**EKSPLORASI JSON ADAPTER MOSHI DAN GSON UNTUK RETROFIT PADA PEMBANGUNAN APLIKASI BERBASIS ANDROID**

**(Studi Kasus Publikasi informasi dan kegiatan 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**

**DESEMBER 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 MOSHI DAN GSON UNTUK RETROFIT PADA PEMBANGUNAN APLIKASI BERBASIS ANDROID**

**(Studi Kasus Publikasi informasi dan kegiatan 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

Web service, REST API, Retrofit, JSON Adapter dan teknologi lainya kian banyak digunakan dalam pembangunan aplikasi berbasis Android. Pengambilan data yang biasanya dijembatanai oleh Web Service dan juga REST API untuk mendapatkan data dari sebuah database yang dimuat oleh sebuah website juga sangat efektif belakangan ini. Namun hal ini biasanya dilakukan dengan mengconvert data tersebut menjadi sebuah objek JSON, maka diperlukanlah sebuah JSON adapter untuk menunjang kecepatan transfer data.

Moshi dan Gson adalah JSON adapter yang dapat menerjemahkan bentuk objek atau data JSON menjadi sebuah data utuh yang kemudian dapat digunakan dalam sebuah aplikasi yang dijalankan di sistem operasi android, Moshi dan Gson pun belum banyak diketahui sehingga perlu diketahui kinerja dan cara implementasi dari kedua JSON Adapter tersebut.

Munculah gagasan untuk melakukan eksplorasi tentang bagaimana cara mengimplementasikan Moshi dan Gson sebagai JSON Adapter untuk pembangunan aplikasi berbasis android yang dibangun dengan metode SDLC – Waterfall Model.

Kata Kunci : Web service, Retrofit, JSON Adapter, Moshi, Gson

# ABSTRACT

Web services, REST API, Retrofit, JSON Adapter and other technologies are increasingly being used in the development of Android-based applications. Retrieval of data that is usually bridged by Web Services and also REST APIs to get data from a database loaded by a website is also very effective lately. However, this is usually done by converting the data into a JSON object, so we need a JSON adapter to support data transfer speeds.

Moshi and Gson are JSON adapters that can translate the form of JSON objects or data into intact data which can then be used in an application running on the Android operating system, Moshi and Gson are not widely known so it is necessary to know the performance and implementation of the two JSON Adapters.

An idea emerged to explore how to implement Moshi and Gson as a JSON Adapter for Android-based application development built using the SDLC - Waterfall Model method.

Keywords: Web service, Retrofit, JSON Adapter, Moshi, Gson

# 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 Moshi dan Gson untuk Retrofit pada Pembangunan Pplikasi berbasis Android (Studi Kasus Publikasi Informasi dan Kegiatan di Studio Chendra PT Dinasti Motekar Grup)”

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](#_Toc58487381)

[ABSTRACT ii](#_Toc58487382)

[KATA PENGANTAR iii](#_Toc58487383)

[DAFTAR ISI iv](#_Toc58487384)

[DAFTAR TABEL vii](#_Toc58487385)

[DAFTAR GAMBAR viii](#_Toc58487386)

[BAB 1 PENDAHULUAN 1-1](#_Toc58487387)

[1.1. Latar Belakang 1-1](#_Toc58487388)

[1.2. Identifikasi Masalah 1-2](#_Toc58487389)

[1.3. Tujuan Tugas Akhir 1-2](#_Toc58487390)

[1.4. Lingkup Tugas Akhir 1-3](#_Toc58487391)

[1.5. Metodologi Tugas Akhir 1-3](#_Toc58487392)

[1.6. Sistematika Penulisan Tugas Akhir 1-4](#_Toc58487393)

[BAB 2 LANDASAN TEORI 2-1](#_Toc58487394)

[2.1. Aplikasi Mobile 2-1](#_Toc58487395)

[2.1.1. Perkembangan Aplikasi Mobile 2-1](#_Toc58487396)

[2.2. Sistem Operasi Android 2-3](#_Toc58487397)

[2.2.1. Aplikasi Android 2-3](#_Toc58487398)

[2.3. Web Service & REST API 2-3](#_Toc58487399)

[2.3.1. Metodologi REST API 2-4](#_Toc58487400)

[2.4. Retrofit Library 2-4](#_Toc58487401)

[2.5. Gson & Moshi 2-5](#_Toc58487402)

[2.6. Konsep MVC 2-5](#_Toc58487403)

[2.7. Software Development Life Cycle – Waterfall Model 2-5](#_Toc58487404)

[2.7.1. Tahapan – tahapan Waterfall Model 2-6](#_Toc58487405)

[2.8. Penelitian terdahulu terkait dengan topik Tugas Akhir 2-7](#_Toc58487406)

[BAB 3 SKEMA PENELITIAN 3-1](#_Toc58487407)

[3.1. Rancangan Penelitian 3-1](#_Toc58487408)

[3.2. Analisis Masalah dan Solusi Tugas Akhir 3-4](#_Toc58487409)

[3.3. Kerangka Pemikiran Teoritis 3-5](#_Toc58487410)

[3.3.1. Rancangan Analisis 3-6](#_Toc58487411)

[3.3.2. Kerangka Berfikir Teoritis 3-7](#_Toc58487412)

[3.4. Analisis Teknologi 3-8](#_Toc58487413)

[3.4.1. Android Studio 3-8](#_Toc58487414)

[3.4.2. Retrofit 3-8](#_Toc58487415)

[3.4.3. JSON 3-9](#_Toc58487416)

[3.4.4. Gson & Moshi 3-9](#_Toc58487417)

[3.4.5. Web Service 3-9](#_Toc58487418)

[3.5. Profile Penelitian 3-10](#_Toc58487419)

[3.5.1. Objek Penelitian 3-10](#_Toc58487420)

[3.5.2. Tempat Penelitian 3-10](#_Toc58487421)

[3.5.3. Struktur Organisasi 3-10](#_Toc58487422)

[BAB 4 ANALISIS JSON ADAPTER MOSHI DAN GSON 4-1](#_Toc58487423)

[4.1. Analisis JSON Adapter Moshi 4-1](#_Toc58487424)

[4.1.1. Tipe Adaptor Bawaan Moshi 4-1](#_Toc58487425)

[4.1.2. Kostum Tipe Adaptor 4-2](#_Toc58487426)

[4.1.3. Parsing Array JSON 4-5](#_Toc58487427)

[4.1.4. Respon Fail 4-5](#_Toc58487428)

[4.1.5. Kostum Nama Field dengan @Json 4-6](#_Toc58487429)

[4.1.6. Tipe Adaptor Alternatif dengan @JsonQualifier 4-6](#_Toc58487430)

[4.1.7. Penggunaan Transient 4-7](#_Toc58487431)

[4.1.8. Nilai & Constructor Default 4-8](#_Toc58487432)

[4.1.9. Menggabungkan Adaptor 4-9](#_Toc58487433)

[4.1.10. Perbedaan dengan Gson 4-10](#_Toc58487434)

[4.2. Analisis Json Adapter Gson 4-10](#_Toc58487435)

[4.2.1. Tujuan Gson 4-11](#_Toc58487436)

[4.2.2. Peforma dan Skalabilitan Gson 4-11](#_Toc58487437)

[4.2.3. Menggunakan Gson 4-11](#_Toc58487438)

[4.2.4. Parsing Array JSON 4-14](#_Toc58487439)

[4.2.5. Collections 4-14](#_Toc58487440)

[4.2.6. Serializing dan deserializing tipe Generic 4-14](#_Toc58487441)

[4.2.1. Serializing and Deserializing Collection dengan tipe objek arbitrary 4-15](#_Toc58487442)

[4.2.2. Serializers dan Deserializers bawaan 4-16](#_Toc58487443)

[4.2.3. Kostum Serialization and Deserialization 4-16](#_Toc58487444)

[4.2.4. Menulis Pembuat Instance 4-17](#_Toc58487445)

[4.2.5. Dukungan Null Objek 4-19](#_Toc58487446)

[BAB 5 ANALISIS DAN PERANCANGAN STUDI KASUS 5-1](#_Toc58487447)

[5.1. Analisis Sistem yang sedang berjalan 5-1](#_Toc58487448)

[5.2. Analisis Kebutuhan 5-1](#_Toc58487449)

[5.2.1. Menentukan Kebutuhan Dasar 5-1](#_Toc58487450)

[5.2.2. Analisis Pengguna 5-2](#_Toc58487451)

[5.2.3. Analisis Ranah 5-2](#_Toc58487452)

[5.2.4. Pendekatan untuk Pemodelan Spesifikasi Kebutuhan 5-3](#_Toc58487453)

[5.2.5. Pemodelan Berbasis Skenario 5-4](#_Toc58487454)

[5.2.6. Pemodelan data 5-26](#_Toc58487455)

[5.2.7. Pemodelan Berbasis Kelas 5-29](#_Toc58487456)

[5.2.8. Pemodelan Berbasis Perilaku 5-39](#_Toc58487457)

[5.3. Perancangan Perangkat Lunak 5-46](#_Toc58487458)

[5.3.1. Perancangan Data 5-46](#_Toc58487459)

[5.3.2. Perancangan Arsitektural 5-47](#_Toc58487460)

[5.3.3. Perancangan Antarmuka Pengguna 5-48](#_Toc58487461)

[DAFTAR PUSTAKA](#_Toc58487462)

# DAFTAR TABEL

[Tabel 2‑1. Daftar Penelitian Terdahulu 2-7](#_Toc58487463)

[Tabel 3‑1. Alur Penyelesaian Tugas Akhir 3-1](#_Toc58487464)

[Tabel 3‑2. Analisis manfaat implementasi dan perbandingan 3-5](#_Toc58487465)

[Tabel 3‑3. Rancangan Analisis 3-6](#_Toc58487466)

[Tabel 3‑4. Penjelasan Rancangan Analisis 3-6](#_Toc58487467)

[Tabel 5‑1. Daftar Stakeholder 5-1](#_Toc58487468)

[Tabel 5‑2. Analisis Fungsional 5-2](#_Toc58487469)

[Tabel 5‑3. Daftar Pengguna 5-2](#_Toc58487470)

[Tabel 5‑4. Deskripsi hasil output analisis ranah 5-3](#_Toc58487471)

[Tabel 5‑5. Deskripsi Use Case 5-5](#_Toc58487472)

[Tabel 5‑6. Skenario Use Case menampilkan informasi workshop berupa foto atau video 5-6](#_Toc58487473)

[Tabel 5‑7. Skenario Use Case menginput pesan berupa teks 5-7](#_Toc58487474)

[Tabel 5‑8. Skenario Use Case menampilkan artikel fotografi berupa foto atau video 5-7](#_Toc58487475)

[Tabel 5‑9. Skenario Use Case menampilkan iklan layanan jasa 5-8](#_Toc58487476)

[Tabel 5‑10. Skenario Use Case Menampilkan portofolio gallery 5-9](#_Toc58487477)

[Tabel 5‑11. Skenario Use Case menginput registrasi order jasa dokumentasi 5-9](#_Toc58487478)

[Tabel 5‑12. Skenario Use Case Memposting video atau foto workshop 5-10](#_Toc58487479)

[Tabel 5‑13. Skenario Use Case Menginput jawaban berupa teks 5-10](#_Toc58487480)

[Tabel 5‑14. Skenario Use Case Memposting video atau foto artikel fotografi 5-11](#_Toc58487481)

[Tabel 5‑15. Skenario Use Case Memposting video atau foto iklan layanan jasa 5-12](#_Toc58487482)

[Tabel 5‑16. Skenario Use Case Memposting foto untuk portofolio gallery 5-12](#_Toc58487483)

[Tabel 5‑17. Skenario Use Case Menampilkan data registrasi order jasa 5-13](#_Toc58487484)

[Tabel 5‑18. Objek Data 5-26](#_Toc58487485)

[Tabel 5‑19. Atribut data 5-27](#_Toc58487486)

[Tabel 5‑20. Relasi Objek Data 5-28](#_Toc58487487)

[Tabel 5‑21. Kelas - Kelas Analisis 5-29](#_Toc58487488)

[Tabel 5‑22. Atribut Kelas 5-32](#_Toc58487489)

[Tabel 5‑23. Perilaku Kelas 5-34](#_Toc58487490)

[Tabel 5‑24. Perancangan Data 5-46](#_Toc58487491)

# DAFTAR GAMBAR

[Gambar 1‑1. Metodologi Tugas Akhir 1-3](#_Toc58487524)

[Gambar 2‑1. Pertumbuhan Aplikasi Mobile di setiap kategori di AS tahun 2009-2010 2-2](#_Toc58487525)

[Gambar 2‑2. Category aplikasi yang sering digunakan 2-2](#_Toc58487526)

[Gambar 2‑3. Web Service Flow 2-4](#_Toc58487527)

[Gambar 2‑4. Tahap pengembangan perangkat lunak pada waterfall model 2-6](#_Toc58487528)

[Gambar 3‑1. Fishbone Diagram Permasalahan 3-5](#_Toc58487529)

[Gambar 3‑2. Fishbone Diagram Solusi. 3-7](#_Toc58487530)

[Gambar 3‑3. Struktur Organisasi Dinasti Motekar Academy 3-11](file:///D:\Document\Work\Kuliah\1.TA\My%20TA\All\TAv9.docx#_Toc58487531)

[Gambar 5‑1. Analisis Ranah 5-3](#_Toc58487532)

[Gambar 5‑2. Elemen Model Analisis 5-4](#_Toc58487533)

[Gambar 5‑3. Use Case Diagram 5-5](#_Toc58487534)

[Gambar 5‑4. Swimlane Diagram pada UC-01 5-14](#_Toc58487535)

[Gambar 5‑5. Diagram Swimlane pada UC-02 5-15](#_Toc58487536)

[Gambar 5‑6. Swimlane Diagram pada UC-03 5-16](#_Toc58487537)

[Gambar 5‑7. Swimlane Diagram pada UC-04 5-17](#_Toc58487538)

[Gambar 5‑8. Swimlane Diagram pada UC-05 5-18](#_Toc58487539)

[Gambar 5‑9. Swimlane Diagram pada UC-06 5-19](#_Toc58487540)

[Gambar 5‑10. Swimlane Diagram pada UC-07 5-20](#_Toc58487541)

[Gambar 5‑11. Swimlane Diagram pada UC-08 5-21](#_Toc58487542)

[Gambar 5‑12. Swimlane Diagram pada UC-09 5-22](#_Toc58487543)

[Gambar 5‑13. Swimlane Diagram pada UC-10 5-23](#_Toc58487544)

[Gambar 5‑14. Swimlane Diagram pada UC-11 5-24](#_Toc58487545)

[Gambar 5‑15. Swimlane Diagram pada UC-12 5-25](#_Toc58487546)

[Gambar 5‑16. Diagram Kelas 5-39](#_Toc58487547)

[Gambar 5‑17. Diagram Sekuen Menampilkan infromasi workshop 5-40](#_Toc58487548)

[Gambar 5‑18. Diagram sekuen menginput pesan berupa teks untuk keluhan 5-40](#_Toc58487549)

[Gambar 5‑19. Diagram Sekuen Menampilkan artikel fotografi berupa foto atau video 5-41](#_Toc58487550)

[Gambar 5‑20. Diagram Sekuen Menampilkan iklan layanan jasa dokumentasi 5-41](#_Toc58487551)

[Gambar 5‑21. Diagram Sekuen Menampilkan portofolio gallery 5-42](#_Toc58487552)

[Gambar 5‑22. Diagram Sekuen Menginput registrasi order jasa dokumentasi 5-42](#_Toc58487553)

[Gambar 5‑23. Diagram Sekuen Memposting video atau foto workshop 5-43](#_Toc58487554)

[Gambar 5‑24. Diagram Sekuen Menginput jawaban berupa teks 5-43](#_Toc58487555)

[Gambar 5‑25. Diagram Sekuen Memposting video atau foto artikel fotografi 5-44](#_Toc58487556)

[Gambar 5‑26. Diagram sekuen Memposting video atau foto iklan layanan jasa 5-44](#_Toc58487557)

[Gambar 5‑27. Diagram Sekuen Memposting foto untuk portofolio gallery 5-45](#_Toc58487558)

[Gambar 5‑28. Diagram Sekuan Menampilkan data registrasi order jasa dokumentasi 5-45](#_Toc58487559)

[Gambar 5‑29. Gaya Arsitektur Berpusat pada data 5-47](#_Toc58487560)

[Gambar 5‑30. Arsitektur Teknologi 5-48](#_Toc58487561)

[Gambar 5‑31. Antarmuka menampilkan informasi workshop berupa foto atau video 5-48](#_Toc58487562)

[Gambar 5‑32. Antarmuka menginput pesan berupa teks untuk keluhan 5-49](#_Toc58487563)

[Gambar 5‑33. Antarmuka menampilkan artikel fotografi berupa foto atau video 5-49](#_Toc58487564)

[Gambar 5‑34. Antarmuka menampilkan iklan layanan jasa dokumentasi 5-50](#_Toc58487565)

[Gambar 5‑35. Antarmuka menampilkan portofolio gallery 5-50](#_Toc58487566)

[Gambar 5‑36. Antarmuka menginput registrasi order jasa dokumentasi 5-51](#_Toc58487567)

[Gambar 5‑37. Antarmuka Memposting video atau foto workshop 5-51](#_Toc58487568)

[Gambar 5‑38. Antarmuka Menginput jawaban berupa teks untuk menjawab 5-52](#_Toc58487569)

[Gambar 5‑39. Antarmuka memposting video atau foto artikel fotografi 5-52](#_Toc58487570)

[Gambar 5‑40. Antarmuka memposting video atau foto iklan layanan jasa dokumentasi 5-53](#_Toc58487571)

[Gambar 5‑41. Antarmuka memposting foto untuk portofolio gallery 5-53](#_Toc58487572)

[Gambar 5‑42. Antarmuka Menampilkan data registrasi order jasa dokumentasi 5-54](#_Toc58487573)

# 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.

## 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 digunakan 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 aplikasi Publikasi informasi dan kegiatan 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 aplikasi publikasi informasi dan kegiatan di Studio Chendra PT Dinasti Motekar Grup.

## 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?

## 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.

## 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 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.

## Metodologi Tugas Akhir



**Gambar 1‑1. Metodologi Tugas Akhir**

1. Identifikasi Masalah

Mengidentifikasi masalah tugas akhir dengan mengetahui latar belakang tugas akhir, tujuan tugas akhir, lingkup tugas akhir dan metode tugas akhir.

1. Pengumpulan data melalui studi literatur dan eksplorasi

Mengumpulan data tentang dasar teori dan penelitian terdahulu yang mendasari penulisan tugas akhir.

1. Menganalisis JSON Adapter Moshi dan Gson

Pada metode analisis dilakukan untuk mengetahui JSON Adapter Moshi dan JSON adapter Gson berdasarkan referensi yang nantinya akan digunakan sebagai dasar pembandingan.

1. Analisis dan Perancangan Studi Kasus

Menganalisis sistem yang berjalan pada studi kasus dan membuat perancnagan perangkat lunak yang akan dibangun.

1. 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

Lalu melakukan pengujian perangkan lunak dengan metode Unit Testing.

1. Menarik kesimpulan dan saran

Menarik kesimpulan dari Tugas akhir yang telah dibuat, dan memberikan saran.

## 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 JSON Adapter Moshi dan Gson**

Pada bab ini berisi tentang Analisis dari JSON Adapter Moshi dan Gson yang nantinya  
akan dibandingkan performanya dari ke dua JSON Adapter tersebut.

**Bab 5 : 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 6 : 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 7 : 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.

# 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.

## 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]

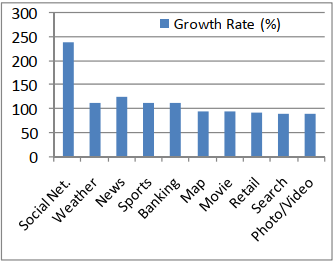
Sejumlah besar aplikasi *mobile* biasanya sudah diinstal sebelumnya di ponsel dan yang lainnya dapat diunduh oleh pengguna di internet dan menginstalnya di ponsel. Jumlah aplikasi *mobile* yang ada di market biasanya dibuat oleh berbagai macam *mobile application developer*, *publisher* dan *provider*. 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]

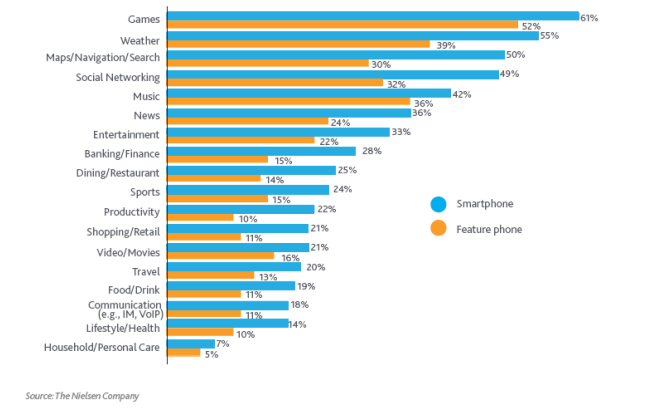
### Perkembangan Aplikasi Mobile

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.

## 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 berbagai macam 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]

### 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]

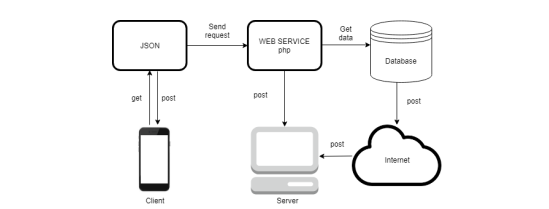
## Web Service & REST API

*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 sistem seperti 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 memodifikasinya dengan 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]

### 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**

## 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]

## 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]

## 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] :

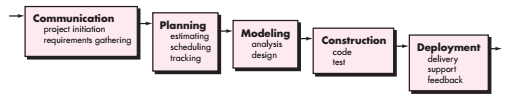
1. *Model*, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada  
   perubahan informasi. Hanya *model* yang mengandung data dan fungsi yang berhubungan  
   dengan pemrosesan data. Sebuah *model* meringkas lebih dari sekedar data dan fungsi yang  
   beroperasi di dalamnya. Pendekatan *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.

## 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]

### 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**

#### **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]

#### **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]

#### **Modelling**

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]

#### **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].

#### **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].

## 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 |
| 7. | Eksplorasi Bahasa Pemograman Kotlin Dalam Membangun Aplikasi Android (Studi Kasus : Aplikasi Berita Komunitas Pecinta Anime) | Septian Pratama [PRA19] | Aplikasi Berita Komunitas Pecinta Anime |

# 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.

## 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 Library   [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] |
| **Tahap 4 :** Analisis JSON Adapter Moshi dan Gson  **Hasil** : Pemahaman JSON Adapter Moshi dan Gson  **Kontribusi :** Berguna untuk memahami JSON Adapter Moshi dan Gson secara keseluruhan. | Menganalisis Json Adapter Moshi dan Gson | 1. Eksplorasi Bahasa Pemograman Kotlin Dalam Membangun Aplikasi Android (Studi Kasus : Aplikasi Berita Komunitas Pecinta Anime) [PRA19] 2. Eksplorasi Python Untuk Penyesuaian Modul Perkuliahan Dan Praktikum Di Teknik Informatika Unpas [NSM16] 3. Eksplorasi Python Menggunakan Framework Django (Studi Kasus : Sistem Perpustakaan Teknik Informatika UNPAS) [AYP16] |
| **Tahap 5 :** Melakukan perancangan perangkat lunak studi kasus  **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  2  2  Membuat pemodelan perilaku (Sequence Diagram |  |  | | --- | | **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 6 :** Pembuatan aplikasi, studi kasus  **Hasil** **:** Aplikasi yang sudah jadi  **Kontribusi :** Berguna untuk perbandingan kinerja 2 Json adapter | Pembuatan aplikasi  3 | 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 7 :** Perbandingan 2 Json Adapter  **Hasil : Hasil** perbandingan kinerja Gson dan Moshi  **Kontribusi :** Berguna untuk tahap pengambilan kesimpulan tugas akhir | 3  Pengujian aplikasi  Pencatatan hasil pengujian kedua JSON adapter | 1. Unit Testing [GUR20] |
| **Tahap 8 :** Penutup  **Hasil :** Kesimpulan dan saran tugas akhir  **Kontribusi : -** | Kesimpulan dan saran |  |

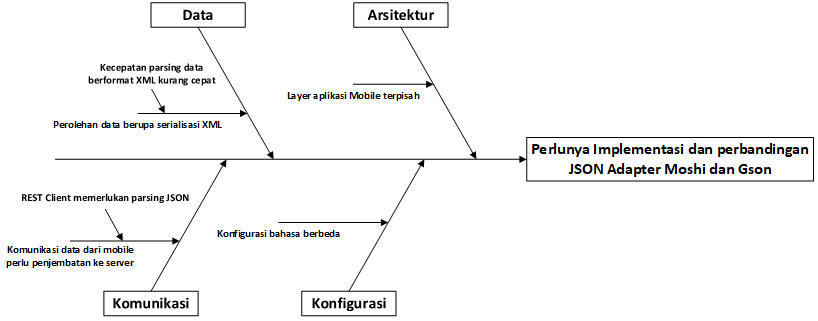
## Analisis Masalah dan Solusi Tugas Akhir

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. Fishbone Diagram 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 tiap Rest Client dan JSON adapter |

## 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.

### 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 |  |

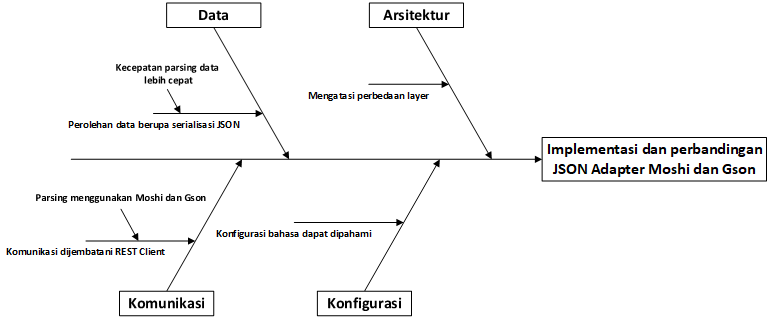
Rancangan analisis pada Table 3-3 dapat dijelaskan secara rinci pada Table 3-4

**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. |

### Kerangka Berfikir Teoritis

Pada kerangfka berpikir teoritis akan dibuat solusi berdasarkan permasalahan yang telah diketahui diatas. Berdasarkan permasalahan diatas dibutuhkan solusi berupa Implementasi dan perbandingan JSON Adapter Moshi dan Gson. untuk Parsing JSON dari REST Client yang dapat mengatasi permasalahan tersebut. Seperti pada gambar di bawah ini.



**Gambar 3‑2. Fishbone Diagram Solusi.**

## 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:

### 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.

#### **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]

### 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.

### JSON

JSON singkatan dari *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.

### 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.

### 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.*

## Profile Penelitian

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

### 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 informasi dan kegiatan 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.

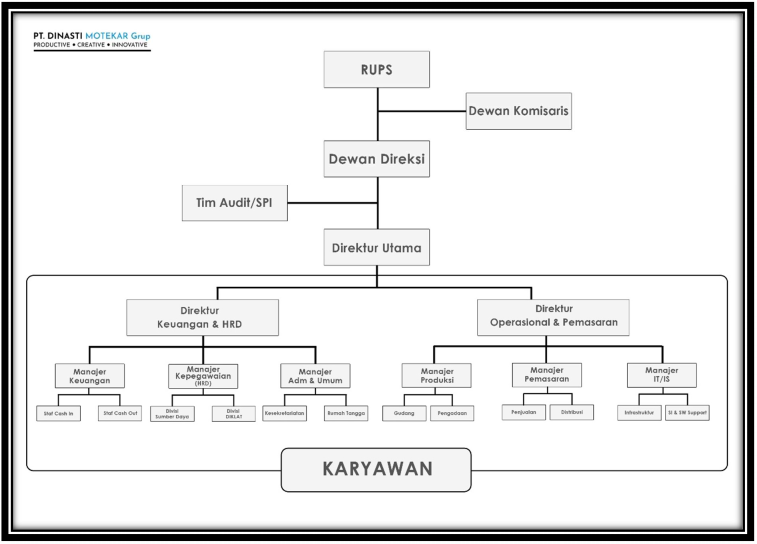
Pada penelitian ini fokus peneliti adalah membangun sebuah aplikasi berbasis android yang menampung semua informasi yang dibutuhkan dari sebuah website yang telah dibuat, agar mobilitas akses informasi meningkat, serta mengimplentasikan dan melakukan perbandingan JSON adapter pada aplikasi yang akan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

### 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.

### 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.

# ANALISIS JSON ADAPTER MOSHI DAN GSON

Pada bab ini berisi tentang Analisis dari JSON Adapter Moshi dan Gson yang nantinya akan dibandingkan performanya dari ke dua JSON Adapter tersebut.

## Analisis JSON Adapter Moshi

Moshi adalah pustaka JSON modern untuk Android dan Java. Membuatnya mudah untuk mengurai JSON menjadi objek Java: [SQU15]

String json = ...;

Moshi moshi = new Moshi.Builder().build();

JsonAdapter<BlackjackHand> jsonAdapter = moshi.adapter(BlackjackHand.class);

BlackjackHand blackjackHand = jsonAdapter.fromJson(json);

System.out.println(blackjackHand);

Dan bisa dengan mudah membuat serial objek Java seperti JSON:

BlackjackHand blackjackHand = new BlackjackHand(

new Card('6', SPADES),

Arrays.asList(new Card('4', CLUBS), new Card('A', HEARTS)));

Moshi moshi = new Moshi.Builder().build();

JsonAdapter<BlackjackHand> jsonAdapter = moshi.adapter(BlackjackHand.class);

String json = jsonAdapter.toJson(blackjackHand);

System.out.println(json);

### Tipe Adaptor Bawaan Moshi

Moshi memiliki dukungan bawaan untuk membaca dan menulis tipe data inti Java seperti :

* Primitif seperti int, float, char atau Integer, float, Character.
* Arrays, Collections, List, Sets dan Maps
* String
* Enumerasi

Yang mendukung kelas model dengan menuliskannya *field* per *field.* Contohnyapada kasus seperti diatas : [SQU15]

class BlackjackHand {

public final Card hidden\_card;

public final List<Card> visible\_cards;

...

}

class Card {

public final char rank;

public final Suit suit;

...

}

enum Suit {

CLUBS, DIAMONDS, HEARTS, SPADES;

}

Menjadi bentuk format JSON :

{

"hidden\_card": {

"rank": "6",

"suit": "SPADES"

},

"visible\_cards": [

{

"rank": "4",

"suit": "CLUBS"

},

{

"rank": "A",

"suit": "HEARTS"

}

]

}

### Kostum Tipe Adaptor

Dengan Moshi sangat mudah untuk menyesuaikan bagaimana nilai diubah ke JSON atau dari JSON yang tipe adaptornya memiliki anotasi method @ToJson dan @FromJson, Sebagai contoh standar pengkodean moshi dari bermain kartu menjadi bertele–tele, JSON mendefinisikan rank dan suit dalam bidang terpisah menjadi {“rank”:”A”,”suit”:”Hearts”}. Dengan kostum tipe adaptor kita dapat mengubah pengkodean menjadi sesuatu yang lebih ringkas menjadi “AH” untuk as hati atau “JD” untuk jack diamond. Berikut contoh kode dari kasus tersebut : [SQU15]

class CardAdapter {

@ToJson String toJson(Card card) {

return card.rank + card.suit.name().substring(0, 1);

}

@FromJson Card fromJson(String card) {

if (card.length() != 2) throw new JsonDataException("Unknown card: " + card);

char rank = card.charAt(0);

switch (card.charAt(1)) {

case 'C': return new Card(rank, Suit.CLUBS);

case 'D': return new Card(rank, Suit.DIAMONDS);

case 'H': return new Card(rank, Suit.HEARTS);

case 'S': return new Card(rank, Suit.SPADES);

default: throw new JsonDataException("unknown suit: " + card);

}

}

}

Contoh mendaftarkan tipe adaptor baru dengan Moshi.Builder

Moshi moshi = new Moshi.Builder()

.add(new CardAdapter())

.build();

Yang hasilnya menjadi :

{

"hidden\_card": "6S",

"visible\_cards": [

"4C",

"AH"

]

}

#### **Anotasi FromJson dan ToJson**

Method yang dianotasi dengan @FromJson tidak perlu menggunakan String sebagai argumen melainkan bisa mengambil input jenis apa saja dan moshi akan mengurai JSON terlebih dahulu ke objek jenis itu dan kemudian menggunakan method @FromJson untuk menghasilkan nilai akhir yang diinginkan, sebaliknya method yang dianotasi dengan @ToJson tidak harus menghasilkan String. Contohnya pada kasus mengurai tanggal dan waktu direpresentasikan menjadi dua string terpisah seperti dibawah ini : [SQU15]

{

"title": "Blackjack tournament",

"begin\_date": "20151010",

"begin\_time": "17:04"

}

Jika ingin menggabungkan dua field ini menjadi satu string untuk memfasilitasi penguraian tanggal dan memiliki nama variabel seperti biasanya maka kelas yang akan dibuat untuk diproduksi oleh moshi menjadi seperti di bawah ini :

class Event {

String title;

String beginDateAndTime;

}

Daripada secara manual memparsing JSON satu per satu, moshi dapat melakukan transformasi secara otomatis. Yang perlu dilakukan hanyalah membuat kelas lain yang sesuai secara langsung dengan struktur JSON nya seperti dibawah ini :

class EventJson {

String title;

String begin\_date;

String begin\_time;

}

Lalu membuat kelas lain dengan method @FromJSon dan ToJson yang memberitahukan moshi cara mengonversi kelas EventJson menjadi kelas Event dan sebaliknya, maka dengan begitu setiap kali meminta moshi untuk mengurai JSON ke kelas Event yang terjadi adalah moshi akan menguraikannya menjadi kelas EventJson sebagai Langkah perantara dan sebaliknya untuk membuat serialisasi kelas Event moshi akan membuat objek EventJson dan kemudian membuat serialisasi objek itu seperti biasa. Contoh kodenya seperti dibawah ini :

class EventJsonAdapter {

@FromJson Event eventFromJson(EventJson eventJson) {

Event event = new Event();

event.title = eventJson.title;

event.beginDateAndTime = eventJson.begin\_date + " " + eventJson.begin\_time;

return event;

}

@ToJson EventJson eventToJson(Event event) {

EventJson json = new EventJson();

json.title = event.title;

json.begin\_date = event.beginDateAndTime.substring(0, 8);

json.begin\_time = event.beginDateAndTime.substring(9, 14);

return json;

}

}

Dan jangan lupa mendaftarkan tipe adaptornya dengan moshi seperti di bawah ini :

Moshi moshi = new Moshi.Builder()

.add(new EventJsonAdapter())

.build();

Dengan begitu maka moshi dapat digunakan untuk memparsing JSON secara langsung ke kelas Event seperti dibawah ini :

JsonAdapter<Event> jsonAdapter = moshi.adapter(Event.class);

Event event = jsonAdapter.fromJson(json);

#### **Method Adapter**

Moshi menyediakan beberapa method untuk menunjang kenyamanan memakainya sebagai Json Adapter objek seperti :

* nullSafe()
* nonNull()
* lenient()
* failOnUnknown()
* indent()
* serializeNulls()

Method bawaan ini menggabungkan Json Adapter yang ada ke dalam fungsionalitas tambahan, misalnya jika memiliki adaptor yang tidak mendukung nilai null atau kosong maka bisa menggunakan fungsi nullSafe() untuk membuatnya aman dari null seperti dibawah ini : [SQU15]

String dateJson = "\"2018-11-26T11:04:19.342668Z\"";

String nullDateJson = "null";

// Hypothetical IsoDateDapter, doesn't support null by default

JsonAdapter<Date> adapter = new IsoDateDapter();

Date date = adapter.fromJson(dateJson);

System.out.println(date); // Mon Nov 26 12:04:19 CET 2018

Date nullDate = adapter.fromJson(nullDateJson);

// Exception, com.squareup.moshi.JsonDataException: Expected a string but was NULL at path $

Date nullDate = adapter.nullSafe().fromJson(nullDateJson);

System.out.println(nullDate); // null

Selain nullSafe() adapula nonNull() untuk membuat adapter menolak nilai null atau kosong.

### Parsing Array JSON

Moshi juga bisa digunakan untuk memparing JSON yang didalamnya sudah dalam bentuk array, sebagai contoh ada JSON array seperti dibawah ini :

[

{

"rank": "4",

"suit": "CLUBS"

},

{

"rank": "A",

"suit": "HEARTS"

}

]

Moshi dapat digunakan untuk memparsing JSON String menjadi List<Card> seperti dibawah ini :

String cardsJsonResponse = ...;

Type type = Types.newParameterizedType(List.class, Card.class);

JsonAdapter<List<Card>> adapter = moshi.adapter(type);

List<Card> cards = adapter.fromJson(cardsJsonResponse);

### Respon Fail

Moshi selalu menampilkan java.io.IOException standar jika ada kesalahan saat membaca dokumen JSON, atau jika format salah, dan akan menampilkan JsonDataException jika format data baik tetapi tidak sesuai dengan format yang diharapkan.

JsonDataException: Expected one of [CLUBS, DIAMONDS, HEARTS, SPADES] but was ANCHOR at path $.visible\_cards[2].suit

at com.squareup.moshi.JsonAdapters$11.fromJson(JsonAdapters.java:188)

at com.squareup.moshi.JsonAdapters$11.fromJson(JsonAdapters.java:180)

...

### Kostum Nama Field dengan @Json

Moshi bekerja paling baik saat objek JSON dan objek Java memiliki struktur yang sama, Tetapi jika tidak, Moshi memiliki anotasi untuk menyesuaikan pengikatan data, menggunakan @Json untuk spesifik memberikan nama field dengan nama Json Ini diperlukan jika nama JSON berisi spasi atau karakter lain yang tidak diizinkan dalam nama kolom Java. Misalnya, JSON ini memiliki nama bidang yang berisi spasi seperti dibawah ini :

{

"username": "jesse",

"lucky number": 32

}

Dengan @Json menjadi seperti dibawah ini :

class Player {

String username;

@Json(name = "lucky number") int luckyNumber;

...

}

Karena nama field JSON selalu ditentukan dengan field Java-nya, Moshi memudahkan untuk menemukan field saat menavigasi antara Java dan JSON.

### Tipe Adaptor Alternatif dengan @JsonQualifier

Gunakan @JsonQualifier untuk menyesuaikan cara suatu tipe dikodekan untuk beberapa field tanpa mengubah pengkodeannya di mana-mana, contoh dibawah ini adalah JSON dengan dua integer dan warna :

{

"width": 1024,

"height": 768,

"color": "#ff0000"

}

Sesuai Ketentuan, program Android juga menggunakan integer untuk warna menjadi seperti dibawah ini :

class Rectangle {

int width;

tinggi int ;

warna int ;

}

Tetapi jika kita menggunakan kode kelas Java di atas sebagai JSON, warnanya tidak dikodekan dengan benar, Cara mengatasinya adalah dengan menentukan anotasi qualifier, yaitu @JsonQualifier menjadi seperti di bawah ini :

@Retention ( RUNTIME )

@JsonQualifier

public @interface HexColor {

}

Selanjutnya, terapkan @HexColor ini ke field yang sesuai menjadi seperti dibawah ini :

class Rectangle {

int width;

int height;

@HexColor int color;

}

Setelah itu definisikan tipe adaptor untuk menanganinya menjadi seperti dibawah ini :

/\*\* Converts strings like #ff0000 to the corresponding color ints. \*/

class ColorAdapter {

@ToJson String toJson(@HexColor int rgb) {

return String.format("#%06x", rgb);

}

@FromJson @HexColor int fromJson(String rgb) {

return Integer.parseInt(rgb.substring(1), 16);

}

}

Gunakan @JsonQualifier jika memerlukan pengkodean JSON yang berbeda untuk jenis yang sama. Sebagian besar program seharusnya tidak memerlukan ini @JsonQualifier, tetapi sangat berguna bagi yang membutuhkannya.

### Penggunaan Transient

Beberapa model mendeklarasikan kolom yang tidak boleh disertakan dalam JSON. Misalnya, pada kelas dibawah ini kita memiliki field total :

public final class BlackjackHand {

private int total;

...

}

Secara default, semua kolom dikeluarkan saat encoding JSON, dan semua kolom diterima saat mendekode JSON, Cegah field dimasukkan dengan menambahkan transient menjadi seperti dibawah ini :

public final class BlackjackHand {

private transient int total;

...

}

Field dengan transient akan diabaikan saat penulisan JSON, bahkan saat membaca JSON maka field tersebut akan dilewati meskipun field tersebut memiliki nilai.

### Nilai & Constructor Default

Saat membaca JSON yang tidak memiliki field, Moshi mengandalkan Java atau Android runtime untuk menetapkan nilai field tersebut. Nilai mana yang digunakannya bergantung pada apakah kelas tersebut memiliki konstruktor tanpa argumen.

Jika kelas memiliki konstruktor tanpa argumen, Moshi akan memanggil konstruktor itu dan nilai apa pun yang ditetapkannya akan digunakan. Misalnya, karena kelas dibawah ini memiliki konstruktor tanpa argumen, maka field total diinisialisasi ke -1 :

public final class BlackjackHand {

private int total = -1;

...

private BlackjackHand() {

}

public BlackjackHand(Card hidden\_card, List<Card> visible\_cards) {

...

}

}

Jika kelas tidak memiliki konstruktor tanpa argumen, Moshi tidak dapat menetapkan nilai default field tersebut, meskipun itu ditentukan dalam deklarasi field. Sebaliknya, default field selalu 0 untuk angka, false untuk boolean, dan null untuk referensi. Dalam contoh ini, nilai default dari total adalah 0 :

public final class BlackjackHand {

private int total = -1;

...

public BlackjackHand(Card hidden\_card, List<Card> visible\_cards) {

...

}

}

Namun hal ini berpotensi menjadi sumber bug, untuk alasan ini, pertimbangkan untuk menentukan konstruktor tanpa argumen di kelas yang akan digunakan dengan Moshi, menggunakan @SuppressWarnings ("unused") untuk mencegahnya terhapus secara tidak sengaja nanti menjadi seperti dibawah ini :

public final class BlackjackHand {

private int total = -1;

...

@SuppressWarnings("unused") // Moshi uses this!

private BlackjackHand() {

}

public BlackjackHand(Card hidden\_card, List<Card> visible\_cards) {

...

}

}

### Menggabungkan Adaptor

Dalam beberapa situasi, konversi Java ke JSON default Moshi tidak cukup, dapat juga menggabungkan adaptor untuk mengembangkan konversi standar. Dalam contoh dibawah ini, mengubah serialize nulls, lalu mendelegasikan ke adaptor bawaan :

class TournamentWithNullsAdapter {

@ToJson void toJson(JsonWriter writer, Tournament tournament,

JsonAdapter<Tournament> delegate) throws IOException {

boolean wasSerializeNulls = writer.getSerializeNulls();

writer.setSerializeNulls(true);

try {

delegate.toJson(writer, tournament);

} finally {

writer.setLenient(wasSerializeNulls);

}

}

}

Ketika menggunakan ini untuk membuat serialisasi tournament, data null atau kosong akan ditulis tetapi data null di tempat lain akan dilewati JSON seperti biasa, Moshi memiliki sistem komposisi yang kuat di antarmuka JsonAdapter.Factory. bahkan dapat mengikuti proses encoding dan decoding untuk semua jenis, bahkan tanpa mengetahui jenisnya sebelumnya. Dalam contoh dibawah ini, menyesuaikan jenis beranotasi @AlwaysSerializeNulls, yang merupakan anotasi yang dibuat, bukan bawaan Moshi :

@Target(TYPE)

@Retention(RUNTIME)

public @interface AlwaysSerializeNulls {}

@AlwaysSerializeNulls

static class Car {

String make;

String model;

String color;

}

Setiap antarmuka JsonAdapter.Factory dipanggil oleh Moshi ketika ia perlu membuat adaptor untuk tipe pengguna. Factory ini mengembalikan adaptor untuk digunakan, atau null jika tidak berlaku untuk jenis yang diminta. Dalam kasus ini, mencocokkan semua kelas yang memiliki anotasi seperti di bawah ini :

static class AlwaysSerializeNullsFactory implements JsonAdapter.Factory {

@Override public JsonAdapter<?> create(

Type type, Set<? extends Annotation> annotations, Moshi moshi) {

Class<?> rawType = Types.getRawType(type);

if (!rawType.isAnnotationPresent(AlwaysSerializeNulls.class)) {

return null;

}

JsonAdapter<Object> delegate = moshi.nextAdapter(this, type, annotations);

return delegate.serializeNulls();

}

}

Setelah menentukan bahwa itu berhasil di apply, Factory akan mencari adaptor bawaan Moshi dengan memanggil Moshi.nextAdapter(), Ini adalah kunci dari mekanisme komposisi yaitu adaptor mendelegasikan satu sama lain, Komposisi dalam contoh ini sederhana: menerapkan transformasi serializeNulls() pada delegasi.

Adapun penulisan Adaptor ini bisa sangat canggih sebagai berikut :

* Adaptor dapat mengubah objek input sebelum dienkode JSON. Sebuah string bisa dipangkas atau dipotong, objek nilai dapat disederhanakan atau dinormalisasi.
* Adaptor dapat memperbaiki objek keluaran setelah didekodekan JSON. Itu bisa mengisi data yang hilang atau membuang data yang tidak diinginkan.
* JSON dapat diberi struktur tambahan, seperti membungkus nilai dalam objek atau array.

### Perbedaan dengan Gson

Moshi menggunakan mekanisme pengaliran dan pengikatan yang sama dengan Gson. Tetapi kedua libraries tersebut memiliki beberapa perbedaan penting diantaranya :

* Moshi memiliki lebih sedikit adaptor tipe bawaan. Misalnya, untuk mengkonfigurasi tanggal adapter harus dilakukan sendiri. Sebagian besar libraries yang mengikat akan meng encode apa pun yang diberikan padanya. Moshi menolak untuk membuat serialisasi jenis platform (java. \*, Javax. \*, Dan android. \*) Tanpa atipe adaptor yang disediakan pengguna. Ini dimaksudkan untuk mencegah mengunci diri secara tidak sengaja ke rilis JDK atau Android tertentu.
* Moshi kurang dapat dikonfigurasi. Tidak ada strategi penamaan field, pembuatan versi, pembuat instance, atau kebijakan serialisasi.
* Moshi tidak memiliki model JsonElement hanya menggunakan tipe bawaan seperti List dan Map.
* Tidak ada HTML-safe escaping. Gson mengenkode= sebagai \ u003d secara default sehingga dapat dienkode dengan aman dalam HTML tanpa pelolosan tambahan. Moshi mengkodekannya secara alami (as=) dan mengasumsikan bahwa encoder HTML - jika ada - akan melakukan tugasnya.

## Analisis Json Adapter 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 miliki kode sumbernya. [GOO08]

Gson awalnya dibuat untuk digunakan di dalam Google yang saat ini digunakan di sejumlah proyek. Sekarang digunakan oleh sejumlah proyek publik dan perusahaan.

### Tujuan Gson

Berikut adalah Goals atau tujuan dari Gson :

* Menyediakan mekanisme yang mudah digunakan seperti toString () dan konstruktor (factory method) untuk mengonversi Java ke JSON dan sebaliknya.
* Dapat digunakan untuk objek yang sudah ada dan tidak dapat dimodifikasi untuk dikonversi ke JSON dan dari JSON
* Dapat merepresentasi kustom untuk objek
* Mendukung artbitraily objek kompleks sewenang-wenang
* Menghasilkan keluaran JSON yang ringkas dan mudah dibaca [GSO20]

### Peforma dan Skalabilitan Gson

Berikut adalah beberapa metrik yang dipperoleh di desktop (dual opetron, RAM 8GB, Ubuntu 64-bit) yang menjalankan banyak hal lain bersama-sama dengan pengujian. test ini dapat dilakukan kembali jika ingin melakukannya dengan kelas PerformanceTest [GSO20]

* Strings: String deserialized lebih dari 25MB tanpa masalah
* Collection besar: Menyerialkan koleksi 1,4 juta objek
* Deserialisasi koleksi 87.000 objek
* Gson 1.4 menaikkan batas deserialization untuk byte array dan koleksi menjadi lebih dari 11MB dari 80KB.

### Menggunakan Gson

Kelas utama yang akan digunakan adalah Gson yang dapat dibuat dengan memanggil new Gson() . Tersedia juga kelas GsonBuilder yang dapat digunakan untuk membuat instance Gson dengan berbagai pengaturan seperti kontrol versi dan sebagainya.

Instance Gson tidak mempertahankan status apa pun saat menjalankan operasi Json. Jadi, bebas menggunakan kembali objek yang sama untuk beberapa serialisasi dan operasi deserialisasi Json.

Contoh penggunaan Variable Primitive : [GSO20]

(Serialisasi)

Gson gson = new Gson();

gson.toJson(1); ==> prints 1

gson.toJson("abcd"); ==> prints "abcd"

gson.toJson(new Long(10)); ==> prints 10

int[] values = { 1 };

gson.toJson(values); ==> prints [1]

(Deserialisasi)  
int one = gson.fromJson("1", int.class);  
Integer one = gson.fromJson("1", Integer.class);  
Long one = gson.fromJson("1", Long.class);  
Boolean false = gson.fromJson("false", Boolean.class);  
String str = gson.fromJson("\"abc\"", String.class);  
String anotherStr = gson.fromJson("[\"abc\"]", String.class);

Contoh Penggunaan Objek :

class BagOfPrimitives {  
  private int value1 = 1;  
  private String value2 = "abc";  
  private *transient* int value3 = 3;  
  BagOfPrimitives() {  
    // no-args constructor  
  }  
}

(Serialisasi)  
BagOfPrimitives obj = new BagOfPrimitives();  
Gson gson = new Gson();  
String json = gson.toJson(obj);   
**==> json is {"value1":1,"value2":"abc"}**

Perhatikan bahwa tidak dapat membuat serial objek dengan referensi melingkar karena akan mengakibatkan rekursi tak terbatas.

(Deserialization)

BagOfPrimitives obj2 = gson.fromJson(json, BagOfPrimitives.class);

==> obj2 is just like obj

Poin Lebih Baik dengan Objek :

* Tidak masalah dan disarankan menggunakan private fields
* Tidak perlu menggunakan anotasi apa pun untuk menunjukkan bahwa field akan disertakan untuk serialisasi dan deserialisasi. Semua field di kelas saat ini dan dari semua kelas super disertakan secara default.
* Jika field ditandai dengan transient, secara default akan diabaikan dan tidak disertakan dalam serialisasi atau deserialisasi JSON.
* Implementasi ini menangani null dengan benar
* Saat serialisasi, field null dilewati dari output
* Saat deserialisasi, entri yang hilang di JSON menyebabkan setelan field terkait di objek menjadi null
* Jika field sintetis, field tersebut akan diabaikan dan tidak disertakan dalam serialisasi atau deserialisasi JSON
* Bidang yang sesuai dengan kelas luar di kelas dalam, kelas anonim, dan kelas lokal diabaikan dan tidak termasuk dalam serialisasi atau deserialization

#### **Kelas dalam Kelas – Nested Classes**

Gson dapat membuat serial kelas statis didalam kelas dengan mudah, Gson juga dapat mendeserialisasi kelas statis didalam kelas. Namun, Gson tidak dapat secara otomatis melakukan deserialisasi kelas dalam kelas karena konstruktor no-args mereka juga memerlukan referensi ke Objek berisi yang tidak tersedia pada saat deserialisasi. masalah ini dapat diatasi dengan membuat kelas dalam menjadi statis atau dengan menyediakan InstanceCreator khusus untuknya. Berikut ini contohnya:

public class A {

public String a;

class B {

public String b;

public B() {

// No args constructor for B

}

}

}

Kelas B di atas tidak dapat (secara default) diserialkan dengan Gson.

Gson tidak bisa mendeserialisasi {"b": "abc"} menjadi instance B karena kelas B adalah kelas dalam. jika itu didefinisikan sebagai kelas B statis maka Gson akan dapat deserialisasi string. Solusi lain adalah menulis pembuat instance kustom untuk B, seperti dibawah ini dimungkinkan namun tidak direkomendasikan :

public class InstanceCreatorForB implements InstanceCreator<A.B> {

  private final A a;

  public InstanceCreatorForB(A a)  {

    this.a = a;

  }

  public A.B createInstance(Type type) {

    return a.new B();

  }

}

### Parsing Array JSON

Gson juga bisa digunakan untuk memparing JSON yang didalamnya sudah dalam bentuk array, seperti dibawah ini :

Gson gson = new Gson();

int[] ints = {1, 2, 3, 4, 5};

String[] strings = {"abc", "def", "ghi"};

(Serialization)

gson.toJson(ints);     **==> prints [1,2,3,4,5]**

gson.toJson(strings);  **==> prints ["abc", "def", "ghi"]**

(Deserialization)

int[] ints2 = gson.fromJson("[1,2,3,4,5]", int[].class);

**==> ints2 will be same as ints**

### Collections

Berikut ini adalah cara menggunakan Collection di Gson :

Gson gson = new Gson();

Collection<Integer> ints = Lists.immutableList(1,2,3,4,5);

(Serialization)

String json = gson.toJson(ints); **==> json is [1,2,3,4,5]**

(Deserialization)

Type collectionType = new TypeToken<Collection<Integer>>(){}.getType();

Collection<Integer> ints2 = gson.fromJson(json, collectionType);

Keterbatasan Collection :

* Dapat menserialisasi koleksi objek arbitrary tetapi tidak dapat melakukan deserialisasi darinya
* Karena tidak ada cara bagi pengguna untuk menunjukkan tipe objek yang dihasilkan
* Saat melakukan deserialisasi, Collection harus dari jenis generik tertentu

### Serializing dan deserializing tipe Generic

Saat memanggil toJson (obj), Gson memanggil obj.getClass() untuk mendapatkan informasi tentang field yang akan diserialisasikan. Demikian pula,hal ini biasanya dapat meneruskan objek MyClass.class dalam metode fromJson (json, MyClass.class). hal ini berfungsi dengan baik jika objek tersebut adalah tipe non-generik. Namun, jika objek bertipe generik, maka informasi tipe Generik hilang karena Java Type Erasure. Berikut adalah contoh yang menggambarkan hal tersebut:

class Foo<T> {

  T value;

}

Gson gson = new Gson();

Foo<Bar> foo = new Foo<Bar>();

gson.toJson(foo); // May not serialize foo.value correctly

gson.fromJson(json, foo.getClass()); // Fails to deserialize foo.value as Bar

Kode di atas gagal menafsirkan nilai sebagai tipe Bar karena Gson memanggil list.getClass() untuk mendapatkan informasi kelasnya, tetapi metode ini mengembalikan kelas Foo.class. Ini berarti Gson tidak memiliki cara untuk mengetahui bahwa ini adalah objek bertipe Foo <Bar>, dan bukan hanya Foo biasa.

Hal ini bisa diatasi dengan menspesifikasikan parameter untuk generic type nya, dengan menggunakan TypeToken Class seperti dibawah ini :

Type fooType = new TypeToken<Foo<Bar>>() {}.getType();

gson.toJson(foo, fooType);

gson.fromJson(json, fooType);

### Serializing and Deserializing Collection dengan tipe objek arbitrary

Terkadang dalam sebuah array JSON ada yang berisi array campuran seperti contoh di bawah ini :

['hello',5,{name:'GREETINGS',source:'guest'}]

Maka Collections nya berisi seperti ini :

Collection collection = new ArrayList();

collection.add("hello");

collection.add(5);

collection.add(new Event("GREETINGS", "guest"));

Dan kelas Eventnya menjadi seperti di bawah ini :

class Event {

  private String name;

  private String source;

  private Event(String name, String source) {

    this.name = name;

    this.source = source;

  }

}

Dapat juga membuat serialisasi Collection dengan Gson tanpa melakukan sesuatu yang spesifik seperti toJson (collection) akan menuliskan keluaran yang diinginkan.

Namun, deserialisasi dengan fromJson (json, Collection.class) tidak akan berfungsi karena Gson tidak memiliki cara untuk mengetahui cara memetakan input ke tipe, Gson mengharuskan memberikan versi umum dari jenis koleksi di fromJson. Jadi, memiliki tiga opsi seperti dibawah ini:

* Gunakan API parser Gson (low streaming parser atau DOM Parser JsonParser) untuk parsing elemen array, lalu gunakan Gson.fromJson() pada setiap elemen array.
* Daftarkan tipe adaptor untuk Collection.class yang melihat setiap anggota array dan memetakannya ke objek yang sesuai. Kerugian dari pendekatan ini adalah bahwa itu akan mengacaukan deserialization jenis koleksi lain di Gson.
* Daftarkan adaptor tipe untuk MyCollectionMemberType dan gunakan fromJson dengan Collection <MyCollectionMemberType> Pendekatan ini praktis hanya jika arrays muncul sebagai elemen tingkat atas atau jika dapat mengubah tipe bidang yang menampung koleksi menjadi dari tipe Collection <MyCollectionMemberType>.

### Serializers dan Deserializers bawaan

Gson memiliki serializers dan deserializers bawaan untuk kelas yang umum digunakan yang representasi defaultnya mungkin tidak sesuai. Berikut adalah daftar kelas tersebut:

1. java.net.URL untuk mencocokkannya dengan string seperti "http://code.google.com/p/google-gson/".
2. java.net.URI untuk mencocokkannya dengan string seperti "/ p / google-gson /"

### Kostum Serialization and Deserialization

Terkadang representasi default bukanlah yang diinginkan. Ini sering terjadi ketika berhadapan dengan library classes (DateTime, dll), Gson memungkinkan untuk mendaftarkan serializer dan deserializer kustom sendiri. Ini dilakukan dengan mendefinisikan dua bagian:

* Json Serialiers: Perlu menentukan serialisasi kustom untuk suatu objek
* Json Deserializers: Diperlukan untuk menentukan deserialisasi khusus untuk suatu jenis
* Pembuat Instance: Tidak diperlukan jika konstruktor no-args tersedia atau deserializer terdaftar

GsonBuilder gson = new GsonBuilder();

gson.registerTypeAdapter(MyType2.class, new MyTypeAdapter());

gson.registerTypeAdapter(MyType.class, new MySerializer());

gson.registerTypeAdapter(MyType.class, new MyDeserializer());

gson.registerTypeAdapter(MyType.class, new MyInstanceCreator());

Panggilan registerTypeAdapter memeriksa apakah tipe adaptor mengimplementasikan lebih dari satu antarmuka dan mendaftarkannya untuk semuanya.

#### **Menulis Serializer**

Berikut adalah contoh cara menulis serializer kustom untuk kelas JodaTime DateTime :

private class DateTimeSerializer implements JsonSerializer<DateTime> {

public JsonElement serialize(DateTime src, Type typeOfSrc,

JsonSerializationContext context) {

return new JsonPrimitive(src.toString());

}

}

Gson memanggil toJson() ketika itu berjalan ke objek DateTime selama serialisasi.

#### **Menulis Deserializer**

Berikut adalah contoh cara menulis serializer kustom untuk kelas JodaTime DateTime :

private class DateTimeDeserializer implements JsonDeserializer<DateTime> {

public DateTime deserialize(JsonElement json, Type typeOfT,

JsonDeserializationContext context)

throws JsonParseException {

return new DateTime(json.getAsJsonPrimitive().getAsString());

}

}

Gson memanggil fromJson() saat perlu deserialisasi fragmen string JSON menjadi objek DateTime

### Menulis Pembuat Instance

Saat Objek di deserialisasi, Gson perlu membuat instance default dari class, Kelas yang baik yang dimaksudkan untuk serialisasi dan deserialisasi harus memiliki konstruktor tanpa argument, Tidak masalah apakah public atau private class dan Biasanya, Pembuat Instance diperlukan saat berurusan dengan library class yang tidak menentukan konstruktor tanpa argument

#### **Contoh Pembuatan Instance**

private class MoneyInstanceCreator implements InstanceCreator<Money> {

public Money createInstance(Type type) {

return new Money("1000000", CurrencyCode.USD);

}

}

#### **InstanceCreator untuk tipe parameterized**

Terkadang tipe yang dicoba buat instantiate adalah tipe berparameter. Umumnya, ini bukan masalah karena instance sebenarnya adalah tipe mentah. Berikut ini contohnya:

class MyList<T> extends ArrayList<T> {

}

class MyListInstanceCreator implements InstanceCreator<MyList<?>> {

@SuppressWarnings("unchecked")

public MyList<?> createInstance(Type type) {

// No need to use a parameterized list since the actual instance will have the

raw type anyway.

return new MyList();

}

}

Namun, terkadang perlu membuat instance berdasarkan tipe parameter yang sebenarnya. Dalam kasus ini, dapat menggunakan parameter type yang diteruskan ke metode createInstance. Berikut ini contohnya:

public class Id<T> {

private final Class<T> classOfId;

private final long value;

public Id(Class<T> classOfId, long value) {

this.classOfId = classOfId;

this.value = value;

}

}

class IdInstanceCreator implements InstanceCreator<Id<?>> {

public Id<?> createInstance(Type type) {

Type[] typeParameters = ((ParameterizedType)type).getActualTypeArguments();

Type idType = typeParameters[0]; // Id has only one parameterized type T

return Id.get((Class)idType, 0L);

}

}

Dalam contoh di atas, sebuah instance dari kelas Id tidak dapat dibuat tanpa benar-benar meneruskan tipe sebenarnya untuk tipe berparameter. masalah ini dapat dipecahkan dengan menggunakan passed metode parameter. tipe objek dalam hal ini adalah representasi tipe berparameter Java dari Id <Foo> di mana instance sebenarnya harus terikat ke Id <Foo>. Karena kelas Id hanya memiliki satu parameter tipe berparameter, T, maka menggunakan elemen nol dari array tipe yang dikembalikan oleh getActualTypeArgument () yang akan menampung Foo.class dalam kasus ini.

### Dukungan Null Objek

Perilaku default yang diterapkan di Gson adalah bahwa field objek null diabaikan. Hal ini memungkinkan format keluaran yang lebih ringkas; namun, client harus menentukan nilai default untuk field ini karena format JSON diubah kembali ke Java-nya. Berikut adalah cara mengonfigurasi instance Gson untuk menghasilkan null:

Gson gson = new GsonBuilder().serializeNulls().create();

Inilah contoh penggunaan pada objek null :

public class Foo {  
  private final String s;  
  private final int i;  
  
  public Foo() {  
    this(null, 5);  
  }  
  
  public Foo(String s, int i) {  
    this.s = s;  
    this.i = i;  
  }  
}  
  
Gson gson = new GsonBuilder().serializeNulls().create();  
Foo foo = new Foo();  
String json = gson.toJson(foo);  
System.out.println(json);  
  
json = gson.toJson(null);  
System.out.println(json);  
  
======== OUTPUT ========  
{"s":null,"i":5}  
null

# ANALISIS DAN PERANCANGAN STUDI KASUS

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.

## Analisis Sistem yang sedang berjalan

Saat ini informasi tentang Studio Chendra PT Dinasti Motekar Grup bisa diakses di *website* studiochendra.co.id, dan semua informasi perihal Studio Chendra ada semua di website tersebut. Calon client biasanya mengakses website untuk melihat hasil pekerjaan dari unit usaha Dokumentasi ini lalu menghubungi pihak perusahaan untuk menyewa jasanya. Dalam hal ini untuk mengakses *website* memerlukan perangkat yang bisa membuka browser seperti desktop dan lainnya. Dengan adanya perangkat berbasis android, mobilitas akses calon client dapat ditingkatkan dengan hanya membuka aplikasinya.

## Analisis Kebutuhan

Analisis kebutuhan akan menghasilkan spesifikasi-spesifikasi dari karekteristik-karakteristik operasional yang akan dimiliki oleh perangkat lunak yang akan dikembangkan. Pada tahap ini penulis selaku analis sistem, melakukan interaksi dengan calon pengguna, yaitu dengan pemilik (*owner*) Studio Chendra Dinasti Motekar Grup.

### Menentukan Kebutuhan Dasar

Pada tahap analisis kebutuhan, harus ditentukan kebutuhan aplikasi yang akan dibangun. Hal-hal apa saja yang harus ada dan bisa dilakukan oleh aplikasi. Berikut ini adalah kebutuhan fungsional diantaranya sebagai berikut :

#### **Analisis Stakeholder**

Berikut ini adalah stakeholder-stakeholder atau orang yang berkepentingan yang terlibat dalam publikasi informasi dan kegiatan di Studio Chendra PT Dinasti Motekar Grup. Dapat dilihat pada Table 4-1 diantaranya sebagai berikut :

**Tabel 5‑1. Daftar Stakeholder**

|  |  |  |
| --- | --- | --- |
| **No.** | **Stakeholder** | **Deskripsi** |
| 1. | Administrator | Administrator adalah orang yang menerima permintaan Dokumentasi di Studio Chendra PT Dinasti Motekar Grup. |
| 2. | Pengguna | Pengguna adalah orang yang menggunakan aplikasi dalam hal ini adalah calon customer yang ingin menggunakan jasa Dokumentasi. |

#### **Analisis Fungsional**

Berikut ini adalah analisis fungsional yang dibutuhkan oleh aplikasi yang akan dibangun, dapat dilihat pada table 5-2.

**Tabel 5‑2. Analisis Fungsional**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kode** | **Kebutuhan Fungsional** | **Deskripsi** |
| 1. | F-01 | Perangkat lunak dapat menampilkan informasi Workshop berupa video atau foto | Perangkat lunak dapat menampilkan informasi mengenai workshop yang diselenggarakan oleh PT Dinasti Motekar Grup. |
| 2. | F-02 | Perangkat lunak melayani chatting dari customer ke customer service | Perangkat lunak memiliki fitur chatting yang menerima keluhan atau pertanyaan tentang sesuatu berkaitan dengan Studio Chendra PT Dinasti Motekar Grup. |
| 3. | F-03 | Perangkat lunak dapat menampilkan artikel fotografi | Perangkat lunak dapat memiliki kemampuan untuk menampilkan berita yang sedang hot topik yang berkaitan dengan jasa Dinasti Motekar Grup. |
| 4. | F-04 | Perangkat Lunak dapat menampilkan iklan layanan dokumentasi berupa video atau foto | Iklan atau Promosi layanan di PT Dinasti Motekar Grup dapat dilihat melalui perangkat lunak. |
| 5. | F-05 | Perangkat lunak dapat menampilkan portofolio gallery | Perangkat Lunak memiliki kemampuan untuk menampilkan portofolio gallery tentang hasil pekerjaan yang telah dilakukan oleh jasa di PT Dinasti Motekar Grup |
| 6. | F-06 | Perangkat lunak dapat menerima input registrasi order jasa dokumentasi | Perangkat lunak dapat menerima input customer yang ingin melakukan registrasi order jasa fotografi |

### Analisis Pengguna

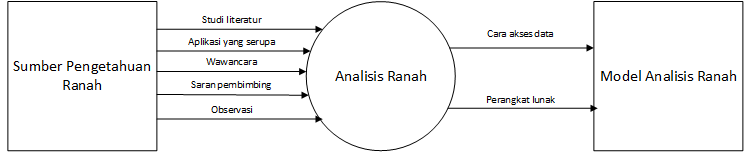
Analisis pengguna merupakan analisis yang digunakan untuk menentukan siapa saja user yang akan menggunakan aplikasi yang akan dibangun, dapat dilihat pada table 5-3.

**Tabel 5‑3. Daftar Pengguna**

|  |  |  |
| --- | --- | --- |
| **No.** | **User** | **Deskripsi** |
| 1. | Admin | Orang yang mengelola data dan informasi yang akan ditampilkan di aplikasi serta menerima order dari customer. |
| 2. | Customer | Orang yang melihat dan menggunakan aplikasi untuk melihat informasi serta memesan jasa fotografi |

### Analisis Ranah

Analisi ranah perangkat lunak adalah proses mengidentifikasi, menganalisis, dan menspesifikasi kebutuhan-kebutuhan umum dari suatu ranah perangkat lunak yang sifatnya spesifik biasanya agar suatu saat dapat digunakan pada berbagai proyek lain yang ada pada ranah aplikasi yang sama.



**Gambar 5‑1. Analisis Ranah**

Dibawah ini adalah deskripsi dari hasil output yang ada pada analisis ranah.

**Tabel 5‑4. Deskripsi hasil output analisis ranah**

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Output** | **Deskripsi** |
| 1. | Cara akses data | Hasil dari output cara akses data ini adalah menggunakan teknologi pertukaran data yaitu web service, dan JSON. |
| 2. | Perangkat Lunak | Hasil output dari perangkat lunak ini adalah dibangunnya perangkat lunak berbasis android untuk menampung informasi dan portofolio gallery. |

### Pendekatan untuk Pemodelan Spesifikasi Kebutuhan

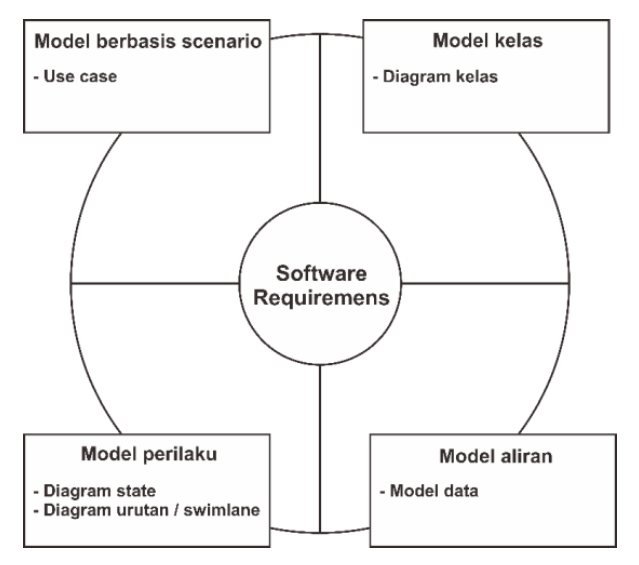
Suatu pandangan dari pemodelan spesifikasi-spesifikasi kebutuhan, yang dinamakan sebagai analisis tertruktur, memperlakukan data dan proses yang melakukan transformasi data tersebut segabai entitas yang terpisah. Jadi, pemodelan spesifikasi kebutuhan ini untuk memisahkan mana untuk keperluan data dan mana untuk proses. Pada sub bab ini akan dijelaskan mengenai perkakas yang  
digunakan dan menggambarkan dari spesifikasi kebutuhan yang akan dibuat.

#### **UML (Unified Modelling Process)**

Unified Modelling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Abstraksi konsep dasar UML terdiri dari structural classification, dynamic behavior, dan model management yang dapat dipahami main concepts sebagai term yang akan muncul pada saat membuat diagram dan view adalah kategori dari diagram tersebut. UML mendefinisikan diagram-diagram sebagai *Use case diagram, Class diagram, Statechart diagram, Activity diagram, Sequence diagram, Collaboration diagram, Component diagram, dan Deployment diagram*. Sebagian dari diagram-diagram itulah yang akan digunakan untuk pemodelan sistem atau perangkat lunak yang akan dibuat.

#### **Elemen – elemen Model Analisis**

Dibawah ini adalah elemen-elemen model analisis kebutuhan yang digunakan. Dapat dilihat pada gambar Gambar 5-2.



**Gambar 5‑2. Elemen Model Analisis**

Masing-masing unsur model kebutuhan pada Gambar 4-2 memperlihatkan permasalahan dari berbagai sudut pandang yang berbeda. Elemen-elemen berbasis skenario memperlihatkan bagaimana interaksi yang akan terjadi antara pengguna dengan sistem atau perangkat lunak yang akan dikembangkan. Elemen model berbasis kelas memodelkan objek-objek yang akan dimanipulasi oleh sistem, memodelkan operasi-operasi yang akan diterapkan pada objek-objek untuk melakukan manipulasi, memodelkan relasi (beberapa bersifat hierarkis) yang terjadi diantara objek satu dengan yang lainnya. Elemen-elemn perilaku memperlihatkan bagaimana event-event eksternal melakukan perubahan pada keadaan (state) sistem atau kelas-kelas yang ada didalamnya. Terakhir, elemen-elemen berorientasi aliran memperlihatkan sistem atau perangkat lunak yang bertindak sebagai pelaku transformasi informasi, memperlihatkan bagaimana objek-objek data ditansformasikan saat megnalir melintasi berbagai fungsi yang dimiliki sistem. [PRE10]

### Pemodelan Berbasis Skenario

Pemodelan berbasis skenario memperlihatkan bagaimana interaksi yang akan terjadi antara pengguna dengan sistem atau perangkat lunak yang akan dikembangkan. Pemodelan spesifikasi kebutuhan pengguna menggunakan UML (*Unified Modelling Language*) pada umumnya dimulai dengan pembuatan skenario-skenario dalam bentuk *use case-use case,* dan diagram *swimlane*. Diagram Use Case untuk aplikasi yang akan dibangun ada pada gambar 5-3 dibawah ini.

|  |
| --- |
| **Use Case Diagram** |
| **Gambar 5‑3. Use Case Diagram** |
| **Deskripsi** |
| Diagram Use Case ini digunakan untuk menggambarkan interaksi antara pengguna dan sistem atau perangkat lunak dimana aplikasi dapat menampilkan informasi workshop, menerima input pesan ke Customer Service, memperlihatkan artikel fotografi berupa foto dan video, menampilkan iklan layanan jasa dokumentasi, menampilkan portofolio gallery, juga dapat menerima input order jasa fotografi, yang semua itu dapat diakses oleh Customer, dan aplikasi dapat memposting video dan foto tentang workshop, menerima input jawaban dari Customer service, memposting video atau foto artikel fotografi, memposting iklan layanan jasa dokumentasi, memposting foto untuk portofolio gallery, menampilkan data registrasi order jasa dokumentasi yang dimana itu semua bisa diakses oleh admin. |

#### **Deskripsi Use Case**

Diagram Use Case pada Gambar 5-3 dapat dijelaskan secara lebih detail pada Tabel 5-5 dibawah ini.

**Tabel 5‑5. Deskripsi Use Case**

| **No.** | **Kode** | **Nama Use Case** | **Deskripsi** |
| --- | --- | --- | --- |
| 1. | UC-01 | Menampilkan informasi workshop berupa video atau foto | Use case ini digunakan untuk menampilkan informasi perihal workshop yang diselengarakan oleh PT Dinasti Motekar Grup berupa foto atau video. |
| 2. | UC-02 | Menginput pesan berupa teks untuk keluhan ke Customer Service | Use case ini digunakan untuk menerima input pesan berupa teks keluhan dari Customer. |
| 3. | UC-03 | Menampilkan artikel fotografi berupa foto atau video | Use case ini digunakan untuk menampilkan artikel berita tentang fotografi berupa foto atau video. |
| 4. | UC-04 | Menampilkan iklan layanan jasa dokumentasi berupa foto atau video | Use case ini digunakan untuk menampilkan ilklan jasa layanan dokumentasi di Studio Chendra PT Dinasti Motekar Grup berupa foto atau video. |
| 5. | UC-05 | Menampilkan portofolio gallery | Use case ini digunakan untuk menampilkan portofolio gallery jasa layanan dokumentasi di Studio Chendra PT Dinasti Motekar Grup. |
| 6. | UC-06 | Menginput registrasi order jasa dokumentasi | Use case ini digunakan untuk menerima input registrasi order jasa dokumentasi di Studio Chendra PT Dinasti Motekar Grup. |
| 7. | UC-07 | Memposting video atau foto workshop | Use Case ini digunakan untuk menerima posting video atau foto tentang workshop yang diselenggarakan di PT Dinasti Motekar Grup. |
| 8. | UC-08 | Menginput jawaban berupa teks untuk menjawab keluhan dari Customer | Use Case ini digunakan untuk menerima input pesan berupa teks jawaban dari Customer Service. |
| 9. | UC-09 | Memposting video atau foto artikel fotografi | Use case ini digunakan untuk menerima posting video atau foto tentang artikel fotografi. |
| 10. | UC-10 | Memposting video atau foto iklan layanan jasa dokumentasi | Use case ini digunakan untuk menerima posting video atau foto tentang iklan layanan jasa dokumentasi di Studio Chendra PT Dinasti Motekar Grup. |
| 11. | UC-11 | Memposting foto untuk portofolio gallery | Use case ini digunakan untuk menerima posting foto portofolio gallery di Studio Chendra PT Dinasti Motekar Grup. |
| 12. | UC-12 | Menampilkan data registrasi order jasa dokumentasi dari customer | Use case ini digunakan untuk menampilkan data registrasi order jasa dokumentasi Studio Chendra di PT Dinasti Motekar Grup. |

#### **Skenario Use Case**

Skenario *use case* merupakan gambaran proses berjalannya perangkat lunak berdasarkan *use case* yang telah didefinisikan. Skenario *use case* melibatkan aktor yang berinteraksi dengan *use case* tersebut serta aksi-reaksi yang terjadi antara aktor dengan sistem atau perangkat lunak. Berikut ini merupakan skenario *use case* aplikasi publikasi informasi dan kegiatan di Studio Chendra PT Dinasti Motekar Grup berbasis Android.

**Tabel 5‑6. Skenario Use Case menampilkan informasi workshop berupa foto atau video**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-01 | | | | |
| **Nama Use Case** | Menampilkan informasi workshop berupa foto dan video | | | | |
| **Tujuan** | Untuk menampilkan informasi workshop yang di selenggarakan di Dinasti Motekar Grup pada aplikasi berupa foto atau video. | | | | |
| **Aktor** | Customer | | | | |
| **Isi Use Case** | Lihat | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Customer masuk ke menu informasi workshop | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Customer masuk ke menu informasi workshop | | | | | 1. Sistem mengambil data informasi workshop |
| 1. Customer menekan informasi workshop yang diinginkan | | | | | 1. Sistem menampilkan informasi workshop berupa foto atau video |
|  | | | | | 1. Sistem menampilkan detail informasi workshop berupa foto atau video |
| **Kondisi Akhir** | | | Customer melihat detail informasi workshop berupa foto atau video | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Customer masuk ke menu informasi workshop | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Customer masuk ke menu informasi workshop | | | | 1. Sistem mengambil data informasi workshop | |
|  | | | | 1. Sistem menampilkan tidak ada workshop | |
| **Kondisi Akhir** | | Customer melihat pemberitahuan tidak ada informasi workshop | | | |

**Tabel 5‑7. Skenario Use Case menginput pesan berupa teks untuk keluhan ke Customer Service**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-02 | | | | |
| **Nama Use Case** | Menginput pesan berupa teks untuk keluhan ke Customer Service | | | | |
| **Tujuan** | Untuk menerima input pesan berupa teks dari customer yang akan disimpan dan dikirimkan ke customer service | | | | |
| **Aktor** | Customer | | | | |
| **Isi Use Case** | Tambah | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Customer masuk ke menu kontak customer service | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Customer masuk ke menu kontak customer service | | | | | 1. Sistem menampilkan tampilan chating |
| 1. Customer menginput pesan berupa teks | | | | | 1. Sistem mengecek input customer |
|  | | | | | 1. Sistem menyimpan input customer |
|  | | | | | 1. Sistem menampilkan pesan yang telah diinput di kolom chatting |
| **Kondisi Akhir** | | | Customer melihat pesan teks yang telah diinput di kolom chatting | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Customer masuk ke menu kontak customer service | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Customer masuk ke menu kontak customer service | | | | 1. Sistem menampilkan tampilan chating | |
| 1. Customer menginput pesan berupa teks | | | | 1. Sistem mengecek input customer | |
|  | | | | 1. Sistem menampilkan input teks tidak valid | |
| **Kondisi Akhir** | | Customer melihat pemberitahuan input teks tidak valid | | | |

**Tabel 5‑8. Skenario Use Case menampilkan artikel fotografi berupa foto atau video**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-03 | | | | |
| **Nama Use Case** | Menampilkan artikel fotografi berupa foto atau video | | | | |
| **Tujuan** | Untuk menampilkan artikel fotografi berupa foto atau video. | | | | |
| **Aktor** | Customer | | | | |
| **Isi Use Case** | Lihat | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Customer masuk ke menu artikel fotografi | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Customer masuk ke menu artikel fotografi | | | | | 1. Sistem mengambil data artikel fotografi |
| 1. Customer menekan artikel fotografi yang diinginkan | | | | | 1. Sistem menampilkan artikel fotografi berupa foto atau video |
|  | | | | | 1. Sistem menampilkan detail artikel fotografi berupa foto atau video |
| **Kondisi Akhir** | | | Customer melihat artilkel fotografi | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Customer masuk ke menu artikel | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Customer masuk ke menu artikel | | | | 1. Sistem mengambil data artikel fotografi | |
|  | | | | 1. Sistem menampilkan tidak ada artikel | |
| **Kondisi Akhir** | | Customer melihat pemberitahuan tidak ada artikel | | | |

**Tabel 5‑9. Skenario Use Case menampilkan iklan layanan jasa dokumentasi berupa foto atau video**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-04 | | | | |
| **Nama Use Case** | Menampilkan iklan layanan jasa dokumentasi berupa foto atau video | | | | |
| **Tujuan** | Untuk menampilkan iklan layanan jasa dokumentasi di Studio Chendra PT Dinasti Motekar Grup | | | | |
| **Aktor** | Customer | | | | |
| **Isi Use Case** | Lihat | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Customer masuk ke menu iklan layanan jasa | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Customer masuk ke menu iklan layanan jasa dokumentasi | | | | | 1. Sistem mengambil data iklan layanan jasa dokumentasi |
| 1. Customer menekan iklan layanan jasa dokumentasi yang diinginkan | | | | | 1. Sistem menampilkan iklan layanan jasa dokumentasi |
|  | | | | | 1. Sistem menampilkan detail iklan layanan jasa dokumentasi berupa foto atau video |
| **Kondisi Akhir** | | | Customer melihat detail iklan layanan jasa dokumentasi berupa foto atau video | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Customer masuk ke menu iklan layanan jasa | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Customer masuk ke menu iklan layanan jasa | | | | 1. Sistem mengambil data iklan layanan jasa dokumentasi | |
|  | | | | 1. Sistem menampilkan tidak ada layanan jasa | |
| **Kondisi Akhir** | | Customer melihat pemberitahuan tidak ada layanan jasa dokumentasi | | | |

**Tabel 5‑10. Skenario Use Case Menampilkan portofolio gallery**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-05 | | | | |
| **Nama Use Case** | Menampilkan portofolio gallery | | | | |
| **Tujuan** | Untuk menampilkan portofolio gallery tentang Studio Chendra PT Dinasti Motekar Grup | | | | |
| **Aktor** | Customer | | | | |
| **Isi Use Case** | Lihat | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Customer masuk ke menu portofolio gallery | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Customer masuk ke menu portofolio gallery | | | | | 1. Sistem mengambil data portofolio gallery |
| 1. Customer menekan foto yang diinginkan | | | | | 1. Sistem menampilkan portofolio gallery |
|  | | | | | 1. Sistem menampilkan detail foto |
| **Kondisi Akhir** | | | Customer melihat detail foto | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Customer masuk ke menu portofolio gallery | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Customer masuk ke menu portofolio gallery | | | | 1. Sistem mengambil data portofolio gallery | |
|  | | | | 1. Sistem menampilkan tidak ada foto | |
| **Kondisi Akhir** | | Customer melihat pemberitahuan tidak ada foto | | | |

**Tabel 5‑11. Skenario Use Case menginput registrasi order jasa dokumentasi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-06 | | | | |
| **Nama Use Case** | Menginput registrasi order jasa dokumentasi | | | | |
| **Tujuan** | Untuk menerima input dan menyimpan registrasi order jasa dokumentasi | | | | |
| **Aktor** | Customer | | | | |
| **Isi Use Case** | Lihat | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | 1. Customer masuk ke menu registrasi order 2. Customer menekan button order di detail iklan layanan jasa dokumentasi dan masuk menu registrasi order | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Customer masuk ke menu registrasi order | | | | | 1. Sistem menampilkan form registrasi order layanan jasa dokumentasi |
| 1. Customer menginput registrasi order | | | | | 1. Sistem mengecek input customer |
| 1. Customer menekan button registrasi | | | | | 1. Sistem menyimpan input customer |
|  | | | | | 1. Sistem menampilkan input berhasil |
| **Kondisi Akhir** | | | Customer melihat pemberitahuan registrasi order berhasil | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | 1. Customer masuk ke menu registrasi order 2. Customer menekan button order di detail iklan layanan jasa dokumentasi dan masuk menu registrasi order | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Customer masuk ke menu registrasi order | | | | 1. Sistem menampilkan form registrasi order layanan jasa dokumentasi | |
| 1. Customer menginput registrasi order | | | | 1. Sistem mengecek input customer | |
| 1. Customer menekan button registrasi | | | | 1. Sistem menampilkan input tidak valid | |
| **Kondisi Akhir** | | Customer melihat pemberitahuan input tidak valid | | | |

**Tabel 5‑12. Skenario Use Case Memposting video atau foto workshop**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-07 | | | | |
| **Nama Use Case** | Memposting video atau foto workshop | | | | |
| **Tujuan** | Untuk menambakan video atau foto workshop yang di selenggarakan di Studio Chendra PT Dinasti Motekar Grup | | | | |
| **Aktor** | Admin | | | | |
| **Isi Use Case** | Tambah | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Admin masuk ke menu tambah workshop | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Admin masuk ke menu tambah workshop | | | | | 1. Sistem menampilkan form tambah workshop |
| 1. Admin menginput informasi workshop | | | | | 1. Sistem mengecek input admin |
|  | | | | | 1. Sistem menyimpan input admin |
|  | | | | | 1. Sistem menampilkan input berhasil |
| **Kondisi Akhir** | | | Admin melihat pemberitahuan informasi Workshop telah berhasil ditambahkan | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Admin masuk ke menu tambah workshop | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Admin masuk ke menu tambah workshop | | | | 1. Sistem menampilkan form tambah workshop | |
| 1. Admin menginput informasi workshop | | | | 1. Sistem mengecek input admin | |
|  | | | | 1. Sistem menampilkan input tidak valid | |
| **Kondisi Akhir** | | Admin melihat pemberitahuan input tidak valid | | | |

**Tabel 5‑13. Skenario Use Case Menginput jawaban berupa teks untuk menjawab keluhan dari Customer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-08 | | | | |
| **Nama Use Case** | Menginput jawaban berupa teks untuk menjawab keluhan dari Customer | | | | |
| **Tujuan** | Untuk menerima input pesan berupa teks yang akan disimpan dan dikirimkan ke customer | | | | |
| **Aktor** | Admin | | | | |
| **Isi Use Case** | Tambah | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Admin masuk ke menu respond customer | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Admin masuk ke menu respond Customer | | | | | 1. Sistem mengambil data list chatting customer |
|  | | | | | 1. Sistem menampilkan tampilan list chatting customer |
| 1. Admin memilih customer | | | | | 1. Sistem menampilkan tampilan chating dengan customer |
| 1. Admin menginput pesan berupa teks | | | | | 1. Sistem mengecek input admin |
|  | | | | | 1. Sistem menyimpan input admin |
|  | | | | | 1. Sistem menampilkan pesan yang telah diinput di kolom chatting |
| **Kondisi Akhir** | | | Admin melihat pesan teks yang telah diinput di kolom chatting | | |
| **Skenario Alternatif 1** | | | | | |
| **Kondisi Awal** | | | Admin masuk ke menu respond customer | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Admin masuk ke menu respond Customer | | | | 1. Sistem mengambil data list chatting customer | |
|  | | | | 1. Sistem menampilkan tidak ada pesan | |
| **Kondisi Akhir** | | Admin melihat pemberitahuan tidak ada pesan dari customer | | | |
| **Skenario Alternatif 2** | | | | | |
| **Kondisi Awal** | | | Admin masuk ke menu respond customer | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Admin masuk ke menu respond Customer | | | | 1. Sistem mengambil data list chatting customer | |
|  | | | | 1. Sistem menampilkan tampilan list chatting customer | |
| 1. Admin memilih customer | | | | 1. Sistem menampilkan tampilan chating dengan customer | |
| 1. Admin menginput pesan berupa teks | | | | 1. Sistem mengecek input admin | |
|  | | | | 1. Sistem menampilkan input tidak valid | |
| **Kondisi Akhir** | | Admin melihat pemberitahuan input tidak valid | | | |

**Tabel 5‑14. Skenario Use Case Memposting video atau foto artikel fotografi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-09 | | | | |
| **Nama Use Case** | Memposting video atau foto artikel fotografi | | | | |
| **Tujuan** | Untuk menambakan video atau foto artikel yang berkaitan dengan fotografi | | | | |
| **Aktor** | Admin | | | | |
| **Isi Use Case** | Tambah | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Admin masuk ke menu tambah artikel fotografi | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Admin masuk ke menu tambah artikel fotografi | | | | | 1. Sistem menampilkan form tambah artikel fotografi |
| 1. Admin menginput informasi artikel fotografi | | | | | 1. Sistem mengecek input admin |
|  | | | | | 1. Sistem menyimpan input admin |
|  | | | | | 1. Sistem menampilkan input berhasil |
| **Kondisi Akhir** | | | Admin melihat pemberitahuan artikel telah berhasil ditambahkan | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Admin masuk ke menu tambah artikel | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Admin masuk ke menu tambah artikel | | | | 1. Sistem menampilkan form tambah artike; | |
| 1. Admin menginput informasi artikel fotografi | | | | 1. Sistem mengecek input admin | |
|  | | | | 1. Sistem menampilkan input tidak valid | |
| **Kondisi Akhir** | | Admin melihat pemberitahuan input tidak valid | | | |

**Tabel 5‑15. Skenario Use Case Memposting video atau foto iklan layanan jasa dokumentasi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-10 | | | | |
| **Nama Use Case** | Memposting video atau foto iklan layanan jasa dokumentasi | | | | |
| **Tujuan** | Untuk menambakan video atau foto iklan layanan jasa dokumentasi yang ada di Studio Chendra PT Dinasti Motekar Grup. | | | | |
| **Aktor** | Admin | | | | |
| **Isi Use Case** | Tambah | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Admin masuk ke menu tambah iklan layanan jasa dokumentasi | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Admin masuk ke menu tambah iklan layanan jasa dokumentasi | | | | | 1. Sistem menampilkan form tambah iklan layanan jasa dokumentasi |
| 1. Admin menginput informasi iklan layanan jasa dokumentasi | | | | | 1. Sistem mengecek input admin |
|  | | | | | 1. Sistem menyimpan input admin |
|  | | | | | 1. Sistem menampilkan input berhasil |
| **Kondisi Akhir** | | | Admin melihat pemberitahuan iklan layanan jasa dokumentasi telah berhasil ditambahkan | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Admin masuk ke menu tambah iklan layanan jasa dokumentasi | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Admin masuk ke menu tambah iklan layanan jasa | | | | 1. Sistem menampilkan form tambah iklan layanan jasa | |
| 1. Admin menginput informasi artikel fotografi | | | | 1. Sistem mengecek input admin | |
|  | | | | 1. Sistem menampilkan input tidak valid | |
| **Kondisi Akhir** | | Admin melihat pemberitahuan input tidak valid | | | |

**Tabel 5‑16. Skenario Use Case Memposting foto untuk portofolio gallery**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-11 | | | | |
| **Nama Use Case** | Memposting foto untuk portofolio gallery | | | | |
| **Tujuan** | Untuk menambakan foto yang akan ditampilkan di portofolio gallery | | | | |
| **Aktor** | Admin | | | | |
| **Isi Use Case** | Tambah | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Admin masuk ke menu tambah foto portofolio gallery | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Admin masuk ke menu tambah foto portofolio | | | | | 1. Sistem menampilkan form tambah foto portofolio |
| 1. Admin menginput informasi foto | | | | | 1. Sistem mengecek input admin |
|  | | | | | 1. Sistem menyimpan input admin |
|  | | | | | 1. Sistem menampilkan input berhasil |
| **Kondisi Akhir** | | | Admin melihat pemberitahuan foto telah berhasil ditambahkan | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Admin masuk ke menu tambah foto portofolio gallery | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Admin masuk ke menu tambah foto portofolio | | | | 1. Sistem menampilkan form tambah foto portofolio | |
| 1. Admin menginput foto | | | | 1. Sistem mengecek input admin | |
|  | | | | 1. Sistem menampilkan input tidak valid | |
| **Kondisi Akhir** | | Admin melihat pemberitahuan input tidak valid | | | |

**Tabel 5‑17. Skenario Use Case Menampilkan data registrasi order jasa dokumentasi dari customer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifikasi** | | | | | |
| **Nomor Use Case** | UC-12 | | | | |
| **Nama Use Case** | Menampilkan data registrasi order jasa dokumentasi dari customer | | | | |
| **Tujuan** | Untuk menampilkan data registrasi order yang telah dilakukan oleh customer | | | | |
| **Aktor** | Admin | | | | |
| **Isi Use Case** | Lihat | | | | |
| **Skenario Utama** | | | | | |
| **Kondisi Awal** | Admin masuk ke menu registrasi order jasa dokumentasi | | | | |
| **Aksi Aktor** | | | | | **Reaksi Sistem** |
| 1. Admin masuk ke menu registrasi order jasa dokumentasi | | | | | 1. Sistem mengambil data list registrasi order customer |
| 1. Admin memilih detail registrasi order customer yang diinginkan | | | | | 1. Sistem menampilkan list registrasi order customer |
|  | | | | | 1. Sistem menampilkan detail registrasi order customer |
| **Kondisi Akhir** | | | Admin melihat detail registrasi order customer | | |
| **Skenario Alternatif** | | | | | |
| **Kondisi Awal** | | | Admin masuk ke menu registrasi order jasa dokumentasi | | |
| **Aksi Aktor** | | | | **Reaksi Sistem** | |
| 1. Admin masuk ke menu registrasi order jasa dokumentasi | | | | 1. Sistem mengambil data list registrasi order customer | |
|  | | | | 1. Sistem menampilkan tidak ada registrasi order | |
| **Kondisi Akhir** | | Admin melihat pemberitahuan ridak ada registrasi order | | | |

#### **Aliran Aktivitas**

Aliran aktivitas digunakan untuk mengetahui alur dari aktivitas-aktivitas yang ada. Untuk membuat aliran aktivitas ini akan digunakan diagram *Swimlane*. Diagram *Swimlane* menjelaskan aliran aktivitas yang dijelaskan oleh use case dan pada saat yang sama menunjukkan aktor mana (jika ada beberapa aktor yang terlibat dalam use case tertentu) atau kelas analisis sebagai tanggung jawab atas tindakan yang dijelaskan oleh sebuah aktivitas [PRE10].

##### **Diagram Swimlane pada Use Case UC-01**

|  |
| --- |
| **Diagram Swimlane Use Case UC-01** |
| **Gambar 5‑4. Swimlane Diagram pada UC-01** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-01 yang dimulai dari memilih menu informasi workshop dan diakhiri dengan melihat detail informasi workshop berupa foto atau video. |

##### **Diagram Swimlane pada Use Case UC-02**

|  |
| --- |
| **Diagram Swimlane Use Case UC-02** |
| **Gambar 5‑5. Diagram Swimlane pada UC-02** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-02 yang dimulai dari memilih menu kontak customer service dan diakhiri dengan melihat pesan teks di kolom chatting. |

##### **Diagram Swimlane pada Use Case UC-03**

|  |
| --- |
| **Diagram Swimlane Use Case UC-03** |
| **Gambar 5‑6. Swimlane Diagram pada UC-03** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-03 yang dimulai dari memilih menu artikel dan diakhiri dengan melihat detail artikel fotografi berupa foto atau video. |

##### **Diagram Swimlane pada Use Case UC-04**

|  |
| --- |
| **Diagram Swimlane Use Case UC-04** |
| **Gambar 5‑7. Swimlane Diagram pada UC-04** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-04 yang dimulai dari memilih menu iklan layanan jasa dokumentasi dan diakhiri dengan melihat detail iklan layanan jasa dokumentasi berupa foto atau video. |

##### **Diagram Swimlane pada Use Case UC-05**

|  |
| --- |
| **Diagram Swimlane Use Case UC-05** |
| **Gambar 5‑8. Swimlane Diagram pada UC-05** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-05 yang dimulai dari memilih menu portofolio gallery dan diakhiri dengan melihat detail foto. |

##### **Diagram Swimlane pada Use Case UC-06**

|  |
| --- |
| **Diagram Swimlane Use Case UC-06** |
| **Gambar 5‑9. Swimlane Diagram pada UC-06** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-06 yang dimulai dari memilih menu registrasi order atau menekan button order di detail ilkan layanan jasa dokumentasi dan diakhiri dengan melihat pemberitahuan registrasi order berhasil. |

##### **Diagram Swimlane pada Use Case UC-07**

|  |
| --- |
| **Diagram Swimlane Use Case UC-07** |
| **Gambar 5‑10. Swimlane Diagram pada UC-07** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-07 yang dimulai dari memilih menu tambah workshopi dan diakhiri dengan melihat pemberitahuan workshop berhasil ditambahkan. |

##### **Diagram Swimlane pada Use Case UC-08**

|  |
| --- |
| **Diagram Swimlane Use Case UC-08** |
| **Gambar 5‑11. Swimlane Diagram pada UC-08** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-08 yang dimulai dari memilih menu respond customer dan diakhiri dengan melihat pesan di kolom chatting. |

##### **Diagram Swimlane pada Use Case UC-09**

|  |
| --- |
| **Diagram Swimlane Use Case UC-09** |
| **Gambar 5‑12. Swimlane Diagram pada UC-09** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-09 yang dimulai dari memilih menu tambah artikel dan diakhiri dengan melihat pemberitahuan artikel fotografi telah berhasil ditambahkan. |

##### **Diagram Swimlane pada Use Case UC-10**

|  |
| --- |
| **Diagram Swimlane Use Case UC-10** |
| **Gambar 5‑13. Swimlane Diagram pada UC-10** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-10 yang dimulai dari memilih menu tambah iklan layanan jasa dokumentasi dan diakhiri dengan melihat pemberitahuan iklan layanan jasa dokumentasi telah berhasil ditambahkan. |

##### **Diagram Swimlane pada Use Case UC-11**

|  |
| --- |
| **Diagram Swimlane Use Case UC-11** |
| **Gambar 5‑14. Swimlane Diagram pada UC-11** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-11 yang dimulai dari memilih menu tambah foto portofolio dan diakhiri dengan melihat pemberitahuan foto telah berhasil ditambahkan ke portofolio gallery. |

##### **Diagram Swimlane pada Use Case UC-12**

|  |
| --- |
| **Diagram Swimlane Use Case UC-12** |
| **Gambar 5‑15. Swimlane Diagram pada UC-12** |
| **Deskripsi** |
| Diagram Swimlane ini menggambarkan aliran aktivitas dari Use Case UC-12 yang dimulai dari memilih menu registrasi order jasa dokumentasi customer dan diakhiri dengan melihat detail registrasi order jasa dokumentasi customer. |

### Pemodelan data

Kebutuhan-kebutuhan perangkat lunak mencakup di dalamnya kebutuhan-kebutuhan untuk membuat, memperluas, atau bersinggungan dengan basis data. Untuk itu, harus didefinisikan semua objek data yang akan diproses di dalam sistem atau perangkat lunak, mendefinisikan relasi antar objek, dan mendefinisikan informasi lain terkait relasi. [PRE10]

#### **Objek Data**

Sebuah objek data sesungguhnya adalah representasi dari informasi komposit yang harus dipahami oleh perangkat lunak. Memiliki sejumlah property atau atribut yang berbeda. Suatu objek data dapat saja berupa entitas eksternal (sesuatu yang menghasilkan atau menggunakan informasi), laporan atau tampilan, panggilan telepon atau suatu alarm, suatu peran misalnya wiraniaga, suatu unit organisasi, suatu tempat, atau suatu struktur contohnya berkas, [PRE10]

**Tabel 5‑18. Objek Data**

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Objek** | **Deskripsi** |
|  | Menu | Objek ini digunakan untuk mengelola menu yang akan digunakan pada aplikasi |
|  | User | Objek ini digunakan untuk mengambil, menyimpan dan menampilkan akun dari user |
|  | Workshop | Objek ini digunakan untuk mengambil, menyimpan, dan menampilkan informasi Workshop |
|  | Chat Customer Service | Objek ini digunakan untuk mengambil, menyimpan, dan menampilkan chatting keluhan customer dan jawaban keluhan dari admin. |
|  | Artikel Fotografi | Objek ini digunakan untuk mengambil, menyimpan, dan menampilkan informasi artikel fotografi |
|  | Iklan Layanan Jasa Dokumenatsi | Objek ini digunakan untuk mengambil, menyimpan, dan menampilkan informasi iklan layanan jasa fotografi |
|  | Portofolio Gallery | Objek ini digunakan untuk mengambil, menyimpan, dan menampilkan portofolio gallery |
|  | Registrasi Order | Objek ini digunakan untuk mengambil, menyimpan, dan menampilkan informasi registrasi order layanan jasa dokumentasi |
|  | User Level | Objek ini digunakan untuk menyimpan data level-level pengguna yang ada. |
|  | Hak akses | Objek ini digunakan untuk menentukan hak akses setiap pengguna aplikasi. |

#### **Atribut Data**

Atribut data pada dasarnya mendefinisikan property-properti yang dimiliki oleh objek data dan pada saranya memiliki tiga karakteristik yang berbeda. Pertama, memberi nama objek dari objek data, mendeskripsikan objek,atau merujuk pada objek lain yang ada pada table yang lain. Berikut ini adalah daftar atribut data dari objek data yang sudah dibuat sebelumnya.

**Tabel 5‑19. Atribut data**

| **No.** | **Objek Data** | **Atribut Data** | **Deskripsi** |
| --- | --- | --- | --- |
|  | Menu | id\_menu | Atribut ini digunakan untuk primary key pada tabel. |
| title\_menu | Atribut ini digunakan untuk memberi nama pada menu. |
| url | Atribut ini digunakan untuk memberi nama url pada menu. |
| is\_mainmenu | Atribut ini digunakan untuk mengecek apakah menu adalah menu utama atau sub menu. |
| Is\_active | Atribut ini digunakan untuk menentukan menu ini aktif atau tidak. |
| order\_num | Atribut ini digunakan untuk Menyusun urutan menu. |
|  | User | Id\_user | Atribut ini digunakan untuk primary key pada tabel |
| nama\_user | Atribut ini digunakan untuk memberi nama user |
| username\_user | Atribut ini digunakan untuk menyimpan username. |
| password\_user | Atribut ini digunakan untuk autentikasi saat login |
| email\_user | Atribut ini digunakan untuk autentikasi saat login |
| no\_hp\_user | Atribut ini digunakan untuk menyimpan no hp. |
| Id\_user\_level | Atribut ini digunakan untuk menentukan level akses pada user. |
|  | Workshop | id\_workshop | Atribut ini digunakan untuk primary key pada tabel. |
| judul\_workshop | Atribut ini digunakan untuk, memberi nama judul workshop. |
| deskripsi\_workshop | Atribut ini digunakan untuk mendeskripsikan workshop |
| gambar\_workshop | Atribut ini digunakan untuk menambahkan gambar sebagai poster workshop |
| video\_workshop | Atribut ini digunakan untuk menambahkan video workshop. |
|  | Chat Customer Service | id\_chat | Atribut ini digunakan untuk primary key pada tabel. |
| id\_user\_1 | Atribut ini digunakan untuk mengetahui pengguna chat pertama untuk bertanya |
| id\_user\_2 | Atribut ini digunakan untuk mengetahui pengguna chat kedua untuk menjawab |
| Id\_user\_level | Atribut ini digunakan untuk mengetahui dan menentukan level akses user |
| teks\_chat\_1 | Atribut ini digunakan untuk menyimpan pesan teks pertanyaan pengguna pertama |
| teks\_chat\_2 | Atribut ini digunakan untuk menyimpan pesan teks jawaban pengguna kedua |
|  | Artikel Fotografi | id\_artikel | Atribut ini digunakan untuk primary key pada tabel. |
| judul\_artikel | Atribut ini digunakan untuk, memberi nama judul artikel. |
| deskripsi\_artikel | Atribut ini digunakan untuk mendeskripsikan artikel |
| gambar\_artikel | Atribut ini digunakan untuk menambahkan gambar artikel |
| video\_artikel | Atribut ini digunakan untuk menambahkan video artikel. |
|  | Iklan Layanan Jasa Dokumentasi | id\_iklan | Atribut ini digunakan untuk primary key pada tabel. |
| judul\_iklan | Atribut ini digunakan untuk, memberi nama judul iklan. |
| deskripsi\_iklan | Atribut ini digunakan untuk mendeskripsikan iklan |
| gambar\_iklan | Atribut ini digunakan untuk menambahkan gambar iklan |
| video\_iklan | Atribut ini digunakan untuk menambahkan video iklan. |
|  | Portofolio Gallery | id\_foto | Atribut ini digunakan untuk primary key pada tabel. |
| judul\_foto | Atribut ini digunakan untuk, memberi nama judul foto. |
| deskripsi\_foto | Atribut ini digunakan untuk mendeskripsikan foto |
| foto\_gallery | Atribut ini digunakan untuk menambahkan foto |
|  | Registrasi Order | id\_registrasi | Atribut ini digunakan untuk primary key pada tabel. |
| Id\_user | Atribut ini digunakan untuk, mengetahui profile user yang melakukan registrasi |
| id\_iklan | Atribut ini digunakan untuk mengetahui jasa layanan apa yang akan disewa |
| tgl\_registrasi | Atribut ini digunakan untuk mengetahui tanggal registrasi dilakukan |
|  | User Level | id\_user\_level | Atribut ini digunakan untuk primary key pada tabel |
| nama\_level | Atribut ini digunakan untuk memberi nama level untuk pemberian hak akses pada user |
|  | Hak Akses | id\_hak\_akses | Atribut ini digunakan untuk primary key pada tabel |
| id\_user\_level | Atribut ini digunakan untuk mengetahui level user |
| id\_menu | Atribut ini digunakan untuk menentukan menu apa saja yang bisa diakses |

#### **Relasi**

Objek-objek data saling terhubung satu dengan yang lainnya dengan menggunakan berbagai cara yang berbeda [PRE10].

**Tabel 5‑20. Relasi Objek Data**

| **No.** | **Objek Data** | **Relasi Data** | **Kardinalitas** | **Deskripsi** |
| --- | --- | --- | --- | --- |
|  | Chat Customer Service |  | one to many | Relasi ini digunakan untuk satu chat customer service dapat dilakukan banyak user |
|  | one to one | Relasi ini digunakan untuk satu chat customer service dapat memilik banyak user level |
|  | Hak Akses |  | one to one | Relasi ini digunakan untuk memberi satu hak pada satu menu |
|  | many to one | Relasi ini digunakan untuk satu user level memiliki banyak hak akses pada aplikasi |
|  | Registrasi Order |  | many to one | Relasi ini digunakan untuk satu user dapat Memiliki banyak registrasi order |
|  | one to many | Relasi ini digunakan untuk satu registrasi order dapat memiliki banyak Iklan Layanan Jasa Dokumentasi |
|  | User Level |  | one to many | Relasi ini digunakan untuk satu user level dapat memiliki banyak user |

### Pemodelan Berbasis Kelas

Pemodelan berbasis kelas pada dasarnya memperlihatkan objek-objek yang akan dimanipulasi oleh sistem atau perangkat lunak, memperlihatkan operasi-operasi yang akan diterapkan pada objekobjek untuk menghasilkan umbas tertentu pada manipulasi objek, memperlihatkan relasi-relasi antar objek [PRE10]. Pada pemodelan berbasis kelas ini akan mengidentifikasi kelas-kelas analisis kemudian menentukan atribut dan perilaku setiap kelas dan terakhir akan membuat diagram kelas.

#### **Mengidentifikasi Kelas – Kelas Analisis**

Proses mengidentifikasi kelas dengan cara memeriksa skenario penggunaan sistem atau perangkat lunak yang telah dikembangkan sebelumnya sebagai bagian dari model-model kebutuhan. [PRE10]

**Tabel 5‑21. Kelas - Kelas Analisis**

| **No.** | **Nama Kelas** | **Jenis** | **Deskripsi** |
| --- | --- | --- | --- |
|  | Activity\_workshop | View | Kelas ini digunakan untuk tampilan pada halaman workshop |
|  | Activity\_iklan\_layanan\_jasa | View | Kelas ini digunakan untuk tampilan pada halaman iklan layanan jasa |
|  | Activity\_portofolio\_gallery | View | Kelas ini digunakan untuk tampilan pada halaman portofolio gallery |
|  | Activity\_chat\_customer\_service | View | Kelas ini digunakan untuk tampilan pada halaman chat customer service |
|  | Activity\_artikel\_fotografi | View | Kelas ini digunakan untuk tampilan pada halaman artikel fotografi |
|  | Activity\_registrasi\_order | View | Kelas ini digunakan untuk tampilan pada halaman registrasi order |
|  | Activity\_menu | View | Kelas ini digunakan untuk tampilan pada menu |
|  | Activity\_login | View | Kelas ini digunakan untuk tampilan pada halaman login |
|  | Activity\_posting\_workshop | View | Kelas ini digunakan untuk tampilan pada halaman posting workshop |
|  | Activity\_posting\_iklan\_layanan\_jasa | View | Kelas ini digunakan untuk tampilan pada halaman posting iklan layanan jasa |
|  | Activity\_posting\_foto\_portofolio | View | Kelas ini digunakan untuk tampilan pada halaman posting foto portofolio |
|  | Activity\_respond\_customer | View | Kelas ini digunakan untuk tampilan pada halaman respond customer |
|  | Activity\_posting\_artikel\_fotografi | View | Kelas ini digunakan untuk tampilan pada halaman posting artikel fotografi |
|  | Activity\_add\_registrasi\_order | View | Kelas ini digunakan untuk tampilan pada halaman add registrasi order |
|  | WorkshopActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan workshop dengan model workshop |
|  | IklanLayananJasaActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan iklan layanan jasa dengan model iklan layanan jasa |
|  | PortofolioGalleryActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan portofolio gallery dengan model portofolio gallery |
|  | ChatCustomerServiceActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan chat customer service dengan model chat customer service |
|  | ArtikelFotografiActivity | Controller | Kelas ini digunakan sebagai controller untuk halaman artikel fotografi |
|  | RegistrasiOrderActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan registrasi order dengan model registrasi order |
|  | MenuActivity | Controller | Kelas ini digunakan sebagai controller untuk menu |
|  | LoginActivity | Controller | Kelas ini digunakan sebagai controller untuk halaman login |
|  | PostWorkshopActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan post workshop dengan model workshop |
|  | PostIklanLayananJasaActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan post iklan layanan jasa dengan model iklan layanan jasa |
|  | PostFotoPortofolioActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan post foto portofolio dengan model portofolio gallery |
|  | RespondCustomerActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan respond customer dengan model chat customer service |
|  | PostArtikelFotografiActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan porst artikel fotografi dengan model artikel fotografi |
|  | AddRegistrasiOrderActivity | Controller | Kelas ini digunakan untuk menghubungkan tampilan add registrasi order dengan model registrasi order |
|  | WorkshopModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data workshop |
|  | IklanLayananJasaModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data Iklan Layanan Jasa |
|  | PortofolioGalleryModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data Portofolio Gallery |
|  | ChatCustomerServiceModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data chat customer dengan admin |
|  | ArtikelFotografiModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data artikel fotografi |
|  | RegistrasiOrderModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data registrasi order |
|  | UserModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data user |
|  | MenuModel | Model | Kelas ini digunakan untuk berinteraksi dengan database dan mengambil atau menyimpan data menu |

#### **Menentukan Atribut-Atribut Kelas**

Atribut mendeskripsikan kelas-kelas yang telah terpilih untuk dimasukan dalam model spesifikasi kebutuhan sistem atau perangkat lunak. Atribut-atribut merupakan sesuatu yang mendefinikan kelas yang mengklasifikasikan makna suatu kelas dalam konteks ruang permasalahan yang telah diketahui sebelumnya.

**Tabel 5‑22. Atribut Kelas**

| **No.** | **Nama Kelas** | **Nama Atribut** |
| --- | --- | --- |
|  | Activity\_workshop | judulWorkshopText |
| deskripsiWorkshopText |
| images |
| video |
|  | Activity\_iklan\_layanan\_jasa | judulIklanText |
| deskripsiIklanText |
| images |
| video |
|  | Activity\_portofolio\_gallery | images |
|  | Activity\_chat\_customer\_service | userText |
| messageText |
| btnSend |
|  | Activity\_artikel\_fotografi | judulArtikelText |
| deskripsiArtikelText |
| images |
| video |
|  | Activity\_registrasi\_order | userText |
| judulIklanText |
|  | Activity\_menu | titleText |
| isActive |
| isMainMenu |
|  | Activity\_login | usernameText |
| passwordText |
| btnLogin |
|  | Activity\_posting\_workshop | judulWorkshopText |
| deskripsiWorkshopText |
| formImages |
| formVideo |
| btnPost |
|  | Activity\_posting\_iklan\_layanan\_jasa | judulIklanText |
| deskripsiIklanText |
| formImages |
| formVideo |
| btnPost |
|  | Activity\_posting\_foto\_portofolio | formImages |
| btnPost |
|  | Activity\_respond\_customer | userText |
| messageText |
| btnSend |
|  | Activity\_posting\_artikel\_fotografi | judulArtikelText |
| deskripsiArtikelText |
| formImages |
| formVideo |
| btnPost |
|  | Activity\_add\_registrasi\_order | userText |
| judulIklanText |
| btnConfirm |
|  | WorkshopActivity | - |
|  | IklanLayananJasaActivity | - |
|  | PortofolioGalleryActivity | - |
|  | ChatCustomerServiceActivity | - |
|  | ArtikelFotografiActivity | - |
|  | RegistrasiOrderActivity | - |
|  | MenuActivity | - |
|  | LoginActivity | username |
| password |
| btnLogin |
|  | PostWorkshopActivity | - |
|  | PostIklanLayananJasaActivity | - |
|  | PostFotoPortofolioActivity | - |
|  | RespondCustomerActivity | - |
|  | PostArtikelFotografiActivity | - |
|  | AddRegistrasiOrderActivity | - |
|  | WorkshopModel | idWorkshop |
| judulWorkshop |
| deskripsiWorkshop |
| gambarWorkshop |
| videoWorkshop |
|  | IklanLayananJasaModel | idIklan |
| judulIklan |
| deskripsiIklan |
| gambarIklan |
| videoIklan |
|  | PortofolioGalleryModel | idFoto |
| judulFoto |
| deskripsiFoto |
| foto |
|  | ChatCustomerServiceModel | idChat |
| idUser1 |
| idUser2 |
| message1 |
| message2 |
|  | ArtikelFotografiModel | idArtikel |
| judulArtikel |
| deskripsiArtikel |
| gambarArtikel |
| videoArtikel |
|  | RegistrasiOrderModel | idRegistrasi |
| idUser |
| idIklan |
| tglRegistrasi |
|  | UserModel | idUser |
| namaUser |
| usernameUser |
| passwordUser |
| emailUser |
| noHpUser |
| idUserLevel |
|  | MenuModel | idMenu |
| titleMenu |
| url |
| isMainMenu |
| isActive |
| orderNum |

#### **Menentukan Perilaku**

Operasi mendefinisikan perilaku suatu objek. Meskipun ada berbagai jenis operasi, mereka umumnya dapat dibagi menjadi empat kategori besar: operasi yang memanipulasi data dalam beberapa cara (misal, Menambahkan, menghapus, memformat ulang, memilih), operasi yang melakukan perhitungan, operasi yang menanyakan tentang keadaan suatu objek, dan operasi yang memantau objek untuk terjadinya peristiwa pengendalian [PRE10].

**Tabel 5‑23. Perilaku Kelas**

| **No.** | **Nama Kelas** | **Jenis Kelas** | **Operasi** | **Perilaku** |
| --- | --- | --- | --- | --- |
|  | WorkshopModel | Model | getData(id) | Mengambil data workshop dari database |
| getDataWorkshop() | Mengambil data workshop dan menghilangkan duplikasi data workshop |
| insert(data) | Menambahkan data workshop ke database |
|  | IklanLayananJasaModel | Model | getData(id) | Mengambil data iklan layanan jasa dari database |
| getDataIklanLayananJasa() | Mengambil data iklan layanan jasa dan menghilangkan duplikasi data. |
| insert(data) | Menambahkan data iklan layanan jasa ke database |
|  | PortofolioGalleryModel | Model | getData(id) | Mengambil data portofolio gallery dari database |
| getDataPortofolioGallery() | Mengambil data portofolio gallery dan menghilangkan duplikasi data. |
| insert(data) | Menambahkan data portofolio gallery ke database |
|  | ChatCustomerServiceModel | Model | getData(id) | Mengambil data chat customer service dari database |
| getDataChatCustomerService() | Mengambil data chat customer service dan menghilangkan duplikasi data. |
| insert(data) | Menambahkan data chat customer service ke database |
|  | ArtikelFotografiModel | Model | getData(id) | Mengambil data artikel fotografi dari database |
| getDataArtikelFotografi() | Mengambil data artikel fotografi dan menghilangkan duplikasi data. |
| insert(data) | Menambahkan data artikel fotografi ke database |
|  | RegistrasiOrderModel | Model | getData(id) | Mengambil data registrasi order dari database |
| getDataRegistrasiOrder() | Mengambil data registrasi order dan menghilangkan duplikasi data. |
| insert(data) | Menambahkan data registrasi order ke database |
|  | UserModel | Model | getData(id) | Mengambil data user dari database |
| getDataUser() | Mengambil data user dan menghilangkan duplikasi data. |
| insert(data) | Menambahkan data user ke database |
|  | MenuModel | Model | getData(id) | Mengambil data menu dari database |
| getDataMenu() | Mengambil data menu dan menghilangkan duplikasi data. |
| insert(data) | Menambahkan data menu ke database |
|  | WorkshopActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data workshop |
| read(id) | Menyediakan data detail workhshop yang diperoleh dari web service |
|  | IklanLayananJasaActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data iklan layanan jasa |
| read(id) | Menyediakan data detail iklan layanan jasa yang diperoleh dari web service |
|  | PortofolioGalleryActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data portofolio gallery |
| read(id) | Menyediakan data detail portofolio gallery yang diperoleh dari web service |
|  | ChatCustomerServiceActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data chat customer service |
| read(id) | Menyediakan data detail chat customer service yang diperoleh dari web service |
| add() | Menyediakan data yang diperlukan untuk view Activity\_chat\_customer\_service |
| addAction() | Menghubungkan view dengan web service untuk menambahkan data pesan teks |
| \_rules() | Menambahkan peraturan saat akan menambahkan data pesan teks |
|  | ArtikelFotografiActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data artikel fotografi |
| read(id) | Menyediakan data detail artikel fotografi yang diperoleh dari web service |
|  | RegistrasiOrderActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data registrasi order |
| read(id) | Menyediakan data detail registrasi order yang diperoleh dari web service |
|  | MenuActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data menu |
| read(id) | Menyediakan data detail menu yang diperoleh dari web service |
|  | LoginActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data yang dibuthkan untuk login |
| cekLogin() | Memeriksa keseuaian username dan password |
| logout() | Mengarahakn kembali ke view login |
|  | PostWorkshopActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data workshop |
| add() | Menyediakan data yang diperlukan untuk view Activity\_posting\_workshop |
| addAction() | Menghubungkan view dengan web service untuk menambahkan data workshop |
| \_rules() | Menambahkan peraturan saat akan menambahkan data workshop. |
|  | PostIklanLayananJasaActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data Iklan Layanan Jasa |
| add() | Menyediakan data yang diperlukan untuk view Activity\_posting\_iklan\_layanan\_jasa |
| addAction() | Menghubungkan view dengan web service untuk menambahkan data iklan layanan jasa |
| \_rules() | Menambahkan peraturan saat akan menambahkan data iklan layanan jasa |
|  | PostFotoPortofolioActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data portofolio gallery |
| add() | Menyediakan data yang diperlukan untuk view Activity\_posting\_foto\_portofolio |
| addAction() | Menghubungkan view dengan web service untuk menambahkan data foto portofolio gallery |
| \_rules() | Menambahkan peraturan saat akan menambahkan data portofolio gallery |
|  | RespondCustomerActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data chat customer service |
| read(id) | Menyediakan data detail chat customer service yang diperoleh dari web service |
| add() | Menyediakan data yang diperlukan untuk view Activity\_respond\_customer |
| addAction() | Menghubungkan view dengan web service untuk menambahkan data pesan teks |
| \_rules() | Menambahkan peraturan saat akan menambahkan data pesan teks |
|  | PostArtikelFotografiActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data artikel fotografi |
| add() | Menyediakan data yang diperlukan untuk view Activity\_posting\_artikel\_fotografi |
| addAction() | Menghubungkan view dengan web service untuk menambahkan data artikel fotografi |
| \_rules() | Menambahkan peraturan saat akan menambahkan data artikel fotografi |
|  | AddRegistrasiOrderActivity | Controller | index() | Menghubungkan view dengan web service untuk mengambil data registrasi order |
| add() | Menyediakan data yang diperlukan untuk view Activity\_add\_registrasi\_order |
| addAction() | Menghubungkan view dengan web service untuk menambahkan data registrasi order |
| \_rules() | Menambahkan peraturan saat akan menambahkan data registrasi order |
|  | Activity\_workshop | View | - | - |
|  | Activity\_iklan\_layanan\_jasa | View | - | - |
|  | Activity\_portofolio\_gallery | View | - | - |
|  | Activity\_chat\_customer\_service | View | - | - |
|  | Activity\_artikel\_fotografi | View | - | - |
|  | Activity\_registrasi\_order | View | - | - |
|  | Activity\_menu | View | - | - |
|  | Activity\_login | View | - | - |
|  | Activity\_posting\_workshop | View | - | - |
|  | Activity\_posting\_iklan\_layanan\_jasa | View | - | - |
|  | Activity\_posting\_foto\_portofolio | View | - | - |
|  | Activity\_respond\_customer | View | - | - |
|  | Activity\_posting\_artikel\_fotografi | View | - | - |
|  | Activity\_add\_registrasi\_order | View | - | - |

#### **Perancangan Kelas**

Perancangan kelas disini adalah merancang diagram kelas dari kelas-kelas yang telah didefinisikan sebelumnya beserta atribut dan perilakunya.

******

**Gambar 5‑16. Diagram Kelas**

Gambar diatas merupakan diagram kelas yang sudah dirancang dari mulai kelas-kelas analisis sampai penentuan perilaku. Class diagram diatas adalah rancangan untuk aplikasi yang menggunakan web service sebagai pertukaran datanya.

### Pemodelan Berbasis Perilaku

Model perilaku pada dasarnya menggambarkan bagaimana perangkat lunak akan berperilaku  
dalam menanggapi *event-event* yang datang dari arah luar atau bagaimana perangkat lunak akan berperilaku terhadap tindakan yang muncul dari luar [PRE10]. Pemodelan berbasis perilaku akan digambarkan dengan *sequence* diagram.

##### Diagram Sekuen Menampilkan infromasi workshop berupa Foto atau Video

|  |
| --- |
| **Menampilkan infromasi workshop berupa Foto atau Video** |
| **Gambar 5‑17. Diagram Sekuen Menampilkan infromasi workshop berupa Foto atau Video** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor customer dengan sistem dimana setelah memilih menu workhop maka sistem akan menampilkan data workshop. |

##### Diagram Sekuen Menginput pesan berupa teks untuk keluhan ke Customer Service

|  |
| --- |
| **Menginput pesan berupa teks untuk keluhan ke Customer Service** |
| **Gambar 5‑18. Diagram sekuen menginput pesan berupa teks untuk keluhan ke customer service** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor customer dengan sistem dimana setelah memilih menu menu kontak customer service sistem menampilkan data chat lalu customer menginput pesan teks maka sistem akan menyimpannya dan menampilkan pesan tersebut di kolom chat. |

##### Diagram Sekuen Menampilkan artikel fotografi berupa foto atau video

|  |
| --- |
| **Menampilkan artikel fotografi berupa foto atau video** |
| **Gambar 5‑19. Diagram Sekuen Menampilkan artikel fotografi berupa foto atau video** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor customer dengan sistem dimana setelah memilih menu artikel maka sistem akan menampilkan data artikel fotografi. |

##### Diagram Sekuen Menampilkan iklan layanan jasa dokumentasi berupa foto atau video

|  |
| --- |
| **Menampilkan iklan layanan jasa dokumentasi berupa foto atau video** |
| **Gambar 5‑20. Diagram Sekuen Menampilkan iklan layanan jasa dokumentasi berupa foto atau video** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor customer dengan sistem dimana setelah memilih menu iklan layanan jasa maka sistem akan menampilkan data iklan layanan jasa. |

##### Diagram Sekuen Menampilkan portofolio gallery

|  |
| --- |
| **Menampilkan portofolio gallery** |
| **Gambar 5‑21. Diagram Sekuen Menampilkan portofolio gallery** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor customer dengan sistem dimana setelah memilih menu portofolio gallery maka sistem akan menampilkan data portofolio gallery. |

##### Diagram Sekuen Menginput registrasi order jasa dokumentasi

|  |
| --- |
| **Menginput registrasi order jasa dokumentasi** |
| **Gambar 5‑22. Diagram Sekuen Menginput registrasi order jasa dokumentasi** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor customer dengan sistem dimana setelah memilih menu registrasi order maka sistem akan menampilkan form data registrasi order lalu customer menginput order dan menekan button konfirmasi maka sistem akan menyimpan dan menampilkan pemberitahuan order berhasil. |

##### Diagram Sekuen Memposting video atau foto workshop

|  |
| --- |
| **Memposting video atau foto workshop** |
| **Gambar 5‑23. Diagram Sekuen Memposting video atau foto workshop** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor admin dengan sistem dimana setelah memilih menu post workshop maka sistem akan menampilkan form data informasi workshop lalu admin menginput informasi workshop dan menekan button post maka sistem akan menyimpan dan menampilkan pemberitahuan workshop berhasil ditambahkan. |

##### Diagram Sekuen Menginput jawaban berupa teks untuk menjawab keluhan dari Customer

|  |
| --- |
| **Menginput jawaban berupa teks untuk menjawab keluhan dari Customer** |
| **Gambar 5‑24. Diagram Sekuen Menginput jawaban berupa teks untuk menjawab keluhan dari Customer** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor admin dengan sistem dimana setelah memilih menu menu kontak respond customer sistem menampilkan data chat lalu admin menginput pesan teks maka sistem akan menyimpannya dan menampilkan pesan tersebut di kolom chat. |

##### Diagram Sekuen Memposting video atau foto artikel fotografi

|  |
| --- |
| **Memposting video atau foto artikel fotografi** |
| **Gambar 5‑25. Diagram Sekuen Memposting video atau foto artikel fotografi** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor admin dengan sistem dimana setelah memilih menu post artikel fotografi maka sistem akan menampilkan form data informasi artikel fotografi lalu admin menginput informasi artikel fotografi dan menekan button post maka sistem akan menyimpan dan menampilkan pemberitahuan artikel fotografi berhasil ditambahkan. |

##### Diagram Sekuen Memposting video atau foto iklan layanan jasa dokumentasi

|  |
| --- |
| **Memposting video atau foto iklan layanan jasa dokumentasi** |
| **Gambar 5‑26. Diagram sekuen Memposting video atau foto iklan layanan jasa dokumentasi** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor admin dengan sistem dimana setelah memilih menu post iklan layanan jasa maka sistem akan menampilkan form data informasi iklan layanan jasa lalu admin menginput informasi iklan layanan jasa dan menekan button post maka sistem akan menyimpan dan menampilkan pemberitahuan iklan layanan jasa berhasil ditambahkan. |

##### Diagram Sekuen Memposting foto untuk portofolio gallery

|  |
| --- |
| **Memposting foto untuk portofolio gallery** |
| **Gambar 5‑27. Diagram Sekuen Memposting foto untuk portofolio gallery** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor admin dengan sistem dimana setelah memilih menu post foto portofolio maka sistem akan menampilkan form data foto lalu admin menginput foto dan menekan button post maka sistem akan menyimpan dan menampilkan pemberitahuan foto berhasil ditambahkan. |

##### Diagram Sekuen Menampilkan data registrasi order jasa dokumentasi dari customer

|  |
| --- |
| **Menampilkan data registrasi order jasa dokumentasi dari customer** |
| **Gambar 5‑28. Diagram Sekuan Menampilkan data registrasi order jasa dokumentasi dari customer** |
| **Deskripsi** |
| Diagram ini menggambarkan interaksi antara aktor admin dengan sistem dimana setelah memilih menu registrasi order customer maka sistem akan menampilkan data registrasi order customer |

## Perancangan Perangkat Lunak

Perancangan perangkat lunak pada dasarnya untuk menghasilkan suatu model atau representasiperangkat lunak yang memperlihatkan kekuatan, komoditi, serta kenyamanan. Selain itu, perancanganperangkat lunak merupakan tindakan rekayasa perangkat lunak yang terakhir didalam aktiviteaspemodelan dan kemuadian merupakan landasan yang sangat penting bagi aktivitas konstruksi [PRE10].Ada beberapa tahapan dalam perancangan perangkat lunak yaitu perancangan data, perancanganarsitektural, dan perancangan antarmuka.

### Perancangan Data

Perancangan data merupakan tahap mendetailkan dari tahap sebelumnya yaitu pemodelan data.  
Yang sudah mendefinisikan objek-objek data, atribut, dan relasi antar objek serta membuat *entity  
relationship diagram*. Berikut ini adalah perancangan data yang akan dibuat, dapat dilihat pada Table  
5-24.

**Tabel 5‑24. Perancangan Data**

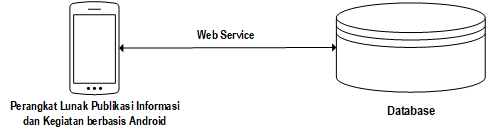
| **No.** | **Nama Objek** | **Atribut** | **Tipe Data** | **Length** |
| --- | --- | --- | --- | --- |
|  | Menu | id\_menu | int | 11 |
| title\_menu | varchar | 25 |
| url | varchar | 255 |
| is\_mainmenu | int | 11 |
| Is\_active | int | 11 |
| order\_num | int | 11 |
|  | User | Id\_user | int | 11 |
| nama\_user | varchar | 50 |
| username\_user | varchar | 30 |
| password\_user | varchar | 30 |
| email\_user | varchar | 50 |
| no\_hp\_user | int | 12 |
| id\_user\_level | int | 11 |
|  | Workshop | id\_workshop | int | 11 |
| judul\_workshop | varchar | 50 |
| deskripsi\_workshop | varchar | 255 |
| gambar\_workshop | text | - |
| video\_workshop | text | - |
|  | Chat Customer Service | id\_chat | int | 11 |
| id\_user\_1 | int | 11 |
| id\_user\_2 | int | 11 |
| Id\_user\_level | int | 11 |
| teks\_chat\_1 | varchar | 255 |
| teks\_chat\_2 | varchar | 255 |
|  | Artikel Fotografi | id\_artikel | int | 11 |
| judul\_artikel | varchar | 50 |
| deskripsi\_artikel | varchar | 255 |
| gambar\_artikel | text | - |
| video\_artikel | text | - |
|  | Iklan Layanan Jasa Dokumentasi | id\_iklan | int | 11 |
| judul\_iklan | varchar | 50 |
| deskripsi\_iklan | varchar | 255 |
| gambar\_iklan | text | - |
| video\_iklan | text | - |
|  | Portofolio Gallery | id\_foto | int | 11 |
| judul\_foto | varchar | 50 |
| deskripsi\_foto | varchar | 255 |
| foto\_gallery | text | - |
|  | Registrasi Order | id\_registrasi | int | 11 |
| Id\_user | int | 11 |
| id\_iklan | int | 11 |
| tgl\_registrasi | datetime | - |
|  | User Level | id\_user\_level | int | 11 |
| nama\_level | varchar | 25 |
|  | Hak Akses | id\_hak\_akses | int | 11 |
| id\_user\_level | int | 11 |
| id\_menu | int | 11 |

### Perancangan Arsitektural

Perancangan arsitektural merepresentasikan struktur data dan komponen-komponen program  
yang diperlukan untuk mengembangkan sistem-sitem berbasis komputer. Perancangan arsitekturalpadadasarnya mempertimbangkan gaya arsitektural yang akan diambil pada sistem danmempertimbangkanjuga hubungan-hubungan yang terjadi di antara semua komponen-komponen yangada dalam arsitektursistem [PRE10].

#### **Gaya Arsitektural**

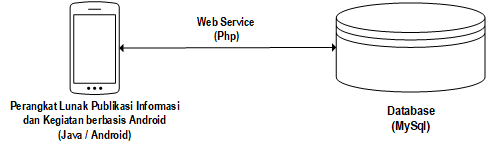
Arsitektural yang digunakan untuk aplikasi berbasis web dan *mobile* yaitu menggunakan gaya arsitektur berpusat pada data. Sebuah tempat penyimpanan data berapa pada pusat arsitektur dan diakses secara sering oleh komponen-komponen lainnya yang melakukan pembaruan, penambahan, penghapusan atau modifikasi data [PRE10].



**Gambar 5‑29. Gaya Arsitektur Berpusat pada data**

#### **Arsitektur Teknologi**

Arsitektur teknologi adalah teknologi-teknologi yang digunakan dalam pembuatan aplikasi berbasis Android. Berikut adalah teknologi yang digunakan untuk pembangunan aplikasi bisa dilihat pada Gambar 5-30.



**Gambar 5‑30. Arsitektur Teknologi**

### Perancangan Antarmuka Pengguna

Perancangan antarmuka menciptakan media komunikasi yang efektif diantara manusia dan komputer. Mengikuti sejumlah prinsip perancangan antarmuka, perancangan antarmuka pengguna bekerja dengan cara mengidentifikasi objek-objek antarmuka dan aksi-aksi dan kemudian membuat tata letak layar yang membentuk basis untuk pembuatan suatu prototype antarmuka pengguna [PRE10].

Berikut ini adalah antarmuka pengguna dari aplikasi yang akan dibangun menggunakan aplikasi Adobe XD.

##### Antarmuka menampilkan infromasi workshop berupa Foto atau Video

|  |
| --- |
| **Antarmuka menampilkan infromasi workshop berupa Foto atau Video** |
| **Gambar 5‑31. Antarmuka menampilkan informasi workshop berupa foto atau video** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk menampilkan informasi workshop berupa foto atau video yang dimana pada antarmuka itu memiliki dua tampilan, yang pertama untuk memilik workshop dan yang kedua untuk menampilkan detail foto atau video |

##### Antarmuka menginput pesan berupa teks untuk keluhan ke Customer Service

|  |
| --- |
| **Antarmuka menginput pesan berupa teks untuk keluhan ke Customer Service** |
| **Gambar 5‑32. Antarmuka menginput pesan berupa teks untuk keluhan ke Customer Service** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk menginput pesan berupa teks untuk keluhan ke Customer Service yang tampil Ketika memilih menu chat customer service. |

##### Antarmuka menampilkan artikel fotografi berupa foto atau video

|  |
| --- |
| **Antarmuka menampilkan artikel fotografi berupa foto atau video** |
| **Gambar 5‑33. Antarmuka menampilkan artikel fotografi berupa foto atau video** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk menampilkan artikel fotografi berupa foto atau video pada antarmuka itu memiliki dua tampilan, yang pertama untuk memilih artikel dan yang kedua untuk menampilkan detail foto atau video. |

##### Antarmuka menampilkan iklan layanan jasa dokumentasi berupa foto atau video

|  |
| --- |
| **Antarmuka menampilkan iklan layanan jasa dokumentasi berupa foto atau video** |
| **Gambar 5‑34. Antarmuka menampilkan iklan layanan jasa dokumentasi berupa foto atau video** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk menampilkan iklan layanan jasa dokumentasi berupa foto atau video pada antarmuka itu memiliki dua tampilan, yang pertama untuk memilih layanan dokumentasi dan yang kedua untuk menampilkan detail foto atau video. |

##### Antarmuka Menampilkan portofolio gallery

|  |
| --- |
| **Antarmuka menampilkan portofolio gallery** |
| **Gambar 5‑35. Antarmuka menampilkan portofolio gallery** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk menampilkan portofolio gallery pada antarmuka itu memiliki dua tampilan, yang pertama untuk memilih foto yang kedua untuk menampilkan detail foto. |

##### Antarmuka menginput registrasi order jasa dokumentasi

|  |
| --- |
| **Antarmuka menginput registrasi order jasa dokumentasi** |
| **Gambar 5‑36. Antarmuka menginput registrasi order jasa dokumentasi** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk menginput registrasi order jasa dokumentasi. |

##### Antarmuka Memposting video atau foto workshop

|  |
| --- |
| **Antarmuka Memposting video atau foto workshop** |
| **Gambar 5‑37. Antarmuka Memposting video atau foto workshop** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk memposting video atau foto workshop. |

##### Antarmuka menginput jawaban berupa teks untuk menjawab keluhan customer

|  |
| --- |
| **Antarmuka Menginput jawaban berupa teks untuk menjawab keluhan customer** |
| **Gambar 5‑38. Antarmuka Menginput jawaban berupa teks untuk menjawab keluhan customer** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk meninput jawaban berupa teks untuk menjawab keluhan customer. |

##### Antarmuka memposting video atau foto artikel fotografi

|  |
| --- |
| **Antarmuka memposting video atau foto artikel fotografi** |
| **Gambar 5‑39. Antarmuka memposting video atau foto artikel fotografi** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk memposting video atau foto artikel fotografi. |

##### Antarmuka memposting video atau foto iklan layanan jasa dokumentasi

|  |
| --- |
| **Antarmuka memposting video atau foto iklan layanan jasa dokumentasi** |
| **Gambar 5‑40. Antarmuka memposting video atau foto iklan layanan jasa dokumentasi** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk memposting video atau foto iklan layanan jasa dokumentasi. |

##### Antarmuka memposting foto untuk portofolio gallery

|  |
| --- |
| **Antarmuka memposting foto untuk portofolio gallery** |
| **Gambar 5‑41. Antarmuka memposting foto untuk portofolio gallery** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk memposting foto untuk portofolio gallery |

##### Antarmuka Menampilkan data registrasi order jasa dokumentasi dari customer

|  |
| --- |
| **Antarmuka Menampilkan data registrasi order jasa dokumentasi dari customer** |
| **Gambar 5‑42. Antarmuka Menampilkan data registrasi order jasa dokumentasi dari customer** |
| **Deskripsi** |
| Gambar diatas merupakan tampilan antarmuka untuk menampilkan data registrasi order jasa dokumentasi dari customer. Pada antarmuka ini ada dua tampilan yaitu yang pertama untuk memilih order customer dan yang kedua untuk menampilkan detail order. |

# KESIMPULAN DAN RENCANA TAHAP LANJUT

Bab ini menjelaskan mengenai kesimpulan yang didapatkan dari hasil penelitian yang dilakukan serta rencana tahap lanjut yang diberikan oleh penulis untuk pembangunan aplikasi.

## Kesimpulan

Berdasarkan hasil ekplorasi dan perancangan studi kasus yang telah dibuat maka kesimpulan yang dapat ditarik antara lain :

1. Moshi dibuat berdasarkan dari Gson sehingga cara penggunaannya hampir mirip
2. Moshi menyederhanakan perintah dari Gson dengan beberapa method baru seperti perintah @Json
3. Gson awalnya dibuat untuk digunakan di dalam Google yang saat ini digunakan di sejumlah proyek. Sekarang digunakan oleh sejumlah proyek publik dan perusahaan.
4. Implementasi Gson dan moshi hampir mirip namun dengan perintah yang berbeda yang di Gson perintah tersebut masih rumit, di Moshi perintah tersebut sudah disederhanakan.

## Rencana Tahap Selanjutnya

Berikut ini adalah rencana tahap selanjutnya yang akan penulis lakukan untuk menyelesaikan tugas akhir ini :

1. Melakukan Implementasi pembangunan perangkat lunak menggunakan Moshi dan Gson.
2. Melakukan Pengujian perangakat Lunak yang telah selesai dibangun dengan metode unit Testing untuk membandingkan Performa Json Adapter Moshi dan Gson.
3. Menarik kesimpulan dan saran tentang tugas akhir

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [ACE19]  [ANA17]  [AYP16]  [BUR19]  [CHE15]  [COD16]  [EAF19]  [GOO20]  [GOO08]  [GSO20]  [GUR20]  [INT19]  [JSO20]  [KEL95]  [INT15]  [KUC19]  [LIM15]  [NSM16]  [MAS15]  [IJE10]  [MUH19]  [OSA17]  [PRA19]  [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  Gson Google. (2020, 12 3). *Gson User Guide.* Retrieved from https://sites.google.com: site/gson/gson-user-guide  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  Pratama, S. (2019). *Eksplorasi Bahasa Pemograman Kotlin Dalam Membangun Aplikasi Android (Studi Kasus : Aplikasi Berita Komunitas Pecinta Anime).* Bandung: Universitas Pasundan.  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*. |