**EKSPLORASI JSON ADAPTER UNTUK RETROFIT PADA PEMBANGUNAN APLIKASI BERBASIS ANDROID**

**(Studi Kasus Publikasi Informasi dan Kegiatan di Dinasti Motekar Academy)**

**TUGAS AKHIR**

Disusun untuk syarat kelulusan Seminar Kualifikasi Penelitian

di Program Studi Teknik Informatika, Universitas Pasundan Bandung

oleh:

Adhitya Sidiq Permana

nrp. 15.304.0049



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PASUNDAN BANDUNG**

**NOVEMBER 2020**

# LEMBAR PENGESAHAN

# LAPORAN TUGAS AKHIR

Telah diujikan dan dipertahankan dalam Sidang Sarjana Program Studi Teknik Informatika Universitas Pasundan Bandung, pada hari dan tanggal sidang sesuai berita acara sidang, tugas akhir dari:

Nama : Adhitya Sidiq Permana

Nrp : 15.304.0049

Dengan judul :

“**EKSPLORASI JSON ADAPTER UNTUK RETROFIT PADA PEMBANGUNAN APLIKASI BERBASIS ANDROID**

**(Studi Kasus Publikasi Informasi dan Kegiatan di Dinasti Motekar Academy)**

Menyetujui,

Bandung,

Pembimbing Utama

(Wanda Gusdya, ST., MT.)

# LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa :

1. Tugas akhir ini adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya.
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing.
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmiah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini.
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung.

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sangsi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya.

|  |
| --- |
| Bandung, \_\_\_\_\_\_\_\_\_\_\_ |
| Yang membuat pernyataan |
|  |
|  |
|  |
|  |
|  |
| **(Adhitya Sidiq Permana)** |
| NRP. 15.304.0049 |

# ABSTRAK

# ABSTRACT

# KATA PENGANTAR

Ucapan dan rasa syukur penulis layangkan ke hadirat Ilahi Robbi, yang telah berkenan menguatkan penulis untuk membuat Laporan Tugas Akhir dengan judul “Eksplorasi JSON Adapter untuk Retrofit pada Pembangunan Pplikasi berbasis Android (Studi Kasus Publikasi Informasi dan Kegiatan di Dinasti Motekar Academy)”

Adapun penulisan laporan ini bertujuan untuk memenuhi salah syarat kelulusan program stuidi strata 1 di Program Studi Teknik Informatika Universitas Pasundan.

Penulis menyadari laporan ini dapat terwujud berkat bantuan dan dorongan dari berbagai pihak. Maka pada kesempatan ini penulis sampaikan terima kasih yang sebesar-besarnya atas segala bantuan yang penulis terima baik secara moril maupun materil, sehingga penulis dapat menyelesaikan laporan ini kepada :

1. Kepada Orang Tua tersayang, dan keluarga yang selalu memberikan motivasi serta do’anya dalam pembuatan tugas akhir ini.
2. Kepada pembimbing utama, Bapak Wanda Gusdya, ST., MT
3. Koordinator Tugas Akhir dan Ketua Kelompok Keilmuan serta seluruh civitas akademika Teknik Informatika di UNIVERSITAS PASUNDAN BANDUNG, yang telah memberikan bekal ilmu selama penulis menimba ilmu.
4. Kepada teman-teman seperjuangan Universitas Pasundan Bandung yang tidak bisa semua penulis sebutkan.

Tiada gading yang tak retak, tiada gelombang tanpa ombak, segala kesalahan merupakan kelemahan dan kekurangan penulis. oleh karena itu, penulis harapkan kritik dan saran dari semua pihak demi perbaikan di masa yang akan datang.

Akhir kata, semoga penulisan laporan ini dapat bermanfaat bagi penulis dan bagi perkembangan ilmu Teknologi dimasa yang akan datang.

Bandung, \_\_\_\_\_\_\_\_\_\_\_

Penulis

# DAFTAR ISI

[ABSTRAK i](#_Toc55641795)

[ABSTRACT ii](#_Toc55641796)

[KATA PENGANTAR iii](#_Toc55641797)

[DAFTAR ISI iv](#_Toc55641798)

[DAFTAR TABEL v](#_Toc55641799)

[DAFTAR GAMBAR vi](#_Toc55641800)

[BAB 1](#_Toc55641801) [PENDAHULUAN 1-1](#_Toc55641802)

[1.1 Latar Belakang 1-1](#_Toc55641803)

[1.2 Identifikasi Masalah 1-2](#_Toc55641804)

[1.3 Tujuan Tugas Akhir 1-2](#_Toc55641805)

[1.3 Lingkup Tugas Akhir 1-3](#_Toc55641806)

[1.4 Metodologi Tugas Akhir 1-3](#_Toc55641807)

[1.5 Sistematika Penulisan Tugas Akhir 1-5](#_Toc55641808)

[BAB 2](#_Toc55641809) [LANDASAN TEORI 2-1](#_Toc55641810)

[2.1. Aplikasi Mobile 2-1](#_Toc55641811)

[2.1.1. Perkembangan Aplikasi Mobile 2-1](#_Toc55641812)

[2.2. Sistem Operasi Android 2-2](#_Toc55641813)

[2.2.1. Aplikasi Android 2-3](#_Toc55641814)

[2.3. Web Service & REST API 2-3](#_Toc55641815)

[2.3.1. Metodologi REST API 2-4](#_Toc55641816)

[2.4. Retrofit Library 2-4](#_Toc55641817)

[2.5. Gson & Moshi 2-4](#_Toc55641818)

[2.6. Konsep MVC 2-5](#_Toc55641819)

[2.7. Software Development Life Cycle – Waterfall Model 2-5](#_Toc55641820)

[2.7.1. Tahapan – tahapan Waterfall Model 2-6](#_Toc55641821)

[2.8. Penelitian Terdahulu Terkait dengan Topik Tugas Akhir 2-7](#_Toc55641822)

[DAFTAR PUSTAKA](#_Toc55641823)

# DAFTAR TABEL

[Tabel 2‑1. Daftar Penelitian Terdahulu 2-7](#_Toc55245479)

# DAFTAR GAMBAR

[Gambar 1‑1. Metodologi Tugas Akhir 1-4](#_Toc55641719)

[Gambar 2‑1. Pertumbuhan Aplikasi Mobile di setiap kategori di AS tahun 2009-2010 2-2](#_Toc55641720)

[Gambar 2‑2. Category aplikasi yang sering digunakan 2-2](#_Toc55641721)

[Gambar 2‑3. Web Service Flow 2-4](#_Toc55641722)

[Gambar 2‑4. Tahap pengembangan perangkat lunak pada waterfall model 2-6](#_Toc55641723)

# BAB 1

# PENDAHULUAN

Bab ini memberikan penjelasan umum mengenai tugas akhir yang penulis lakukan. Penjelasan tersebut meliputi latar belakang masalah, identifikasi masalah, tujuan, ruang lingkup, metodologi penyelesaian tugas akhir, dan sistematika penulisan.

## 1.1 Latar Belakang

Selama lima tahun terakhir, perangkat mobile, seperti smartphone, dan tablet telah jauh lebih populer daripada perangkat desk-based tradisional seperti komputer pribadi dan laptop. Sejak iOS telah dirilis pada iPhone pada 2007, dan Android telah dirilis pada berbagai smartphone dan tablet pada tahun 2008, Sistem Operasi paling populer yang berjalan pada perangkat komputasi telah menjadi sistem operasi mobile. Dengan demikian, semakin banyak aplikasi yang berjalan pada Sistem Operasi mobile daripada Sistem Operasi desktop. [LIM14]

Pembangunan aplikasi berbasis android sering kali dilakukan setelah adanya sebuah website yang menampung data yang dibutuhkan. Sehingga dalam pembangunannya data yang dimuat selalu mengacu pada sebuah website. Dalam hal ini jika aplikasi dibuat dengan membuat skema database dari awal maka membuat pekerjaan bertambah juga kurang efektif penggunaannya.

Web Service adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi mesin-ke-mesin yang interoperable melalui jaringan. Memiliki antarmuka yang dijelaskan dalam format yang dapat diproses mesin (khususnya WSDL). Sistem lain berinteraksi dengan layanan Web dengan cara yang ditentukan oleh deskripsinya menggunakan pesan SOAP, biasanya disampaikan menggunakan HTTP dengan serialisasi XML bersama dengan standar lain yang berhubungan dengan Web. [W3C04]

REST Web adalah subset dari WWW (berdasarkan HTTP) di mana agen menyediakan semantik antarmuka yang seragam - pada dasarnya membuat, mengambil, memperbarui dan menghapus - bukan antarmuka yang sewenang-wenang atau khusus aplikasi, dan memanipulasi sumber daya hanya dengan pertukaran dari representasi. Interaksi REST adalah "tanpa kewarganegaraan" dalam arti bahwa makna pesan tidak tergantung pada keadaan percakapan. [W3C04]

Pada arsitektur REST, REST server menyediakan resources yang direpresentasikan dalam bentuk format teks, JSON atau XML. JSON singkatan dari JavaScript Object adalah suatu format ringkas pertukaran data yang sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi dan dapat direpresentasikan oleh berbagai bahsa pemrograman. Resource yang berbentuk format JSON yang disediakan REST server dapat dimanfaatkan oleh aplikasi android dengan library Retrofit.

Retrofit adalah REST Client Library (Helper Library) yang digunakan di Android dan Java untuk membuat permintaan HTTP dan juga untuk memproses respons HTTP dari REST API yang dibuat oleh Square, Retrofit juga dapat menggunakan retrofit untuk menerima struktur data selain JSON, misalnya SimpleXML dan Jackson. [OSA17]

Memparsing Data ke bentuk JSON dalam pembangunan aplikasi berbasis android, sudah banyak JSON adapter yang dapat membantu memparsing data seperti Moshi dan Google Gson.

Gson adalah Java library yang dapat digunakan untuk mengubah Java Objects menjadi representasi JSON. Itu juga dapat digunakan untuk mengonversi string JSON ke objek Java yang setara. Gson dapat bekerja dengan objek Java yang arbitrer termasuk objek yang sudah ada sebelumnya yang tidak Anda miliki kode sumbernya. [GOO08]

Moshi adalah Modern JSON Library untuk Android dan Java. Dan itu bisa dengan mudah membuat serialisasi objek Java seperti JSON [SQU15]

Perbandingan Kinerja Gson dan Moshi perlu diketahui agar kedepannya lebih dapat dipahami kinerja mana yang lebih cepat, efisien dan efektif dalam memparsing data ke dalam bentuk JSON yang nantinya akan digunakan dalam membangun aplikasi berbasis android menggunakan Retrofit yang akan digunakan untuk membantu upaya menyebarkan informasi secara luas dan secara cepat juga dalam rangka mempublikasikan suatu informasi dan kegiatan yang ada di Dinasti Motekar Academy

Metode Pembangunan Aplikasi ini mengadopsi metode pembangunan perangkat lunak SDLC Waterfall Model, dengan menerapkan metode yang ada maka diharapkan aplikasi Berbasis Android ini dapat dibangun dengan sempurna dan dapat dimanfaatkan untuk perbandingan penggunaan Gson dan Moshi juga untuk menunjang publikasi informasi dan kegiatan di Dinasti Motekar secara cepat dan lebih menarik minat pembaca.

## 1.2 Identifikasi Masalah

Berdasarkan latarbelakang yang telah dipaparkan sebelumnya, maka permasalahan yang dimunculkan pada tugas akhir ini adalah :

1. Bagaimana Mengimplementasikan JSON Adapter Gson dan Moshi dalam Pembangunan Aplikasi Berbasis Android?
2. Bagaimana Cara membandingkan JSON Adapter Gson dan Moshi dalam Pembangunan Aplikasi Berbasis Android?
3. Bagaimana Hasil Perbandingan JSON adapter Gson dan Moshi yang digunakan dalam Pembangunan Aplikasi Berbasis Android?

## 1.3 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah :

1. Mengetahui cara mengimplementasikan JSON adapter Gson dan Moshi dalam pembangunan Aplikasi Berbasis Android
2. Mengetahui cara membandingkan JSON adapter Gson dan Moshi dalam pembangunan Aplikasi Berbasis Android.
3. Menegtahui Hasil perbandingan dari JSON adapter Gson dan Moshi yang digunakan dalam Pembangunan Aplikasi Berbasis Android.

## 1.3 Lingkup Tugas Akhir

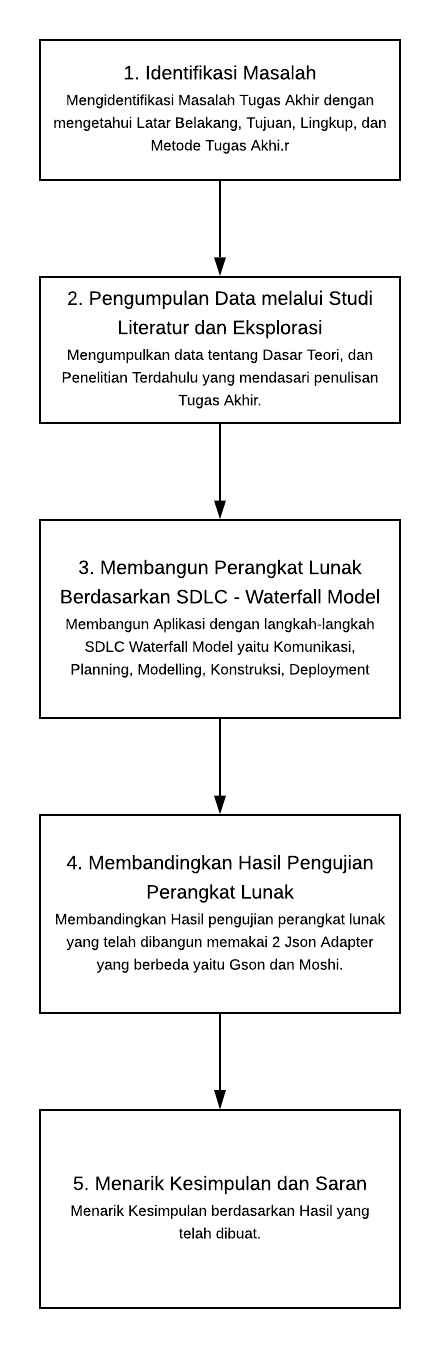
Penyelesaian Tugas Akhir dibatasi sebagai berikut :

1. Kasus yang dijadikan fokus penelitian adalah perbandingan dan pengimplementasian JSON adapter Aplikasi berbasis Android untuk Publikasi Informasi dan Kegiatan di Dinasti Motekar Academy.
2. Penelitian dibatasi hingga aplikasi telah selesai dibangun dan diketahui hasil Perbandingan JSON adapter Moshi dan Gson.
3. Metodologi yang digunakan adalah Software Development Life Cycle (SDLC) – Waterfall Model
4. Tidak melakukan pengukuran kepuasan pada aplikasi yang telah dibuat.

## 1.4 Metodologi Tugas Akhir

Berikut langkah penyelesaian tugas akhir :

1. Melakukan Identifikasi Masalah
2. Melakukan pengumpulan data melalui studi literatur dan Eksplorasi
3. Membangun Perangkat Lunak berdasarkan metode SDLC – Waterfall Model yang langkah – langkahnya sebagai berikut :
   1. Komunikasi
   2. Planning
   3. Pemodelan
   4. Kontruksi Perangkat Lunak
   5. Deployment
4. Membandingkan Hasil Pengujian Perangkat Lunak
5. Menarik kesimpulan dan saran



Gambar 1‑1. Metodologi Tugas Akhir

## 1.5 Sistematika Penulisan Tugas Akhir

Buku Tugas Akhir ditulis dengan mengikuti sistematika sebagaiberikut :

**Bab 1 : Pendahuluan**

Bab ini memberikan penjelasan umum mengenai tugas akhir yang penulis lakukan. Penjelasan tersebut meliputi latar belakang masalah, identifikasi masalah, tujuan, ruang lingkup, metodologi penyelesaian tugas akhir, dan sistematika penulisan.

**Bab 2 : Landasan Teori**

Dalam bab ini membahas beberapa teori yang mendasari penulisan dari tugas akhir ini. Teori umum yang digunakan meliputi teori pengembangan perangkat lunak, dan beberapa teori lainnya yang mendasari penyusunan tugas akhir ini.

**Bab 3 : Skema Penelitian**

Bab ini berisi kerangka pengerjaan tugas akhir, kerangka pemikiran teoritis,skema analisis yang dilakukan, penjelasan skema analisis yang telah dibuat, dan profil studi kasus.

**Bab 4 : Analisis dan Perancangan**

Bab ini menjelaskan proses awal pembangunan perangkat lunak yang terdiri dari analisis dan perancangan. Proses analisis terdiri dari pendefinisian perangkat lunak, pendefinisian kebutuhan yang harus dipenuhi perangkat lunak, dan analisis terhadap kebutuhan dengan menggunakan pemodelan spesifikasi kebutuhan dan pemodelan kebutuhan. Sedangkan proses perancangan perangkat lunak terdiri dari perancangan arsitektur, perancangan antarmuka, dan perancangan data.

**Bab 5 : Implementasi dan Pengujian**

Bab ini tahapan inti dan tahapan akhir dari pembangunan perangkat lunak, yaitu konstruksi perangkat lunak (coding) berdasarkan analisis dan perancangan yang telah dibuat serta pengujian dari hasil implementasi.

**Bab 6 : Kesimpulan dan Saran**

Bab ini menjelaskan mengenai kesimpulan yang didapatkan dari hasil penelitian yang dilakukan serta saran yang diberikan oleh penulis untuk pengembangan penelitian selanjutnya.

# BAB 2

# LANDASAN TEORI

Dalam bab ini membahas beberapa teori yang mendasari penulisan dari tugas akhir ini. Teori umum yang digunakan meliputi teori pengembangan perangkat lunak, dan beberapa teori lainnya yang mendasari penyusunan tugas akhir ini.

## 2.1. Aplikasi Mobile

Mobile Application terdiri dari perangkat lunak / kumpulan sebuah program yang berjalan di perangkat seluler dan bekerja dengan tugas tertentu untuk pengguna. Mobile Application merupakan hal yang baru dengan kemampuan developing segment yang cepat pada global information dan comunication technology. Mobile Application sangat mudah, user friendly, murah, dapat diunduh dan dijalankan di ponsel seperti apapun termasuk ponsel yang murah bahkan sampai entry level. Mobile application memiliki kegunaan yang luas seperti calling, messaging, browsing, chatting, social network communication, audio, video, games dll. [IJE10]

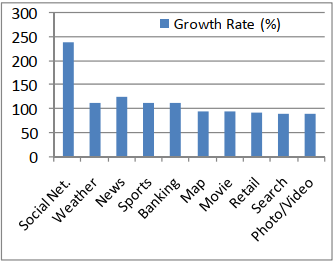
Sejumlah besar mobile application biasanya sudah diinstal sebelumnya di ponsel dan yang lainnya dapat diunduh oleh pengguna di internet dan menginstalnya di ponsel. Jumlah mobile application yang ada di market biasanya dibuat oleh berbagai macam mobile application developer, publisher dan provider. Dari segi teknis melihat berbagai mobile application yang dapat dijalankan di berbagai platform terkelola seperti iPhone, BlackBerry,Android, Symbian, windows; juga beberapa mesin virtualseperti Java / J2ME, BREW, Flash Light, Silverlight. [IJE10]

Menurut area aplikasi, ada perbedaan kategori Aplikasi Mobile :

1. Komunikasi: Browsing Internet, email IMklien, Jejaring Sosial
2. Permainan: Teka-Teki / Strategi, Kartu / Kasino,Aksi petualangan
3. Multimedia: Penampil Grafik / Gambar,Penampil presentasi, Pemutar Video, Audiopemain
4. Produktivitas: Kalender, Kalkulator, Buku Harian,Notepad / Memo / Pengolah Kata,Spreadsheet
5. Perjalanan: Panduan kota, Konverter mata uang,Penerjemah, GPS / Peta, Rencana Perjalanan / Jadwal, Cuaca
6. Utilitas: Manajer profil, Layar / Layar Idlepenghemat, Buku alamat, Manajer tugas, Panggilanmanajer, Manajer file [IJE10]

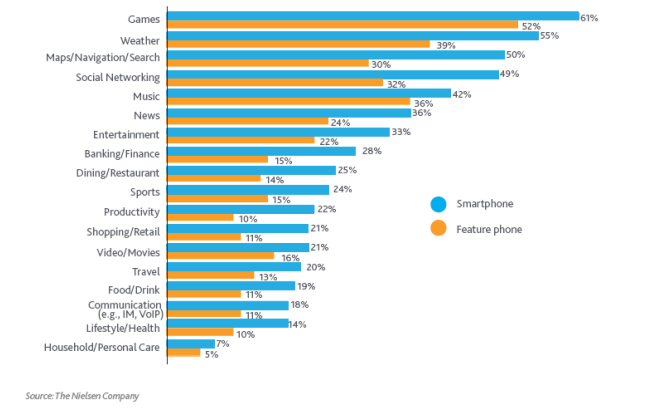
### 2.1.1. Perkembangan Aplikasi Mobile

Menurut sebuah studi oleh MobiLens, jumlah pengguna aplikasi seluler di AS tumbuh sebesar 28% antara April 2009 dan April 2010. Berikut adalah tabel pertumbuhan tingkat semua kategori aplikasi dalam satu tahun hingga April 2010. [IJE10]



Gambar 2‑1. Pertumbuhan Aplikasi Mobile di setiap kategori di AS tahun 2009-2010

Perusahaan Nielsen, merilis survei pada bulan September 2010 telah menerbitkan laporan mobile application yang sring di download. Dalam survei itu mereka menunjukkan beberapa data statistik penting dari aplikasi mobile 30hari.Menurut survei, gambar berikut mewakili tingkat penggunaan berbagai kategori aplikasi mobile. [IJE10]



Gambar 2‑2. Category aplikasi yang sering digunakan

Dari data-data diatas yang telah dirilis menunjukan bahwa pertumbuhan pengguna aplikasi mobile paling banyak dilakukan di social media dan kedua terbanyak untuk berita, ini menunjukan bahwa pengguna lebih sering mencari dan membagikan informasi. Bahkan dari data yang dirilis Nielsen penggunaan Smartphone untuk berbagai macam kegiatan sangat tinggi, daripada tipe-tipe perangkat seluler yang lainnya. Dan biasanya banyaknya smartphone di support oleh sistem Operasi Android.

## 2.2. Sistem Operasi Android

Android adalah platform perangkat lunak dan sistem operasi untuk perangkat mobile, dibuat dari kernel Linux, dan dikembangkan oleh Google dan kemudian dijadikan Open Source. membuat pengembang memungkinkan untuk menulis kode yang dikelola bahasa Java, mengontrol perangkat melalui pustaka Java yang dikembangkan Google. Android tersedia secara Open Source yang dapat diunduh secara gratis dan menyediakan berbagaimacam hal seperti sistem operasi, middleware dan aplikasi utama berbasis Linux dan Java. Google membeli pengembang Android pada tahun 2005, dan Android diresmikan pada tahun 2007. Google merilis kode Android sebagai open-source di bawah Lisensi Apache. Android memiliki banyak pengembang aplikasi di seluruh dunia. [INT15]

## 2.2.1. Aplikasi Android

Aplikasi, yang memperluas fungsionalitas perangkat, dibangun menggunakan Software Development kit (SDK) dan, seringkali, menggunakan bahasa pemrograman Java yang memiliki akses lengkap ke API Android. Java dapat digabungkan dengan C / C ++, bersama dengan pilihan runtime non-default memungkinkan dukungan C ++ yang lebih baik,bahasa pemogram Go juga didukung sejak versi 1.4, yang juga dapat digunakan secara eksklusif meskipun dengan API Android yang terbatas. SDK mencakup aset perangkat alat pengembangan yang komprehensif, termasuk debugger, software libraries, handset emulator based di QEMU, dokumentasi, sample code, dan tutorial. Awalnya, Google mendukung integrated development environment (IDE) dengan Eclipse menggunakan Android Development Tools (ADT) Plugin, namun pada bulan Desember 2014,Google merilis Android Studio, berbasis IntelliJIDEA, sebagai IDE utamanya untuk aplikasi pengembangan Android . Alat pengembangan lainnya tersedia,termasuk native development kit (NDK) untuk aplikasi atau ekstensi di C atau C ++, Google App Inventor, lingkungan visual untuk pemrogram pemula, dan berbagai cross platform mobile web applications frameworks. Pada Januari 2014, Google meluncurkan framework berbasis Apache Cordova untuk porting Aplikasi web Chrome HTML 5 ke Android, dibungkus dalam native application shell. [INT15]

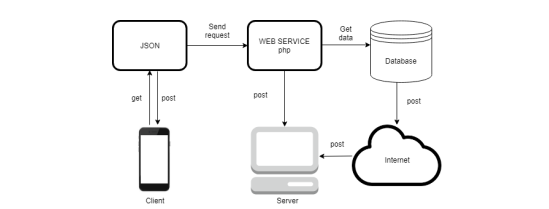
## 2.3. Web Service & REST API

Web Service adalah Standarisasi distribusi berbagai macam data dan proses dari berbagai macam aplikasi dan services yang ada di internet. Aplikasi di internet biasanya di bangun dengan berbagai bahasa pemograman yang berbeda tergantung sistem operasi dan perangkat. Pemanfaatan Web Service adalah untuk mengatasi masalah pengintegrasian data dan proses saat pertukaran informasi antar platform, serta tingkat keamanan data dapat ditangani dengan baik. Arsitektur yang menjadi dasar web service adalah REST dan SOAP

REST API didefinisikan sebagai alat arsitektur yang dirancang pada layanan web yang berfokus pada sumber daya sistemseperti mentransfer & meminta data menggunakan http. REST telah diterima secara luas oleh masyarakat karena simple digunakan dalam paradigma cloud seluler. Layanan web REST bekerja dengan mengidentifikasi URL lalu memodifikasinyadengan perintah GET, POST, PUT atau DELETE yang dapat direpresentasikan pada beberapa bahasa pemrograman seperti XML atau JSON. Berbagai aplikasi dibuat dengan memanfaatkan REST API seperti catatan kesehatan pasien, indeks kesejahteraan manusia badan perlindungan lingkungan AS(HWBI) yang mencirikan kesejahteraan manusia menggunakan data sosial, ekonomi dan ekologi, Smartmeter air yang secara otomatis dapat mengumpulkan air penggunaan terperinci, dan penambangan aturan asosiasi layanan yang digunakan untuk menemukan pola yang sering dari data besar. Beberapa penelitian menunjukkan bahwa SOAP based Service memang sangat berat untuk dijalankan pada perangkat seluler sehingga Restful Web Services lebih cocok untuk aplikasi mobile bahkan menunjukkan bahwa REST dengan JSON telah menunjukkan kinerja yang lebih baik dibandingkan dengan SOAP. [INT19]

### 2.3.1. Metodologi REST API

Web services dibangun dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai databasenya menyimpan data dalam aplikasi di server internet. REST API diperlukan untuk mengelola akses ke data diweb dan tindakan ke aplikasi seluler android. [INT19]



Gambar 2‑3. Web Service Flow

## 2.4. Retrofit Library

Library merupakan kumpulan resource nonvolatile yang digunakan oleh program computer dan sering digunakan untuk mengembangkan software. Library tersebut dapat berisikonfigurasi data, dokumentasi, data penunjang,maupun class. [CHE15]

Retrofit merupakan REST client library yang aman untuk android dan java. Dibuat oleh Square. Retrofit menyediakan cara yang aman untuk autentikasi dan interaksi dengan berbagai API lainnya. Sehingga memungkinkan pengiriman permintaan jaringan dengan OkHttp. Retrofit mengambil data JSON atau XML dari web API dan saat data diterima akan langsung diubah ke Plain Old Java Object (POJO). Sehingga harus ditentukan setiap class yang akan dipakai saat response diterima. [COD16]

Retrofit juga bekerja dengan REST API menggunakan implementasi java interface, yang dapat dihasilkan dengan bantuan RestAdapter. Implementasi dalam hal ini bertindak sebagai local instance dari layanan dan setiap panggilan sesuai dengan permintaan HTTP. [MAS15]

## 2.5. Gson & Moshi

Gson adalah Java library yang dapat digunakan untuk mengubah Java Objects menjadi representasi JSON. Itu juga dapat digunakan untuk mengonversi string JSON ke objek Java yang setara. Gson dapat bekerja dengan objek Java yang arbitrer termasuk objek yang sudah ada sebelumnya yang tidak Anda miliki kode sumbernya. [GOO08]

Moshi adalah Modern JSON Library untuk Android dan Java. Dan itu bisa dengan mudah membuat serialisasi objek Java seperti JSON [SQU15]

## 2.6. Konsep MVC

Model-View-Controller atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller). Dalam implementasi kebanyakan framework dalam aplikasi web adalah berbasis arsitektur MVC (Model-View-Control). [ROS11]. Definisi teknis dari arsitektur MVC dibagi menjadi  
tiga lapisan menurut Burbeck [BUR92] :

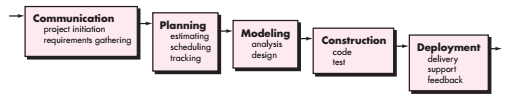
1. Model, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada  
   perubahan informasi. Hanya model yang mengandung data dan fungsi yang berhubungan  
   dengan pemrosesan data. Sebuah model meringkas lebih dari sekedar data dan fungsi yang  
   beroperasi di dalamnya. Pendekatan model yang digunakan untuk komputer model atau  
   abstraksi dari beberapa proses dunia nyata. Hal ini tidak hanya menangkap keadaan proses  
   atau sistem, tetapi bagaimana sistem bekerja. Sebagai contoh, programmer dapat menentukan  
   model yang menjembatani komputasi back-end dengan front-end GUI (graphical user  
   interface).
2. View, bertanggung jawab untuk pemetaan grafis ke sebuah perangkat. View biasanya memiliki  
   hubungan 1-1 dengan sebuah permukaan layar dan tahu bagaimana untuk membuatnya. View  
   melekat pada model dan me-render isinya ke permukaan layar. Selain itu, ketika model  
   berubah, view secara otomatis menggambar ulang bagian layar yang terkena perubahan untuk  
   menunjukkan perubahan tersebut. Terdapat kemungkinan beberapa view pada model yang  
   sama dan masing-masing view tersebut dapat me-render isi model untuk permukaan tampilan  
   yang berbeda.
3. Controller, menerima input dari pengguna dan menginstruksikan model dan view untuk  
   melakukan aksi berdasarkan masukan tersebut. Sehingga, controller bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi. Sebagai contoh, ketika pengguna mengklik tombol atau memilih item menu, controller bertanggung jawab untuk menentukan bagaimana aplikasi seharusnya merespon.

## 2.7. Software Development Life Cycle – Waterfall Model

Waterfall Modeladalah model yang menunjukkan sistematika, pendekatan sekuensial untuk  
pengembangan perangkat lunak yang dimulai dengan spesifikasi kebutuhan pelanggan dan  
berkembang melalui perencanaan, pemodelan, konstruksi dan deployment, yang berpuncak pada  
dukungan berkelanjutan dari perangkat lunak yang sudah selesai [PRE10]

### 2.7.1. Tahapan – tahapan Waterfall Model

Di dalam tahapan Waterfall Model terdapat lima tahapan, yaitu tahap communication, planning, tahap modeling, tahap construction, dan tahap deployment seperti yang digambarkan pada Gambar 2-4. [PRE10]



Gambar 2‑4. Tahap pengembangan perangkat lunak pada waterfall model

#### **2.7.1.1. Communication**

Sebelum persyaratan customer dapat dianalisis, dimodelkan, atau ditentukan, mereka harus  
dikumpulkan melalui aktivitas komunikasi. Customer memiliki masalah yang mungkin terjadi dapat  
menerima solusi berbasis komputer. Permintaan customer tersebut ditanggapi untuk memberikan  
bantuan. Tetapi saat komunikasi telah dimulai, jalan dari komunikasi untuk membentuk suatu  
pemahaman sering penuh dengan lubang (kesulitan). Komunikasi yang efektif (di antara rekan-rekan  
teknis, dengan customer dan stakeholder lainnya, dan dengan manajer proyek) adalah salah satu  
kegiatan yang paling menantang yang akan dihadapi. Aktivitas komunikasi membantu untuk  
menentukan tujuan dan sasaran Anda secara keseluruhan (subjek, tapi berubah seiring berjalannya  
waktu) [PRE10]

#### **2.7.1.2. Planning**

Memahami tujuan dan sasaran pada tahap komunikasi tidak sama dengan mendefinisikan  
rencana selanjutnya. Kegiatan perencanaan mencakup seperangkat praktik manajemen dan teknis yang  
memungkinkan tim perangkat lunak untuk menentukan peta jalan ketika suatu proses bergerak menuju  
tujuan strategis dan tujuan taktis. Tidak ada cara mudah untuk menentukan masalah teknis yang tak  
terduga yang akan dihadapi. Namun, tim perangkat lunak yang baik harus merencanakan  
pendekatannya. Pada banyak proyek, perencanaan berlebihan memakan banyak waktu dan tidak  
membuahkan hasil (terlalu banyak hal berubah) [PRE10]

#### **2.7.1.3. Modeling**

Memahami tujuan dan sasaran pada tahap komunikasi tidak sama dengan mendefinisikan  
rencana selanjutnya. Kegiatan perencanaan mencakup seperangkat praktik manajemen dan teknis yang  
memungkinkan tim perangkat lunak untuk menentukan peta jalan ketika suatu proses bergerak menuju  
tujuan strategis dan tujuan taktis. Tidak ada cara mudah untuk menentukan masalah teknis yang tak  
terduga yang akan dihadapi. Namun, tim perangkat lunak yang baik harus merencanakan  
pendekatannya. Pada banyak proyek, perencanaan berlebihan memakan banyak waktu dan tidak  
membuahkan hasil (terlalu banyak hal berubah) [PRE10]

#### **2.7.1.4. Construction**

Kegiatan konstruksi mencakup seperangkat tugas pengkodean dan pengujian yang mengarah  
ke perangkat lunak operasional yang siap dikirim ke pelanggan atau pengguna akhir. Dalam pekerjaan  
rekayasa perangkat lunak modern, pengkodean dapat dilakukan dengan : pembuatan langsung kode  
sumber bahasa pemrograman (misalnya, Java), pembuatan otomatis kode sumber menggunakan  
representasi antara desain seperti komponen yang akan dibangun atau generasi otomatis kode yang  
dapat dijalankan menggunakan "bahasa pemrograman generasi keempat" (misalnya, Visual C++).  
Fokus awal pengujian adalah pada tingkat komponen, yang sering disebut pengujian unit. [PRE10].

#### **2.7.1.5. Deployment**

Kegiatan deployment mencakup tiga tindakan yaitu : pengiriman, dukungan, dan umpan balik.  
Setiap siklus deployment kepada costomer dan pengguna akhir akan ada peningkatan perangkat lunak  
dalam hal operasional yang menyediakan fungsi dan fitur yang dapat digunakan. Setiap siklus  
dukungan menyediakan dokumentasi dan human assistance untuk semua fungsi dan fitur yang diperkenalkan dalam semua siklus penggunaan. Setiap siklus umpan balik menyediakan tim perangkat  
lunak dengan panduan perangkat lunak untuk dapat modifikasi fungsi, fitur, dan mengetahui  
pendekatan yang diambil untuk perbaikan berikutnya. [PRE10].

## 2.8. Penelitian Terdahulu Terkait dengan Topik Tugas Akhir

Ada beberapa penelitian terdahulu yang terkait dengan tugas akhir ini yang dapat dilihat pada Tabel 2-1.

Tabel 2‑1. Daftar Penelitian Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Judul** | **Peneliti** | **Hasil** |
| 1. | Rancang Bangun Aplikasi Pengelolaan Data Anggota Koordinator Olahraga Mahasiswa Universitas Pasundan Bandung | Eka Ardilah Febriyanti | Aplikasi Pengelolaan Data Anggota Koordinator Olahraga |
| 2. | Perangkat Lunak Untuk Layanan Warga Berbasis Android Di Lingkungan Rukun Warga Menggunakan Konsep M-Government (Studi Kasus Rw 10 Kelurahan Sukabungah Kota Bandung) | Andri Nurul Akbar | Aplikasi berbasis android di Lingkungan RW |
| 3. | Pembangunan Aplikasi Pakan Sapi Berbasis Android | Berta Erwin SLAM | Aplikasi android untuk membantu peternak menyusun ramsun dari bahan  pakan ternak |
| 4. | Eksplorasi Python Untuk Penyesuaian  Modul Perkuliahan Dan Praktikum  Di Teknik Informatika Unpas | Nur Sofia Malikhah | Topik dan pembelajaran dengan bahasa python |
| 5. | Eksplorasi Python Menggunakan Framework Django (Studi Kasus : Sistem Perpustakaan Teknik Informatika UNPAS) | Anggiyasti Yaktining Putri | Sistem Aplikasi Perpustakaan |

# DAFTAR PUSTAKA

[BUR92] Burbeck, S. (1992). *Applications Programming in Smalltalk-80 (TM) How to use Model-View-Controller (MVC). Vol. 2.5.* ParcPlace Systems, Inc.

[COD16] Codepath. (2016). *Consuming APIs with Retrofit*. Retrieved from Codepath: https://guides.codepath.com/android/Con

[GOO8] Google.Inc. (2008). *Github of Google Gson*. Retrieved from Github.com: https://github.com/google/gson

[CHE15] H., C. (2015). *Management of contextual information for data.*

[INT19] Ika Oktavia Suzanti, N. F. (2019). International Conference on Science and Technology. *REST API Implementation on Android Based Monitoring*.

[INT15] Kirthika.B, P. V. (2015). International Journal of Trend in Research and Development. *Android Operating System: A Review*, 260.

[KUC19] Kucherenko, S. M. (2019). *Learn Spring for Android Application Development.* Brimingham: Packt.

[LIM15] Lim, S. H. (2015). Experimental Comparison of Hybrid and Native Applications for Mobile System. *International Journal of Multimedia and Ubiquitous Engineering*.

[MAS15] Maskov. (2015). *Implementing REST Client for Android.* Helsinki: Metropolia University of Applied Sciences.

[IJE10] Md. Rashedul Islam, M. R. (2010). International Journal of Engineering & Technology IJET-IJENS Vol: 10 No: 06. *Mobile Application and Its Global Impact*, 104.

[OSA17] Osahon, G. (2017). *Consuming REST API using Retrofit Library in Android.* Retrieved from android.jlelse.eu: https://android.jlelse.eu/: https://android.jlelse.eu/consuming-rest-api-using-retrofit-library-in-android-ed47aef01ecb

[PRE10] Pressman, R. S. (2010). *Software Engineering Practitioner’s Approach. 2010. 7th Ed.* New York : McGraw-hill.

[ROS11] Rosmala, D. M. (2011). *Komparasi Framework MVC (Codeigniter, dan CakePHP) pada Aplikasi Berbasis Web (Studikasus: Sistem Informasi Perwalian Di Jurusan Informatika Institut Teknologi Nasional).* Bandung: ITENAS.

[SAT11] Satzinger. (2011). *Systems Analysis and Design in a Changing World, 6th Edition. Course Technology.*

[SQU15] Square Inc. (2015). *Github of Square Moshi.* Retrieved from Github.com: https://github.com/square/moshi

[W3C04] W3C. (2004). W3C Working Group Note. *Web Service Architecture*.