BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Android

Android adalah sistem operasi *mobile* yang dikembangkan oleh *Google* berdasarkan hasil modifikasi dari versi *kernel Linux* dan dirancang khusus untuk perangkat layar sentuh seperti *smartphone* dan *tablet* (Open Handset Alliance, 2012). Pada perkembangannya, *android* tidak hanya merambah perangkat *mobile* saja tetapi *google* juga mengembangkan *Android TV* untuk televisi, *Android Auto* untuk mobil, dan *Android Wear* untuk jam tangan.

Antarmuka *android* umumnya berupa manipulasi langsung menggunakan gerakan sentuh seperti menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. *Android* juga mendukung perangkat internal seperti *accelerometers*, *gyroscopes* and *proximity sensors* yang digunakan beberapa aplikasi untuk merespons tindakan pengguna, seperti menyesuaikan layar *portrait* menjadi *landscape* saat perangkat diputar.

Aplikasi atau yang biasanya disebut *apps* pada *android* berguna untuk menambahkan fungsionalitas perangkat. Aplikasi *android* biasanya ditulis dalam bahasa pemrograman *Java* dengan menggunakan *Android Software Development Kit* (SDK) (Android Developers, 2017). SDK mencakup seperangkat alat pengembangan yang komprehensif, termasuk *debugger*, pustaka perangkat lunak, emulator *handset*, dokumentasi, sampel kode, dan tutorial. Perangkat tersebut mengelola kode bersama data dan *file* menjadi sebuah *Android PacKage* (APK), yang berupa *file* arsip dengan akhiran *.apk.* Satu *file* APK berisi semua materi aplikasi *Android* dan merupakan *file* yang digunakan perangkat *Android* untuk memasang aplikasi (Android Developers, 2017).

2.1.1 Android Studio

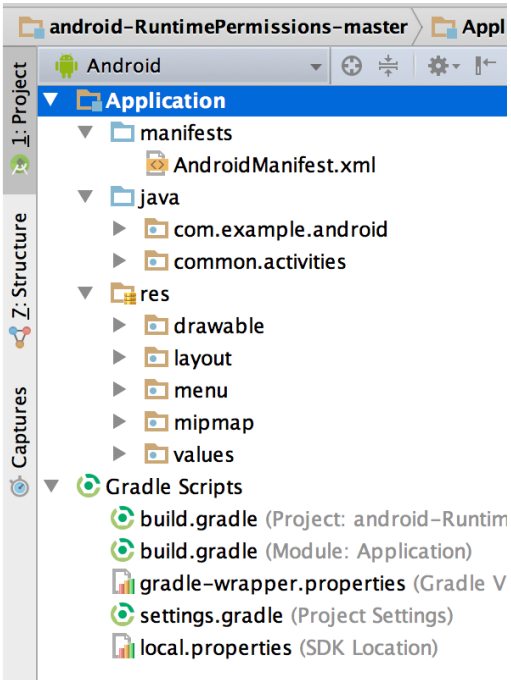
Android Studio adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi *Android* yang dibangung diatas perangkat lunak *JetBrains IntelliJ*

IDEA dan didesain khusus untuk pengembangan *Android*. IDE ini merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi *Android* (Forum-id, 2018).

Android Studio memiliki fitur - fitur sebagai berikut:

1. *Build system* berbasis *Gradle* yang fleksibel.
2. Emulator yang kaya fitur.
3. Mampu mengembangkan semua perangkat *Android*.
4. *Instant Run* untuk memberikan perubahan ke aplikasi yang sedang berjalan tanpa membuat APK baru.
5. *Template* kode dan integrasi *GitHub* yang membantu dalam membuat aplikasi dengan fitur biasa dan mengimpor sampel kode.
6. Alat pengujian dan *frameworks* yang ekstensif.
7. *Lint tools* untuk menganalisis kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan *Native Development Kit* (NDK).
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine.*

Setiap *project* di dalam *Android* Studio memiliki satu atau beberapa modul seperti modul aplikasi, modul *library*, dan modul *google cloud* yang berisi *source code files* dan *resource files.* Tampilan *project* di dalam *Android* Studio dapat dilihat pada Gambar 2-1.



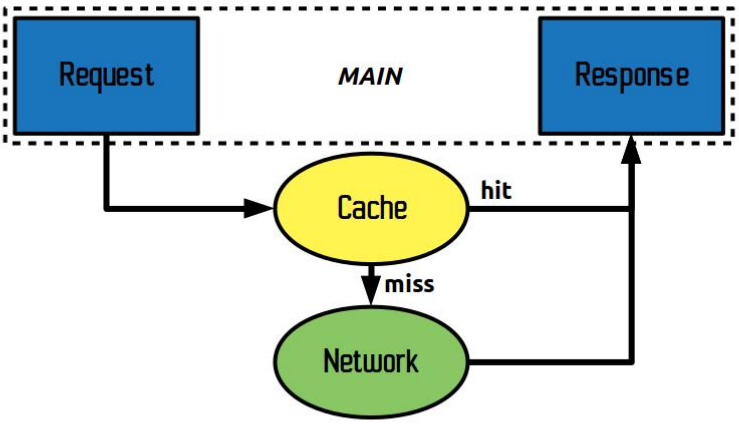
Gambar 2-1. *Project files* dalam tampilan *Android view* (*Android Developers*, 2018)

Pada Gambar 2-1 *project files* ditampilkan secara *Android view* yang disusun berdasarkan modul. Seluruh *build files* dapat dilihat pada bagian *Gradle Scripts* dan modul aplikasi yang terdiri dari folder berikut:

1. *Manifests* yang berisi *file AndroidManifest.xml.*
2. *Java* yang berisi *file Java source code,* termasuk *JUnit test code*.
3. *Res* yang berisi semua *non-code resources,* seperti XML *layout,* UI *strings,* dan gambar bitmap.

2.1.2 Volley

Volley adalah HTTP *library* yang membuat aktifitas jaringan aplikasi *android* menjadi lebih mudah dan cepat. *Volley* mampu menyajikan data yang terstruktur dalam bentuk *string*, gambar, dan JSON. *Volley* menyimpan semua *response* ke dalam *memory* ketika sedang beroperasi sehingga *volley* tidak cocok untuk melakukan proses *streaming* atau *download* data yang besar. Sebagai gambaran bagaimana cara kerja volley bisa dilihat pada Gambar 2-2.



Gambar 2-2. Arsitektur volley

Terdapat tiga bagian pada proses volley, yaitu *main*, *cache*, dan *network*. Pada bagian *main* hanya memberikan *request* dan menerima *response* saja. Ketika *request* telah diberikan, *volley* akan memeriksa apakah *request* dapat dilayani oleh *cache*. Apabila *cache* bisa melayani *request* tersebut, *response* yang telah tersimpan pada cache akan diberikan. Jika *cache* tidak bisa melayani *request*, maka *request* akan diteruskan ke *network*. Ketika *network* menerima *request*, bagian *network* akan membuat permintaan pada *HTTP* lalu tanggapan dari *HTTP* akan diteruskan ke *response* pada bagian *main* dan memasukan *response* tersebut ke dalam *cache*.

Salah satu cara untuk menambahkan *volley library* ke dalam *project* adalah dengan menambahkan *dependencies* di dalam *build.gradle* (modul aplikasi) sebagai berikut:

```
dependencies {
  ...
  implementation 'com.android.volley:volley:1.1.1'
}
```

2.2 XAMPP

XAMPP adalah *open source software* yang berisi *Apache* server, basis data *MySQL*, dan penerjemah skrip *PHP* dan *Perl*. *Software* ini berfungsi sebagai *local server* yang biasanya digunakan untuk pengujian situs *web* sebelum situs tersebut diunggah ke *live server* karena *XAMPP* menggunakan komponen yang sama dengan server pada umumnya. Ada beberapa modul yang juga terdapat pada *XAMPP* seperti *OpenSSL*, *phpMyAdmin*, *MediaWiki*, *Joomla*, dan *WordPress*.

2.2.1 Apache Web Server

Apache Web Server adalah *open-source web* server yang memfasilitasi satu atau beberapa situs *web* agar dapat diakses melalui *web browser*. *Apache Web Server* dikembangkan dan dikelola oleh komunitas pengembang *software* dalam pengawasan *Apache Software Foundation*.

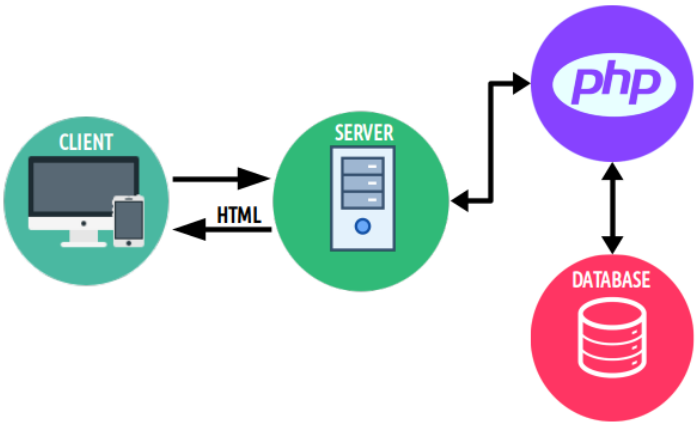
Apache Web Server mendukung beberapa bahasa pemrograman *server-side* seperti *Perl*, *Python*, *Tcl* dan *PHP*. Fitur lain yang juga diunggulkan dalam *Apache Web Server* adalah mekanisme autentikasi dan mendukung akses basis data. Fitur – fitur tersebut dimanfaatkan oleh perusahaan *web hosting* untuk menyediakan *virtual private server* (VPS) atau *shared hosting* untuk para penggunanya.

2.2.2 PHP

PHP (*Hipertext Preprocessor*) adalah bahasa skrip *server-side* yang dirancang untuk pengembangan *web* dan pengembangan perangkat lunak. Kode *php* dapat disematkan pada kode *html*, atau dapat digabungkan dengan berbagai *web template*, *web content management*, dan *web frameworks*. *PHP* biasanya diproses dengan *php interpreter* yang diterapkan sebagai modul di dalam *web server* atau sebagai *Common Gateway Interface* (CGI) yang dapat dieksekusi. *PHP* juga dapat dijalankan dengan *Command-Line Interface* (CLI) dan dapat digunakan untuk menerapkan aplikasi grafis tersendiri.

PHP pada dasarnya adalah sistem internet yang terbuka dengan modul *build-in* untuk mengakses server *File Transfer Protocol* (FTP) dan server basis

data seperi *PostgreSQL*, *MySQL*, *Microsoft SQL*, dan *SQLite*. *PHP* memberi keleluasaan kepada pengembang untuk membuat *extensions* dalam bahasa pemrograman C untuk menambahkan fungsionalitas bahasa *php*. Banyak *extensions* yang telah dibuat untuk menambahkan dukungan terhadap *Windows API*, manajemen proses pada sistem operasi *Unix-like*, *multibyte strings* (*Unicode*), *cURL*, dan beberapa *compression formats* yang populer. Fitur lain *php* yang didapatkan melalui *extensions* adalah integrasi dengan *Internet Relay Chat* (IRC), konten *Adobe Flash*, *PHP Data Object* (PDO) sebagai lapisan abstraksi yang digunakan untuk mengakses basis data, dan *speech synthesis*.



Gambar 2-3. Cara kerja *server-side scripting*

Dapat dilihat pada Gambar 2-3 *php* pada umumnya berjalan pada *web* server. Kode *php* dalam sebuah *file* dijalankan oleh *php runtime* yang biasanya untuk membuat konten *web page* atau gambar menjadi dinamis pada situs *web*. *PHP* dapat dimanfaatkan di sebagian besar *web* server, sistem operasi, *platforms*, dan *Relational DataBase Management Systems* (RDBMS).

Peran utama *php* adalah sebagai filter, yaitu mengambil masukan dari *file* berupa teks atau instruksi *php* dan menghasilkan data yang lain. Pada umumnya hasil data bisa berupa *HTML*, *JSON*, *XML*, dan *binary data* seperti gambar dan suara.

2.3 PostgreSQL

PostgreSQL adalah sebuah O*bject Relational Database Management System* (ORDBMS) yang mampu menangani *small single-machine applications* hingga *large Internet-facing applications* (*data warehousing*) dengan banyak pengguna yang bersamaan. Di dalam *macOS Server*, *PostgreSQL* adalah basis data bawaan tetapi juga tersedia untuk *Microsoft Windows* dan *Linux*.

2.4 System Development Life Cycle (SDLC)

SDLC adalah suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap dalam pembuatan sebuah software (Fatwa 2007). Selain itu, SDLC juga penting untuk proses maintenance software itu sendiri.

Model SDLC yang dipakai dalam pengembangan aplikasi adalah model Waterfall. Menurut (Sommerville 2011), waterfall model adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. Waterfall Model atau Classic Life Cycle merupakan model yang paling banyak dipakai dalam Software Engineering (SE). Disebut waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.