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

**(Studi Kasus Portofolio Gallery di Studio Chendra PT Dinasti Motekar Grup)**

**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 Portofolio Gallery di Studio Chendra PT Dinasti Motekar Grup)**

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](#_Toc56345680)

[ABSTRACT ii](#_Toc56345681)

[KATA PENGANTAR iii](#_Toc56345682)

[DAFTAR ISI iv](#_Toc56345683)

[DAFTAR ISTILAH vi](#_Toc56345684)

[DAFTAR TABEL vii](#_Toc56345690)

[DAFTAR GAMBAR viii](#_Toc56345691)

[BAB 1](#_Toc56345692) [PENDAHULUAN 1-1](#_Toc56345693)

[1.1 Latar Belakang 1-1](#_Toc56345694)

[1.2 Identifikasi Masalah 1-2](#_Toc56345695)

[1.3 Tujuan Tugas Akhir 1-2](#_Toc56345696)

[1.3 Lingkup Tugas Akhir 1-2](#_Toc56345697)

[1.4 Metodologi Tugas Akhir 1-3](#_Toc56345698)

[1.5 Sistematika Penulisan Tugas Akhir 1-4](#_Toc56345699)

[BAB 2](#_Toc56345700) [LANDASAN TEORI 2-1](#_Toc56345701)

[2.1. Aplikasi Mobile 2-1](#_Toc56345702)

[2.1.1. Perkembangan Aplikasi Mobile 2-1](#_Toc56345703)

[2.2. Sistem Operasi Android 2-2](#_Toc56345704)

[2.2.1. Aplikasi Android 2-3](#_Toc56345705)

[2.3. Web Service & REST API 2-3](#_Toc56345706)

[2.3.1. Metodologi REST API 2-4](#_Toc56345707)

[2.4. Retrofit Library 2-4](#_Toc56345708)

[2.5. Gson & Moshi 2-5](#_Toc56345709)

[2.6. Konsep MVC 2-5](#_Toc56345710)

[2.7. Software Development Life Cycle – Waterfall Model 2-5](#_Toc56345711)

[2.7.1. Tahapan – tahapan Waterfall Model 2-6](#_Toc56345712)

[2.8. Penelitian Terdahulu Terkait dengan Topik Tugas Akhir 2-7](#_Toc56345713)

[BAB 3](#_Toc56345714) [SKEMA PENELITIAN 3-1](#_Toc56345715)

[3.1. Rancangan Penelitian 3-1](#_Toc56345716)

[3.2. Analisis Masalah dan Solusi TA 3-4](#_Toc56345717)

[3.3. Kerangka Pemikiran Teoritis 3-5](#_Toc56345718)

[3.3.1 Rancangan Analisis 3-5](#_Toc56345719)

[3.3.2. Kerangka Berfikir Teoritis 3-7](#_Toc56345720)

[3.4. Analisis Teknologi 3-7](#_Toc56345721)

[3.4.1. Android Studio 3-7](#_Toc56345722)

[3.4.2. Retrofit 3-8](#_Toc56345723)

[3.4.3. JSON 3-8](#_Toc56345724)

[3.4.4. Gson & Moshi 3-9](#_Toc56345725)

[3.4.5. Web Service 3-9](#_Toc56345726)

[3.5. Profile Penelitian 3-9](#_Toc56345727)

[3.5.1. Objek Penelitian 3-9](#_Toc56345728)

[3.5.2. Tempat Penelitian 3-10](#_Toc56345729)

[3.5.3. Struktur Organisasi 3-10](#_Toc56345730)

[DAFTAR PUSTAKA](#_Toc56345731)

# DAFTAR ISTILAH

Tabel 1. Daftar Istilah

|  |  |  |
| --- | --- | --- |
| No. | Istilah | Definisi |
| 1. |  |  |
| 2. |  |  |

# DAFTAR TABEL

[Tabel 1. Daftar Istilah vi](#_Toc56345997)

[Tabel 2‑1. Daftar Penelitian Terdahulu 2-7](#_Toc56345998)

[Tabel 3‑1. Alur Penyelesaian Tugas Akhir 3-1](#_Toc56345999)

[Tabel 3‑2. Analisis manfaat implementasi dan perbandingan JSON Adapter Moshi dan Gson. 3-5](#_Toc56346000)

[Tabel 3‑3. Rancangan Analisis 3-5](#_Toc56346001)

[Tabel 3‑4. Penjelasan Rancangan Analisis 3-6](#_Toc56346002)

# DAFTAR GAMBAR

[Gambar 1‑1. Metodologi Tugas Akhir 1-4](#_Toc56346057)

[Gambar 2‑1. Pertumbuhan Aplikasi Mobile di setiap kategori di AS tahun 2009-2010 2-2](#_Toc56346058)

[Gambar 2‑2. Category aplikasi yang sering digunakan 2-2](#_Toc56346059)

[Gambar 2‑3. Web Service Flow 2-4](#_Toc56346060)

[Gambar 2‑4. Tahap pengembangan perangkat lunak pada waterfall model 2-6](#_Toc56346061)

[Gambar 3‑1. Kerangka Permasalahan 3-4](#_Toc56346062)

[Gambar 3‑2. Fishbone Diagram Solusi 3-7](#_Toc56346063)

[Gambar 3‑3. Struktur Organisasi Dinasti Motekar Academy 3-10](#_Toc56346064)

# 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 Notation* 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 membuat Portofolio Gallery di Studio Chendra PT Dinasti Motekar Grup. 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 membuat Portofolio Gallery di Studio Chendra PT Dinasti Motekar Grup.

## 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. Mengetahui Hasil perbandingan dari JSON adapter Gson dan Moshi yang digunakan dalam Pembangunan Aplikasi Berbasis Android.

## 1.3 Lingkup Tugas Akhir

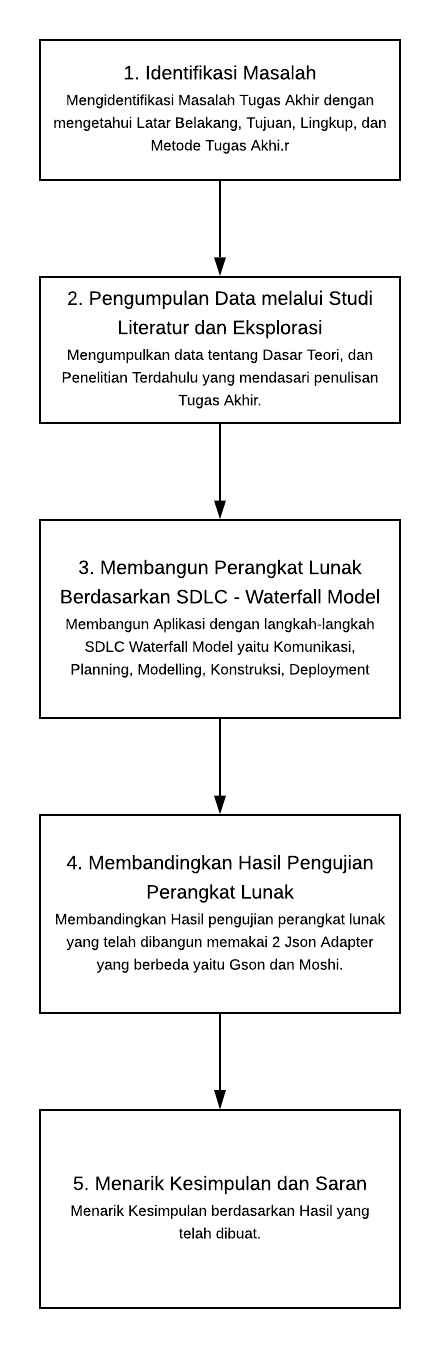
Penyelesaian Tugas Akhir dibatasi sebagai berikut :

1. Kasus yang dijadikan fokus penelitian adalah perbandingan dan pengimplementasian JSON adapter Aplikasi berbasis Android untuk Portofolio Gallery di Studio Chendra PT Dinasti Motekar Grup.
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

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

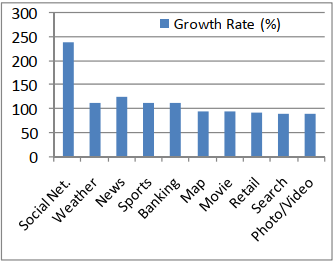
Sejumlah besar aplikasi *mobile* biasanya sudah diinstal sebelumnya di ponsel dan yang lainnya dapat diunduh oleh pengguna di internet dan menginstalnya di ponsel. Jumlah aplikasi *mobile* yang ada di market biasanya dibuat oleh berbagai macam *mobile application developer*, *publisher* dan *provider*. Dari segi teknis melihat berbagai aplikasi *mobile* yang dapat dijalankan di berbagai platform terkelola seperti iPhone, BlackBerry, Android, Symbian, Windows; juga beberapa mesin virtual seperti 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, Audio
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 Idle penghemat, Buku alamat, Manajer tugas, Manajer panggilan, Manajer file [IJE10]

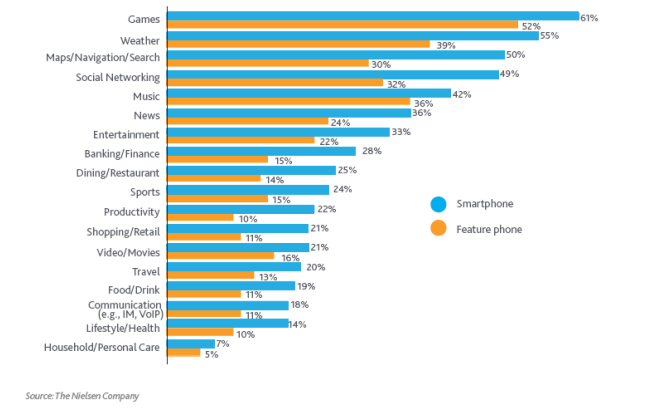
### 2.1.1. Perkembangan Aplikasi Mobile

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 aplikasi *mobile* yang sering di download. Dalam survei itu mereka menunjukkan beberapa data statistik penting dari aplikasi *mobile* 30 hari. 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 media sosial 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 dukung 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]

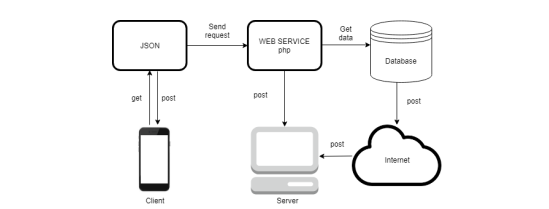
## 2.3. Web Service & REST API

*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 [INT19]

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 simpel 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 pemrpograman PHP dan MySQL sebagai *database* nya 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 respon diterima. [COD16]

Retrofit juga bekerja dengan REST API menggunakan implementasi java interface, yang dapat dihasilkan dengan bantuan Rest Adapter. 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 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] :

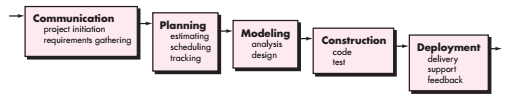
1. *Model*, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada  
   perubahan informasi. Hanya *model* yang mengandung data dan fungsi yang berhubungan  
   dengan pemrosesan data. Sebuah *model* meringkas lebih dari sekedar data dan fungsi yang  
   beroperasi di dalamnya. Pendekatan *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 satu banding satu 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 Model* adalah 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  [EAF19] | 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  [ANA17] | Aplikasi berbasis android di Lingkungan RW |
| 3. | Pembangunan Aplikasi Pakan Sapi Berbasis Android | Berta Erwin SLAM [SLA18] | 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 [NSM16] | Topik dan pembelajaran dengan bahasa python |
| 5. | Eksplorasi Python Menggunakan Framework Django (Studi Kasus : Sistem Perpustakaan Teknik Informatika UNPAS) | Anggiyasti Yaktining Putri [AYP16] | Sistem Aplikasi Perpustakaan |
| 6. | Rancang Bangun Aplikasi Pemesanan Tiket Di Kolam Renang Failda Sukanagara Cianjur Selatan | Muhamad Faisal Ilhami Akbar [MUH19] | Aplikasi Pemesanan Tiket Di Kolam Renang Failda Sukanagara Cianjur Selatan |

# 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.

## 3.1. Rancangan Penelitian

Rancangan penelitian ini memiliki tujuan untuk memudahkan dalam memahami secara singkat mengenai tahapan pengerjaan dan literatur yang digunakan dalam pengerjaan tugas akhir. Setiap tahapan yang dilakukan memiliki tujuan dan output. Sebagaimana dijelaskan di Tabel 3-1.

Tabel 3‑1. Alur Penyelesaian Tugas Akhir

| **Tahap dan hasil** | **Langkah penelitian** | **Literatur dan referensi** |
| --- | --- | --- |
| **Tahap 1** : Menentukan topik Tugas Akhir  **Hasil** : Mendapatkan Topik tugas akhir dan masalah yang telah teridentifikasi  **Kontibusi** : Berguna untuk pengumpulan data | Menentukan Topik Tugas Akhir  Mengidentifikasi Masalah |  |
| **Tahap 2 :** Studi literatur dan pengumpulan data  **Hasil** : Teori yang sudah dipahami dan kumpulan data yang didapat  **Kontribusi :** Berguna untuk tahap pengumpulan data dan spesifikasi kebutuhan | Mengumpulkan jurnal dan teori terkait  Memahami teori yang terkait  Melakukan observasi dan wawancara | 1. Wawancara,   [ACE19]   1. Perkembangan Aplikasi Android   [LIM14] [IJE10]   1. Aplikasi Android   [INT15]   1. Web Service & REST API   [W3C04][INT19]   1. Retrofit Libr ary   [CHE15][COD16][MAS15]   1. Gson & Moshi  [GOO08][SQU15] |
| **Tahap 3 :** Menerapkan rancangan penelitian  **Hasil :** Alur penyelesaian tugas akhir  **Kontribusi** : Berguna untuk perancangan perangkat lunak | Rancangan Penelitian  1  1  Membuat Alur Penyelesaian TA  Analisis Persoalan TA  Membuat solusi TA | 1. Rancang Bangun Aplikasi Pengelolaan Data Anggota Koordinator Olahraga Mahasiswa Universitas Pasundan Bandung [EAF19] 2. Perangkat Lunak Untuk Layanan Warga Berbasis Android Di Lingkungan Rukun Warga Menggunakan Konsep M-Government (Studi Kasus Rw 10 Kelurahan Sukabungah Kota Bandung) [ANA17] 3. Pembangunan Aplikasi Pakan Sapi Berbasis Android [SLA18] 4. Eksplorasi Python Untuk Penyesuaian Modul Perkuliahan Dan Praktikum Di Teknik Informatika Unpas [NSM16] 5. Eksplorasi Python Menggunakan Framework Django (Studi Kasus : Sistem Perpustakaan Teknik Informatika UNPAS) [AYP16] |
| **Tahap 4 :** Melakukan perancangan perangkat lunak  **Hasil :** Mendapatkan spesifikasi perangkat lunak, kebutuhan fungsional dan non-fungsional  **Kontribusi :** Berguna untuk tahap implementasi (koding) dan testing. | |  | | --- | | **Analisis Kebutuhan** | | Membuat pemodelan berbasis skenario  Membuat pemodelan berbasis data  Membuat pemodelan berbasis kelas  Membuat pemodelan perilaku (Sequence Diagram |   2  2   |  | | --- | | **Perancangan** | | Membuat perancangan data  Membuat perancangan arsitektural  Membuat perancangan antarmuka | | 1. Observasi dan wawancara [ACE19] 2. Model Proses Waterfall [PRE10] 3. Rancang Bangun Aplikasi Pemesanan Tiket Di Kolam Renang Failda Sukanagara Cianjur Selatan [MUH19] |
| **Tahap 5 :** Pembuatan aplikasi,  **Hasil** **:** Aplikasi yang sudah jadi  **Kontribusi :** Berguna untuk perbandingan kinerja 2 Json adapter | Pembuatan aplikasi | 1. Model Proses Waterfall [PRE10] 2. Pembangunan Aplikasi Pakan Sapi Berbasis Android [SLA18] 3. Perangkat Lunak Untuk Layanan Warga Berbasis Android Di Lingkungan Rukun Warga Menggunakan Konsep M-Government (Studi Kasus Rw 10 Kelurahan Sukabungah Kota Bandung) [ANA17] |
| **Tahap 6 :** Perbandingan 2 Json Adapter  **Hasil : Hasil** perbandingan kinerja Gson dan Moshi  **Kontribusi :** Berguna untuk tahap pengambilan kesimpulan tugas akhir | Pengujian aplikasi  Pencatatan hasil pengujian kedua JSON adapter | 1. Unit Testing [GUR20] |
| **Tahap 7 :** Penutup  **Hasil :** Kesimpulan dan saran tugas akhir  **Kontribusi : -** | Kesimpulan dan saran |  |

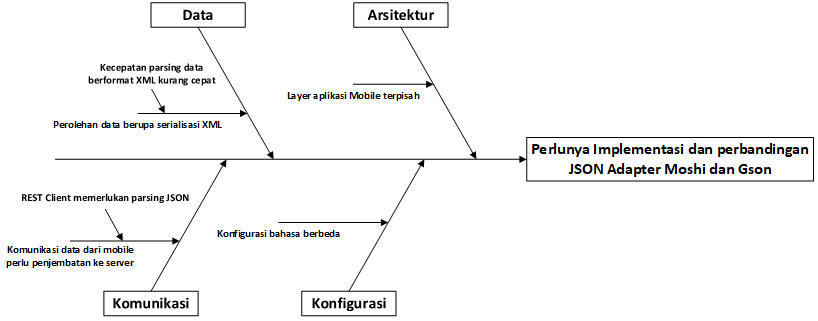
## 3.2. Analisis Masalah dan Solusi TA

Analisis Masalah Dan Solusi TA bertujuan untuk mengetahui ketepatan solusi terhadap suatu  
permasalahan pada penelitian tugas akhir, yaitu pada Perlunya implementasi dan perbandingan *JSON Adapter* Moshi dan Gson. Dengan memberikan sebuah solusi diharapkan dapat menyelesaikan permasalahan-permasalahan yang ada. Sehingga permasalahan yang telah didefinisikan dapat diselesaikan dengan solusi yang sesuai.

Tahap analisis pada bagian ini menggunakan diagram sebab dan akibat (*cause and effect diagram*) atau biasa disebut dengan *Fishbone* Diagram. *Fishbone* diagram adalah alat untuk mengidentifikasi dan mengorganisir penyebab yang mungkin terjadi dari suatu masalah dalam format yang terstruktur [KEL95]. Terdapat 3 komponen utama pada *fishbone* diagram, yaitu sebagai berikut :

1. *Head* (Kepala), merupakan fokus masalah yang sedang diamati, dideskripsikan dalam kotak di bagian kepala pada diagram. *Head* terletak pada bagian kanan diagram.
2. *Spine* (Tulang Belakang), ditunjukkan dengan arah panah yang menunjukkan penyebab dari masalah yang terjadi pada bagian *head*.
3. *Bones* (Tulang atau Duri), menunjukkan kategori utama yang menyebabkan permasalahan terjadi. Jika terdapat bones yang lebih kecil, memperlihatkan kedalaman penyebab dari permasalahan tersebut. *Bones* yang saling terkoneksi memperlihatkan bahwa penyebab permasalahan tersebut saling berhubungan [KEL95].

Berikut merupakan Gambar 3-1 kerangka permasalahan untuk analisis Implementasi dan Perbandingan kedua *JSON Adapter* Moshi dan Gson.



Gambar 3‑1. Kerangka Permasalahan

Berikut ini merupakan Tabel 3-2 hasil analisis manfaat implementasi dan perbandingan *JSON Adapter* Moshi dan Gson. Terdapat masalah penyebab, dan solusi untuk menyelesaikan permasalahan.

Tabel 3‑2. Analisis manfaat implementasi dan perbandingan JSON Adapter Moshi dan Gson.

|  |  |  |
| --- | --- | --- |
| **Masalah** | **Penyebab** | **Solusi** |
| Perlunya Implementasi dan perbandingan JSON Adapter Moshi dan Gson. | **Data**  Perolehan data berupa serialisasi XML yang kecepatan parsing data nya kurang cepat | Melakukan Implementasi dan perbandingan JSON Adapter Moshi dan Gson.untuk Parsing JSON dari REST Client |
| **Arsitektur**  Layer data aplikasi mobile berbeda dengan web yang semuanya menyatu dalam satu app, sedangkan untuk mobile karena keterbatasannya menjadikannya terpisah - pisah |
| **Komunikasi**  Komunikasi data dari aplikasi mobile perlu penjembatan ke server  Setelah dipilih penjembatan yaitu Rest Client, lalu memerlukan parsing JSON untuk menampilkan datanya |
| **Konfigurasi**  Konfigurasi penggunaan Bahasa yang berbeda pada tial Rest Client dan JSON adapter |

## 3.3. Kerangka Pemikiran Teoritis

Pada tahap kerangka pemikiran teoritis ini akan dijelaskan megenai tahapan yang dilakukan dalam Implementasi dan Perbandingan *JSON Adapter* Moshi dan Gson. Tahap ini berisi mengenai kerangka pemikiran teoritis yang di dalamnya menunjukkan tentang komponen-komponen pembentuk produk tugas akhir yang disertai dengan kajian konsep dan teori-teori yang mendasarinya.

### 3.3.1 Rancangan Analisis

Rancangan analisis merupakan gambaran dari proses-proses analisis yang dilakukan. Dalam penggambaran rancangan analisis, terdapat input, proses, dan output. Input merupakan data yang akan dimanfaatkan pada proses analisis yang akan menghasilkan output. Rancangan analisis dapat dilihat pada Tabel 3-3.

Tabel 3‑3. Rancangan Analisis

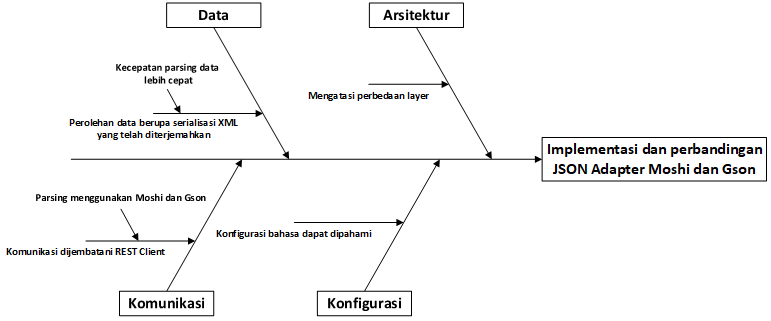
| **Input** | **Analisis** | **Output** |
| --- | --- | --- |
|  | Communication  Menganalisis perangkat lunak yang akan dibangun |  |
|  | Mendefinisikan kebutuhan  perangkat lunak  Planning |  |
|  | Modelling  Membuat rancangan dari perangkat lunak yang akan dibangun |  |

Rancangan analisis pada Table 3-3 dapat dijelaskan secara rinci pada Table 3-4

Tabel 3‑4. Penjelasan Rancangan Analisis

| **No.** | **Input** | **Analisis** | **Output** |
| --- | --- | --- | --- |
| 1. | Hasil wawancara dengan stakeholder | Menganalisis perangkat lunak yang akan dibangun | Lingkup perangkat lunak yang akan dibangun  Lingkup dari perangkat lunak yang akan dibangun bertujuan mendefinisikan lingkup pembangunan perangkat lunak yang akan dibangun terkait hasil wawancara yang dilakukan |
| 2. | Lingkup perangkat lunak yang akan dibangun | Mendefinisikan kebutuhan perangkat lunak | Kebutuhan perangkat lunak  Kebutuhan perangkat lunak tersebut mendeskripsikan fitur-fitur yang akan tersedia pada perangkat lunak demi menunjang penyelesaian masalah yang terjadi. |
| 3. | Kebutuhan perangkat lunak | Membuat rancangan dari perangkat lunak yang akan dibangun | Rancangan antarmuka dan pemodelan aplikasi yang akan dibangun  Pemodelan aplikasi bertujuan untuk merepresentasikan hasil analisis yang dilakukan berdasarkan kebutuhan perangkat lunak yang sudah ditentukan. Sedangkan untuk rancangan antarmuka bertujuan untuk membuat interaksi antara user dan aplikasi ketika digunakan. |

### 3.3.2. Kerangka Berfikir Teoritis

Pada kerangfka berpikir teoritis akan dibuat solusi berdasarkan permasalahan yang telah diketahui diata. Berdasarkan permasalahan diatas dibutuhkan solusi berupa Aplikasi publikasi informasi dan kegiatan di Dinasti Moterkar Academy yang dapat mengatasi permasalahan tersebut. Seperti pada gambar di bawah ini.

Gambar 3‑2. Fishbone Diagram Solusi

## 3.4. Analisis Teknologi

Analisis teknologi bertujuan menentukan teknologi-teknologi yang sesuai untuk menyelesaikan permasalahan yang sesuai dengan topik tugas akhir. Teknologi-teknologi yang akan digunakan akan dipaparkan pada poin-poin berikut ini:

### 3.4.1. Android Studio

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android, dibuat berdasarkan IntelliJ IDEA . Selain editor kode dan alat pengembang IntelliJ yang kuat, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas saat membuat aplikasi Android. [GOO20]

Alasan menggunakan Android Studio untuk membuat aplikasi mobile adalah karena android  
merupakan sistem operasi yang *open source*. Kode dengan sumber terbuka atau open source dan  
terdapat lisensi perizinan pada Android memungkinkan perangkat lunak untuk memodifikasi secara  
bebas dan didistribusikan oleh para pembuat perangkat atau pengembang aplikasi.

#### **3.4.1.1. Fitur – Fitur Android Studio**

* Sistem *build* berbasis *Gradle* yang fleksibel
* *Emulator* yang cepat dan kaya fitur
* Lingkungan terpadu yang dapat mengembangkan untuk semua perangkat Android
* Membuat perubahan dan pergantian resource tanpa harus me *restart* app
* Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
* Alat dan kerangka pengujian yang ekstensif
* Alat Lint untuk mengetahui kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya
* Dukungan C ++ dan NDK
* Dukungan bawaan untuk Google Cloud Platform , yang memudahkan integrasi Google Cloud *Messaging* dan App Engine [GOO20]

### 3.4.2. Retrofit

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 Rest Adapter. Implementasi dalam hal ini bertindak sebagai local instance dari layanan dan setiap panggilan sesuai dengan permintaan HTTP. [MAS15]

Alasan menggunakan retrofit adalah karena kemudahan dalam interaksi dengan API dan  
fleksibilitasnya yang memudahkan untuk membuat *custom service* sesuai yang kita inginkan.

### 3.4.3. JSON

JSON adalah JSON (*JavaScript Object Notation*) adalah format pertukaran data (*lightweight data inter change format*), mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 –Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. [JSO20]

Alasan utama menggunakan JSON (*Javascript Object Notation*) adalah karena JSON format pertukaran datanya yang ringan dan juga banyak didukung oleh bahasa pemrograman. Maka dari itu, penulis menggunakan JSON sebagai format pertukaran datanya.

### 3.4.4. 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]

Alasan uatama menggunakan Gson dan moshi adalah karena penulis ingin membandingkan kinerja dari ke-2 JSON *Library* atau sering dipanggil JSON *adapter*, sehingga nanti dapat diketahui JSON *Adapter* mana yang lebih baik dari kinerjanya.

### 3.4.5. Web Service

*Web service* merupakan sebuah perangkat lunak yang digunakan sebagai jembatan sehingga memungkinkan berbagai sistem dapat saling berkomunikasi tanpa terpengaruh dengan perbedaan platform.

Pada Implementasi dan perbandingan JSON Adapter Moshi dan Gson terdapat dua platform yang akan menggunakan *web service* untuk melakukan pertukaran data dengan *database* yang berada di sistem yaitu aplikasi *Web* yang telah dibangun dan Android yang akan dibangun. Dengan adanya *web service* kedua platform tersebut hanya perlu memanggil fungsi-fungsi yang sudah disediakan oleh *web service.*

## 3.5. Profile Penelitian

Profil penelitian menjelaskan tentang objek penelitian, tempat dimana peneliti melakukan penelitian, dan hal-hal lain yang berkaitan dengan penelitian.

### 3.5.1. Objek Penelitian

Di PT. Dinasti Motekar Grup terdapat bidang usaha salah satunya adalah unit usaha dokumentasi yang bernama Studio Chendra yang berfokus pada layanan jasa Dokumentasi, layanan jasa yang disediakan berupa Fotografi, Videografi, dan Desain Grafis.

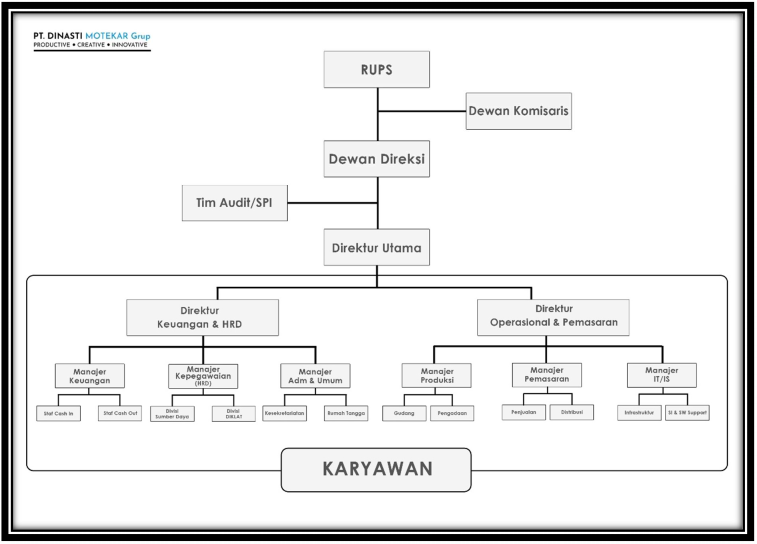
Layanan jasa Dokumentasi ini memiliki portofolio gallery yang berjalan di sebuah website untuk menunjukan hasil pekerjaan serta informasi tentang perusahaan, dengan adanya website yang telah dibuat ini diharapkan calon client dapat mengaksesnya untuk melihat keseluruhuan informasi yang dibutuhkan tentang perusahaan ini.

Pada penelitian ini fokus peneliti adalah membangun sebuah aplikasi berbasis android yang menampung semua informasi yang dibutuhkan dari sebuah website yang telah dibuat, agar mobilitas akses informasi meningkat, serta mengimplentasikan dan melakukan perbandingan JSON adapter pada aplikasi yang akan dibangun nantinya.

### 3.5.2. Tempat Penelitian

Tempat penelitian berada di Kota Cimahi tepatnya di Dinasti Motekar Academy yang beralamatkan di Jl. Rorojongrang Raya A12-12 Jawa Barat. Penelitian dilakukan pada bagian IT/IS. Dinasti Motekar Academy didirikan pada tanggal 17 Mei 2018 oleh Acep Hendra bersama Ferry Mulyanto dan Muhammad Tirta Mulia yang sama – sama berguru dengan Tung Desem Waringin, sebagai perwujudan dari konsep bebagi, dengan tagline “HayyunYuhyi” yang berarti Tumbuh dan Menumbuhkan.

### 3.5.3. Struktur Organisasi

Berikut adalah gambar dari struktur organisasi tempat peneliti melakukan penelitian tugas akhir yaitu PT. Dinasti Motekar Grup. Sebagaimana dapat dilihat pada Gambar 3-3.

Gambar 3‑3. Struktur Organisasi Dinasti Motekar Academy

Bagian Infrastruktur, IT dan SI memiliki job desk sebagai berikut :

1. Mengelola Teknologi Informasi dan sistem computer.
2. Bertanggung jawab pada kesiapan dan ketersediaan sistem komputer / aplikasi dalam lingkungan perusahaan.
3. Membuat dan/atau implementasi semua sistem dan aplikasi
4. Merancang, mengelola dan mengawasi serta meng-evaluasi operasional dari sistem informasi (software dan aplikasi) dan pendukungnya (hardware, infrastruktur, telekomunikasi).
5. Membuat dan mengimplementasikan kebijakan dan prosedur TI (IT policy) termasuk kebijakan keamanan TI (IT security policy).
6. Berkerja sama dengan TI vendor untuk merancang , membuat dan mengimplementasikan sistem atau aplikasi jika diperlukan.
7. Membuat dan mengawasi anggaran TI (budget) dan expenditures.
8. Memberikan solusi Teknologi Informasi.
9. Bertanggung jawab pada penyediaan layanan infrastruktur termasuk aplikasi, jaringan komputer (LAN / WAN), keamanan Teknologi Informasi dan telekomunikasi.
10. Memberikan rekomendasi tentang solusi sistem informasi dan pendukungnya.
11. Memberikan orientasi kepada pegawai baru mengenai aplikasi atau system yang digunakan saat ini dan rencana atau strategi TI secara umum.
12. Memberikan arahan pada bawahan mengenai penggunaan dan solusi teknologi.
13. Pengawasan dan perawatan Teknologi Informasi.
14. Bekerja sama dengan senior manajemen untuk membuat, merancang pelayanan TI dalam dokumen Service Level Agreements.
15. Memberikan laporan bulanan kepada Direktur Keuangan mengenai semua aspek dari departemen TI (Teknologi Informasi).

Dalam kasus kali ini peneliti melakukan tahapan komunikasi dengan direktur utama yakni Acep Hendra, ST., C.Ht., CFDM., dikarenakan pada divisi Infarstruktur IT/IS masih belum memiliki manager.

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [ACE19]  [ANA17]  [AYP16]  [BUR19]  [CHE15]  [COD16]  [EAF19]  [GOO20]  [GOO08]  [GUR20]  [INT19]  [JSO20]  [KEL95]  [INT15]  [KUC19]  [LIM15]  [NSM16]  [MAS15]  [IJE10]  [MUH19]  [OSA17]  [PRE10]  [ROS11]  [SAT11]  [SLA18]  [SQU15]  [W3C04] | Acep, H. (2019). Dinasti Motekar Academy.  Akbar, A. N. (2017). Perangkat Lunak Untuk Layanan Warga Berbasis Android Di Lingkungan Rukun Warga Menggunakan Konsep M-Government (Studi Kasus Rw 10 Kelurahan Sukabungah Kota Bandung).  Anggiyasti, Y. P. (2016). Eksplorasi Python Menggunakan Framework Django (Studi Kasus : Sistem Perpustakaan Teknik Informatika UNPAS).  Burbeck, S. (1992). *Applications Programming in Smalltalk-80 (TM) How to use Model-View-Controller (MVC). Vol. 2.5.* ParcPlace Systems, Inc.  Chen, H. (2015). *Management of contextual information for data.*  Codepath. (2016). *Consuming APIs with Retrofit*. Retrieved from Codepath: https://guides.codepath.com/android/Con  Febriyanti, E. A. (2019). Rancang Bangun Aplikasi Pengelolaan Data Anggota Koordinator Olahraga Mahasiswa Universitas Pasundan Bandung.  Google, JetBrains. (2020, 11 11). *Android Studio Intro.* Retrieved from developer.android.com: https://developer.android.com/studio/intro  Google.Inc. (2008). *Github of Google Gson*. Retrieved from Github.com: https://github.com/google/gson  Guru99. (2020, 11 10). *Unit Testing Guide.* Retrieved from Guru99: https://www.guru99.com/unit-testing-guide.html  Ika Oktavia Suzanti, N. F. (2019). International Conference on Science and Technology. *REST API Implementation on Android Based Monitoring*.  JSON. (2020, 11 11). *Intoducing JSON.* Retrieved from json.org: https://www.json.org/json-en.html  Kelleher, K. C. (1995). *Cause and Effect Diagram : Plain and.* Joiner Associates Inc USA.  Kirthika.B, P. V. (2015). International Journal of Trend in Research and Development. *Android Operating System: A Review*, 260.  Kucherenko, S. M. (2019). *Learn Spring for Android Application Development.* Brimingham: Packt.  Lim, S. H. (2015). Experimental Comparison of Hybrid and Native Applications for Mobile System. *International Journal of Multimedia and Ubiquitous Engineering*.  Malikhah, N. S. (2016). Eksplorasi Python Untuk Penyesuaian.  Maskov. (2015). *Implementing REST Client for Android.* Helsinki: Metropolia University of Applied Sciences.  Md. Rashedul Islam, M. R. (2010). International Journal of Engineering & Technology IJET-IJENS Vol: 10 No: 06. *Mobile Application and Its Global Impact*, 104.  Muhammad, F. I. (2019). Rancang Bangun Aplikasi Pemesanan Tiket Di Kolam Renang Failda Sukanagara Cianjur Selatan.  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  Pressman, R. S. (2010). *Software Engineering Practitioner’s Approach. 2010. 7th Ed.* New York: McGraw-hill.  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.  Satzinger. (2011). *Systems Analysis and Design in a Changing World, 6th Edition. Course Technology.*  Slam, B. E. (2018). Pembangunan Aplikasi Pakan Sapi Berbasis Android.  Square Inc. (2015). *Github of Square Moshi.* Retrieved from Github.com: https://github.com/square/moshi  W3C. (2004). W3C Working Group Note. *Web Service Architecture*. |