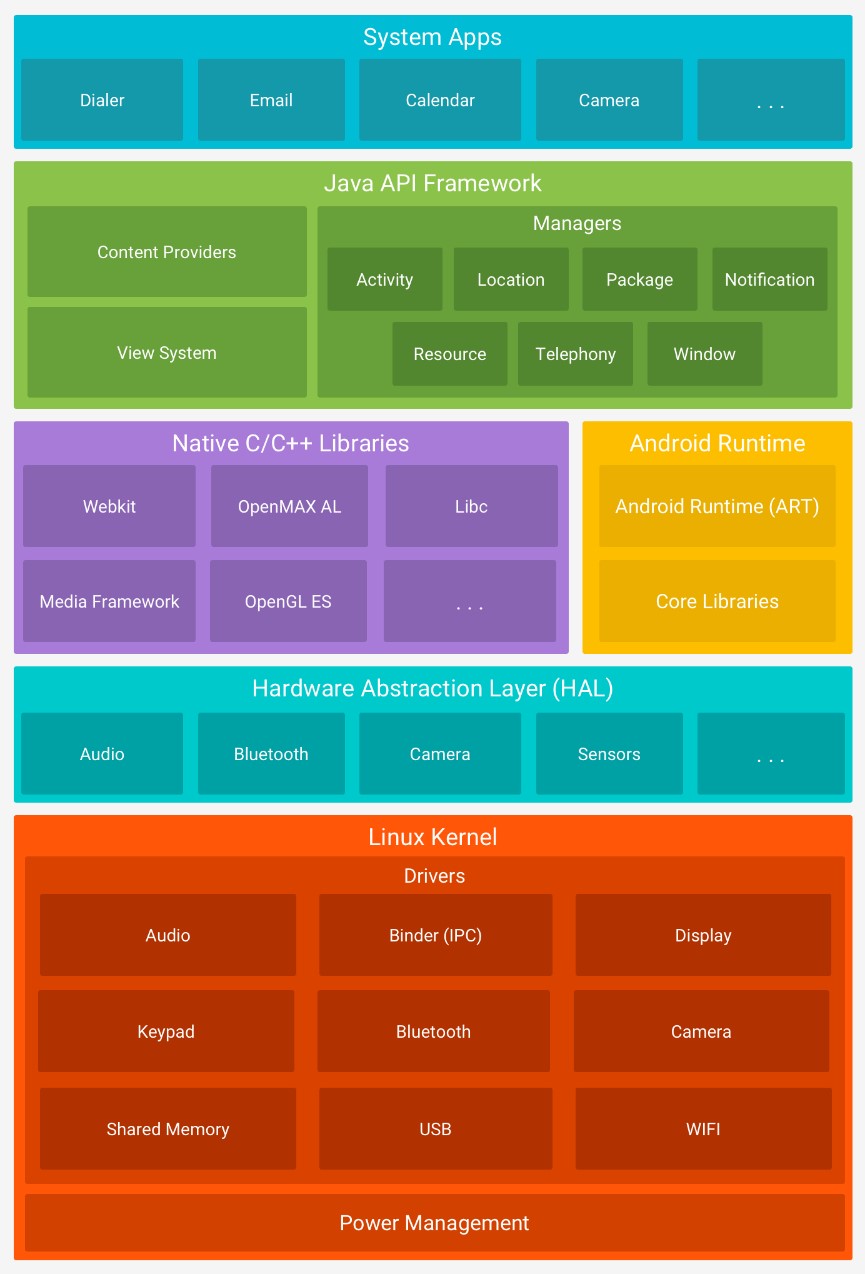
maupun video yang dikirim oleh penggunanya. Akan tetapi video tersebut berupa *link* yang di *upload* pada *web* lain. Sehingga jika web untuk *upload* video tersebut terjadi masalah, maka video tersebut tidak dapat dilihat oleh penerima laporan.

## 2.2 Teori Terkait

### 2.2.1 Android

Android merupakan sebuah perangkat yang berjalan pada sebuah sistem

operasi untuk *smartphone* yang berbasis linux (Arifianto, 2011). Diagram di bawah ini menunjukkan arsitektur dari platform Android.



**Gambar 2.1** Arsitektur Android

Tidak seperti iOS dan Windows, Android bersifat *open-source*, yang

berarti seorang *programmer* bisa mengubah dan memodifikasi sistem operasi dari GUI (*Graphical User Interface*) yang ada pada setiap telepon seluler. Maka dari itu telepon seluler yang menggunakan sistem operasi Android akan terlihat berbedabeda. Sampai saat ini terdapat 11 versi Android yang telah beredar dengan versi terbaru Android 11 dengan API level 30.

 **Android *Software Development Kit* (Android SDK)**

Android *Software Development Kit* (SDK) adalah *tool* API yang penting untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. (Safaat, 2011). Android adalah bagian dari perangkat lunak untuk *smartphone* yang mencakup sistem operasi, *middleware*, dan aplikasi utama yang di rilis oleh Google.

#### Arsitektur Android

Arsitektur android terdiri dari Linux Kernel, *Hardware Application*

*Layer* (HAL), *Library,* *Android Runtime,* Java API *Framework*, dan *System*

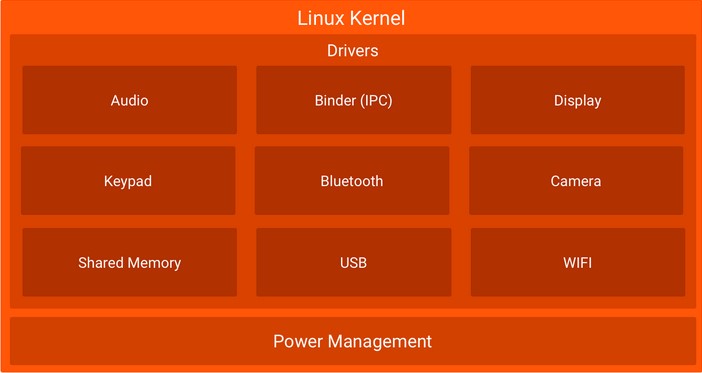
*Apps*.(*Arsitektur Platform*, n.d.)

##### 2.2.1.2.1 Linux Kernel

Dasar dari platform Android adalah kernel Linux. Kernel adalah lapisan

terdalam yang berada pada sebuah sistem operasi baik itu Linux maupun Android. Linux memiliki banyak turunan salah satu turunan tersebut adalah Android, maka

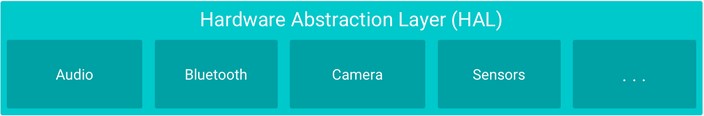
Android mempunyai kernel.



**Gambar 2.2** Linux Kernel

##### 2.2.1.2.2 Hardware Abstraction Layer (HAL)

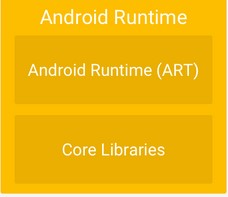
Hardware Abstraction Layer (HAL) meyediakan *interface* standar yang memaparkan kemampuan perangkat keras perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri dari beberapa *library*, masing-masing menerapkan *interface* untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Ketika API *framework* melakukan panggilan untuk mengakses *hardware*, sistem Android memuat *library* untuk komponen *hardware* tersebut.



**Gambar 2.3** Hardware Abstraction Layer

##### 2.2.1.2.3 Android Runtime

Terletak pada tingkatan yang sama dengan lapisan *Library* juga terdapat Lapisan Android Runtime dan juga sekumpulan *Library* Java yang dikhususkan untuk Android. Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang lebih baru, setiap aplikasi menjalankan prosesnya sendiri dengan tahap Android Runtime (ART).



**Gambar 2.4** Android Runtime

##### 2.2.1.2.4 Libraries

Library berisi instruksi untuk mengarahkan perangkat Android untuk menangani berbagai tipe data. Misalnya, *Media Framework Library merekam* perekam dari berbagai macam format *Video* dan *Audio*.

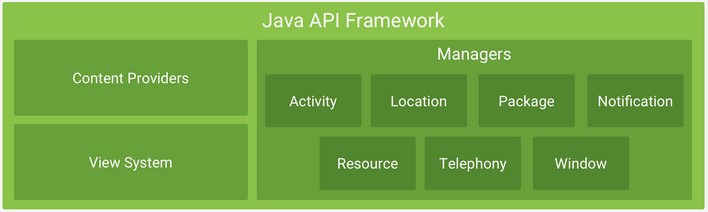


**Gambar 2.5** Libraries

##### 2.2.1.2.5 Java Framework

Lapisan ini berinteraksi langsung dengan aplikasi. Fungsi dasar dari perangkat dikelola pada lapisan ini, seperti manajemen *resource*, Manajemen

Panggilan, Manajemen Window dll.



**Gambar 2.6** Java Framework

##### 2.2.1.2.6 System Apps

Aplikasi sistem bertindak untuk pengguna dan menyediakan

fungsionalitas utama yang dapat diakses oleh pengembang dari aplikasi mereka

sendiri.



**Gambar 2.7** System Apps

### 2.2.2 GPS

GPS (*Global Positioning System*) adalah sistem navigasi berbasis satelit

yang menyediakan informasi tempat (koordinat) dan waktu dalam berbagai kondisi cuaca, dimana saja di diatas permukaan bumi, selama masih menerima sinyal GPS yang dikirimkan dari satelit (Marjuki, 2015)

### 2.2.3 Framework

*Framework* adalah suatu kumpulan kode yang berupa pustaka (*library*)

dan alat (*tool*) yang digabungkan atau diorganisasikan sedemikian rupa sehingga menjadi sebuah *framework* untuk memudahkan dan mempercepat proses pengembangan aplikasi *web* (Raharjo, 2011b). Menggunakan *framework* membangun sebuah aplikasi menjadi lebih mudah.

Metode yang banyak digunakan dalam *framework* Adalah MVC. MVC

(*Model-View-Controller*) adalah metode yang memisahkan data (*model*), tampilan (*view*), dan proses (*controller*). Dengan begitu proses yang akan dibuat akan terpisah pisah sehingga memudahkan pengguna untuk membagi pekerjaan dalam mengembangkan suatu aplikasi.

#### Laravel

Laravel adalah sebuah *Framework* pengembangan *website* berbasis MVP yang dibuat dalam PHP yang didesain untuk meningkatkan kualitas *software* dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja membuat aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. (*Installation - Laravel - The PHP*

*Framework For Web Artisans*, n.d.)

##### 2.2.3.1.1 Lumen

Lumen adalah Micro Framework yang dibuat oleh *developer* Laravel untuk memenuhi kebutuhan *developer* yang ingin membangun aplikasi dalam *size* yang lebih kecil dari Laravel. Karena banyak library yang dihilangkan dalam bundle source code, *framework* Lumen dapat digunakan untuk memudahkanpengembangan REST API.(Fauzi, 2017)

### 2.2.4 Database

*Database* ataubasis data merupakan sebuah susunan dan kumpulan data operasional yang komprehensif dari suatu organisasi serta perusahaan yangdiorganisir atau dikelola lalu disimpan secaraterintegrasi dengan memakai metode tertentu menggunakan komputer sehingga bisa menyediakan informasi optimal yang dibutuhkan oleh penggunanya(Marlinda, 2004). *Database* digunakan untuk mengklasifikasikan data dan memudahkan dalam identifikasi data. *Database* akan menampilkan data yang diminta sesuai oleh pengguna.

#### MySQL

MySQL merupakan salah RDBMS (*server database*) yang berbasis SQL(*Structured Query* Language). MySQL dapat mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat diakses oleh banyak *user*

*(Raharjo, 2011)*

### 2.2.5 API

API atau (*Application Programming Interface*) adalah sebuah aplikasi perangkat lunak yang berfunsi agar pengembang untuk menggabungkan dan menghubungkan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung satu sama lain. Tujuan dari penggunaan API adalah mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah metode yang terpisah sehingga *developer* tidak perlu lagi membuat fungsi yang serupa (Intermedia, 2020).

### 2.2.6 Java

Java dikembangkan Sun Microsystem. Java adalah nama seperangkat teknologi untuk membuat dan menjalankan *softare* pada komputer *standalone* ataupun pada lingkungan *network*.Java 2 adalah generasi kedua dari *java platform*

(Rosa, 2016).

Java adalah bahasa berorientasi objek untuk mengembangkan sebuah

aplikasi mandiri, aplikasi berbasis internet, aplikasi untuk perangkat pintar yang dapat berkomunikasi melalui jaringan internet/komunikasi.

#### Java Development Kit (JDK)

Bahasa Pemrograman java memiliki *Java Development Kit* atau

disingkat JDK. JDK sendiri merupakan sebuah alat yang digunakan untuk membuat program dalam bahasa pemrograman Java.

### 2.2.7 Pengujian Black Box

Pengujian Black Box adalah suatu metode pengujian dimana penguji

hanya fokus pada apa yang seharusnya dilakukan oleh sistem. Ketika sebah sistem dapat memproses data dan membuat hasil yang sesuai dengan apa yang diharapkan maka Suatu Uji dapat dikatakan berhasil. Dalam penggunaannya, pada metode black box tester tidak perlu mengetahui bagaimana struktur dan desain data yang ada di dalam sistem. Penguji hanya melihat apakah sistem yang di uji terdapat *error* atau tidak.

Pengujian Black Box befokus pada spesifikasi fungsional dari perangkat

lunak, kumpulan kondisi input dan melakukan eksekusi pada fungsional program program. (Mustaqbal, 2015).