

## **BAB VI**

### **PENUTUP**

#### **6.1. Kesimpulan**

Berdasarkan analisis, desain dan implementasi perangkat lunak dan pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini yaitu perangkat lunak Sistem Layanan Berbasis Lokasi Jogja *Backpacker* (Joker) berhasil dibangun dengan menggunakan *framework* Google API yang dapat memberikan layanan dengan baik.

#### **6.2. Saran**

Saran yang dapat diambil dari proses analisis sampai pada pembuatan Joker pada pembuatan tugas akhir ini adalah sebagai berikut:

1. Dalam pengembangan lebih lanjut, sebaiknya dikembangkan sistem login yang memanfaatkan API dari aplikasi lain seperti Facebook dan Foursquare supaya pengguna tidak perlu memiliki banyak akun baru.
2. Mengembangkan aplikasi Joker dengan menambahkan fungsionalitas baru, seperti penggunaan kamera dalam menangkap gambar lokasi yang akan ditambahkan supaya pengguna dapat lebih mengetahui gambaran lokasi yang akan dikunjungi.

## DAFTAR PUSTAKA

- Agnew, Duncan., Larson, Kristine. 2010, *Finding the repeat times of the GPS constellation*, *Gps Solutions*, Volume 11, Number 1, Pages 71-76.
- Arifianto, Teguh., 2011, *Membuat Interface Aplikasi Android Lebih Keren dengan LWUIT*, Andi.
- Buliali, Joko Lianto., Suhadi Lili & Deddy Cahyadi, 2007, *Sistem Pencatatan Informasi Medis Berbasis Teknologi Microsoft .NET*. *Jurnal Informatika*, Vol 3, 97-118.
- Callaghan, MJ., Gormley, P., McBride, M., Harkin, J. McGinnity, TM. 2009, *Internal Location Based Services using Wireless Sensor Networks and RFID Technology*, *IJCSNS International Journal of Computer Science and Network Security*, VOL.6 No.4, page 108-113.
- Deitel, H.M., P.J. Deitel, 2002, *Wireless Internet & Mobile Business - How To Program*, Prentice Hall Inc.
- Deligiannis, Nikos., Louvros, Spiros. 2010, *Hybrid TOA-AOA Location Positioning Techniques in GSM Networks*, *Wireless Personal Communications*, Volume 54, Number 2, Pages 321-348.
- Hendrawan, Nikolaus Hendy Bimo, 2006, *Pembangunan Perangkat Lunak E-Commerce untuk Toko Merchandise Menggunakan ASP.NET*, Universitas Atma Jaya Yogyakarta, Yogyakarta
- Jan, Shau-Shiun., Hsu, Li-Ta., Tsai, Wen-Ming., 2010. *Development of an Indoor Location Based Service Test Bed and Geographic Information System with a Wireless Sensor Network*, *Sensor Journal*, Volume 10, Pages 2957-2974.
- Ku, Wei-Shinn., Chen, Yu., Zimmermann, Roger., 2009. *Privacy Protected Spatial Query Processing for Advanced Location Based Services*, *Wireless Personal Communications*, Volume 51, Number 1, Pages 53-65.

Küpper, Axel, 2005, *Location-Based Services: Fundamentals and Operation*, John Wiley & Sons.

Li, Mo., Sandrasegaran, Kumbesan., Tung, Tracy. 2009, *Trust-Assisted Handover Approach in Hybrid Wireless Networks*, *Wireless Personal Communications*, Volume 52, Number 3, Pages 477-499.

Li, Xiang., Zhang, Xihui., Lin, Hui. 2009, *Deriving network-constrained trajectories from sporadic tracking points collected in location-based services*, *Geo-Spatial Information Science*, Volume 12, Number 2, Pages 85-94.

Park, Kwangjin., Choo, Hyunseung. 2010, *Location-Based Data Dissemination for Spatial Queries in Wireless Broadcast Environments*, *Journal of Computer Science and Technology*, Volume 25, Number 2, Pages 330-346.

Parson, D., 2009. *Implementing a Map Based Simulator for The Location API for J2ME*, *Res. Left. Inf. Math. Sci*, Volume 7, Pages 157-160.

Petrova, Krassie., Wang, Bin., 2011, *Location-based Service Deployment and Demand: a Roadmap Model*, *Electronic Commerce Research*, Volume 11, Number 1, Pages 5-29.

Richards, Greg., Wilson Julie., 2004, *The Global Nomad: Backpacker Travel in Theory and Practice*, British Library.

Safaat, Nazruddin., 2011, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Informatika.

Sariana, Krisna, 2010, *Perancangan Sistem Informasi Berbasis Web Pada Kantor BPN (Badan Pertanahan Nasional) Sleman*, Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM, Yogyakarta.

Shiode, N., Li, C., Batty, M., Longley, P., Maguire, D., 2004. *The Impact and Penetration of Location Based Services*. In: Karimi, H. A., Hammad, A., ed. *Telegeoinformatics*. CRC Press, 349-366.

Schuurman, Nadine., Bell, Nathalie., L'Heureux, Randy., Hammed, Syed. 2009, *Modelling Optimal Location for Pre-hospital Helicopter Emergency Medical Services*, BMC Emergency Medicine, Volume 9, Number 1, Page 6.

Steiniger, Stefan, Moritz Neun dan Alistair Edwardes, 2006, *Foundations of Location Based Services, Lecture Notes on LBS*, V. 1.0.

Tiarasany, Meilyna Rakhmi, 2007, *Pembangunan Situs E-Commerce Produk Fashion dengan Menggunakan Teknologi .NET dan web services*, Universitas Atma Jaya Yogyakarta, Yogyakarta.

Virrantaus, K., Markkula, J., Garmash, A., Terziyan, Y.V., 2001, *Developing GIS-Supported Location-Based Services*, In: *Proc. of WGIS'2001 - First International Workshop on Web Geographical Information Systems.*, Kyoto, Japan. , 423-432.

Wang, Jiexian., Iz, Bâki H., Lu, Caiping. 2010, *Dependency of GPS positioning precision on station location*, Gps Solutions, Volume 6, Numbers 1-2, Pages 91-95.

Wijanarko, Sigit, 2009, *Aplikasi Mobile Ticketing pada Perusahaan Travel*, Universitas Atma Jaya Yogyakarta, Yogyakarta.

Wu, Chen., Congwei, Hu., Yongqi, Chen., Xiaoli, Ding., Kowk, Simon. 2009, *GPS rapid static and kinematic positioning based on GPS active network*, - *Geo-Spatial Information Science*, Volume 6, Number 1, Pages 5-1.

Yu, Shijun., Spaccapietra, Stefano. 2010, *A knowledge infrastructure for intelligent query answering in location-based services*, GeoInformatica, Volume 14, Number 3, Pages 379-404.

Yudistira, I.B. Putu Sandhi, 2007, *Pembangunan Aplikasi LBS Kota Yogyakarta pada Perangkat Mobile Berbasis J2ME*, Universitas Atma Jaya Yogyakarta, Yogyakarta..

# SKPL

## SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

**JOKER**

**(Jogja Backpacker)**

**Dipersiapkan oleh:**

**Yulius Dharmayanto / 08 07 05720**

**Program Studi Teknik Informatika - Fakultas Teknologi  
Industri**

**Universitas Atma Jaya Yogyakarta**

|                                                                                     |                                                                             |                     |  |             |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|---------------------|--|-------------|
|  | Program Studi<br>Teknik<br>Informatika<br>Fakultas<br>Teknologi<br>Industri | Nomor Dokumen       |  | Halaman     |
|                                                                                     |                                                                             | <b>SKPL - JOKER</b> |  | <b>1/26</b> |
|                                                                                     |                                                                             | Revisi              |  |             |

## DAFTAR PERUBAHAN

| Revisi   | Deskripsi |
|----------|-----------|
| <b>A</b> |           |
| <b>B</b> |           |
| <b>C</b> |           |
| <b>D</b> |           |
| <b>E</b> |           |
| <b>F</b> |           |

| INDEX<br>TGL       | -       | A | B | C | D | E | F | G |
|--------------------|---------|---|---|---|---|---|---|---|
| Ditulis<br>oleh    | YD      |   |   |   |   |   |   |   |
| Diperik<br>sa oleh | KA, BYD |   |   |   |   |   |   |   |
| Disetuj<br>ui oleh | KA, BYD |   |   |   |   |   |   |   |

## Daftar Halaman Perubahan

| Halaman | Revisi | Halaman | Revisi |
|---------|--------|---------|--------|
|         |        |         |        |

## Daftar Isi

|                                                               |           |
|---------------------------------------------------------------|-----------|
| <b>1 PENDAHULUAN .....</b>                                    | <b>6</b>  |
| 1.1 Tujuan .....                                              | 6         |
| 1.2 Lingkup Masalah .....                                     | 6         |
| 1.3 Definisi, Akronim dan Singkatan .....                     | 7         |
| 1.4 Referensi .....                                           | 8         |
| 1.5 Deskripsi umum (Overview) .....                           | 8         |
| <b>2 DESKRIPSI KEBUTUHAN .....</b>                            | <b>9</b>  |
| 2.1 Perspektif produk .....                                   | 9         |
| 2.2 Fungsi Produk .....                                       | 10        |
| 2.3 Karakteristik Pengguna .....                              | 13        |
| 2.4 Batasan-batasan .....                                     | 13        |
| 2.5 Asumsi dan Ketergantungan .....                           | 13        |
| <b>3 KEBUTUHAN KHUSUS .....</b>                               | <b>14</b> |
| 3.1 Kebutuhan antarmuka eksternal .....                       | 14        |
| 3.1.1 Antarmuka pemakai .....                                 | 14        |
| 3.1.2 Antarmuka perangkat keras .....                         | 14        |
| 3.1.3 Antarmuka perangkat lunak .....                         | 14        |
| 3.1.4 Antarmuka Komunikasi .....                              | 15        |
| 3.2 Kebutuhan fungsionalitas Perangkat Lunak .....            | 16        |
| 3.2.1 Use Case Diagram .....                                  | 16        |
| <b>4 SPESIFIKASI RINCI KEBUTUHAN .....</b>                    | <b>17</b> |
| 4.1 Spesifikasi Kebutuhan Fungsionalitas .....                | 17        |
| 4.1.1 Use case Spesification: Login Administrator .....       | 17        |
| 4.1.2 Use case Spesification: Pengelolaan Data Venue .....    | 18        |
| 4.1.3 Use case Spesification: Pencarian Posisi Sekarang ..... | 19        |
| 4.1.4 Use case Spesification: Login User .....                | 21        |
| 4.1.5 Use case Spesification: Lihat Detail Venue .....        | 22        |
| 4.1.6 Use case Spesification: Tampil Navigasi Venue .....     | 23        |
| 4.1.7 Use case Spesification : Tambah Komentar .....          | 24        |
| <b>5 ENTITY RELATIONSHIP DIAGRAM (ERD) .....</b>              | <b>26</b> |

## Daftar Gambar

|                                                 |    |
|-------------------------------------------------|----|
| Gambar 1. Arsitektur Perangkat lunak JOKER..... | 10 |
| Gambar 2. Use Case Diagram.....                 | 16 |
| Gambar 3. Entity Relationship Diagram.....      | 26 |



## **1 Pendahuluan**

### **1.1 Tujuan**

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak JOKER(Jogja Backpacker) untuk mendefinisikan kebutuhan perangkat lunak yang akan dikembangkan meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-JOKER ini juga mendefinisikan batasan perancangan perangkat lunak.

Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

### **1.2 Lingkup Masalah**

Perangkat Lunak JOKER dikembangkan dengan tujuan untuk :

1. Menangani pencarian posisi pengguna.
2. Menangani pencarian lokasi tempat wisata, tempat penginapan, tempat makan, dan tempat mendapatkan transportasi umum dari posisi pengguna.
3. Menangani penyimpanan data lokasi tempat wisata, tempat penginapan, tempat makan, dan tempat mendapatkan transportasi umum.
4. Menangani informasi seputar tempat wisata, tempat penginapan, tempat makan, dan tempat mendapatkan transportasi umum.

|                                                                                                                                                                                                                    |            |       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-------|
| Program Studi Teknik Informatika                                                                                                                                                                                   | SKPL-JOKER | 6/ 26 |
| Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika |            |       |

5. Menangani navigasi rute perjalanan dari posisi pengguna hingga tempat tujuan.

### 1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

| Keyword/Phrase | Definisi                                                                                                                                                                                                                    |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SKPL           | Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.                                                                                                                                                |
| SKPL-JOKER-XXX | Kode yang merepresentasikan kebutuhan pada JOKER (Jogja Backpacker) dimana XXX merupakan nomor fungsi produk.                                                                                                               |
| JOKER          | Perangkat lunak untuk menyediakan informasi dan memeberikan rute perjalanan di sekitar wilayah Daerah Istimewa Yogyakarta.                                                                                                  |
| LBS            | LBS (Location Based Service) adalah suatu bentuk layanan informasi yang diakses oleh perangkat mobile melalui jaringan mobile dan memiliki kemampuan untuk memakai posisi geografis dari perangkat mobile tersebut          |
| Location       | Posisi spesifik dari suatu ruang secara fisik                                                                                                                                                                               |
| Internet       | Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e- |

|               |                                                                                   |
|---------------|-----------------------------------------------------------------------------------|
|               | mail, FTP, dan <i>World Wide Web</i> .                                            |
| Server        | Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan. |
| <i>Client</i> | Komputer yang menghubungi <i>server</i> melalui jaringan                          |

#### 1.4 Referensi

Referensi yang digunakan pada pembangunan perangkat lunak tersebut adalah:

1. Dharmayanto Yulius, Spesifikasi Kebutuhan Perangkat Lunak Asset Manajemen System (SKPL-AMS) , Jurusan Teknik Informatika UAJY, 2011.
2. Hendradjaya Bayu, Panduan Penulisan Spesifikasi Kebutuhan Perangkat Lunak (SKPL), Jurusan Teknik Informatika ITB.
3. Sapta Juli, Spesifikasi Kebutuhan Perangkat Lunak SC3, Universitas Atma Jaya Yogyakarta, 2006.

#### 1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 2 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak JOKER yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam

penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak JOKER tersebut.

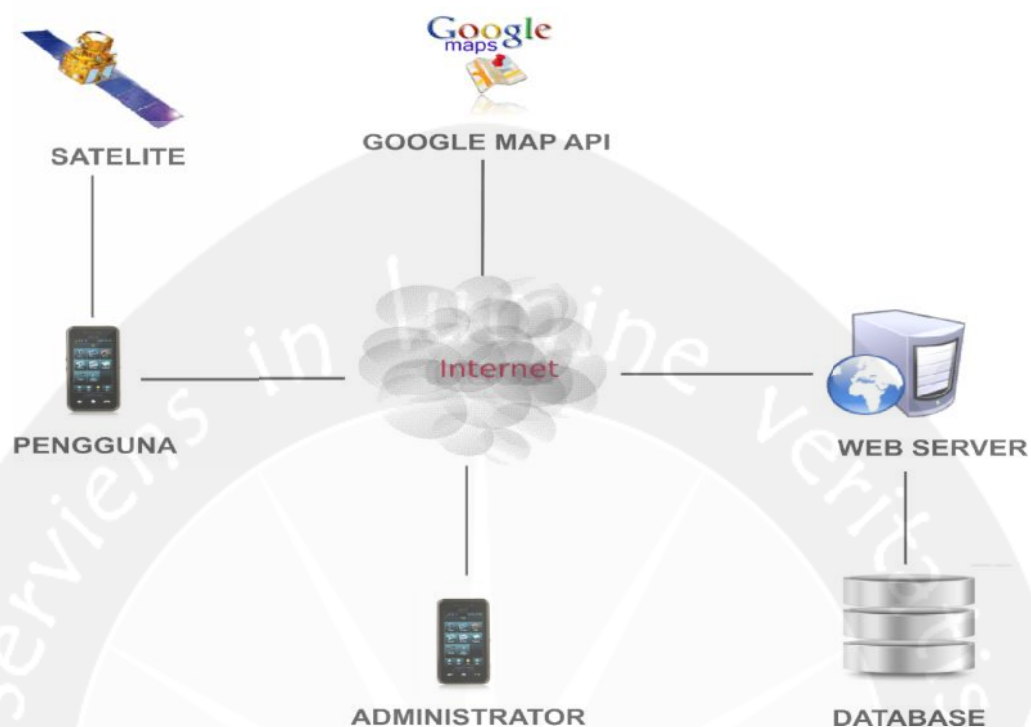
## 2 Deskripsi Kebutuhan

### 2.1 Perspektif produk

JOKER merupakan perangkat lunak yang dikembangkan untuk membantu pengguna mengetahui posisi dimana dirinya berada, kemudian dari posisi tersebut pengguna dapat mengetahui lokasi tempat wisata, penginapan, dan transportasi umum terdekat di sekitar dia berada. Pengguna juga dapat mengetahui informasi yang lebih detail tentang tempat yang ditampilkan tersebut. Selain itu aplikasi ini juga dapat membantu pengguna dalam memandu untuk mencapai lokasi tertentu yang telah ditampilkan sebelumnya dalam peta.

Perangkat lunak JOKER ini berjalan pada *platform* Android dengan kebutuhan minimal memiliki sistem operasi Android versi 2.2 (froyo) dan dibuat dengan bahasa pemrograman Java. Sedangkan untuk lingkungan pemrogramannya menggunakan Eclipse 3.6.2.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*) pada perangkat mobile. Pada sistem ini, seperti terlihat pada gambar 1, aplikasi ini bersifat *client-server*, yaitu pengguna mengakses data yang terdapat pada web server. Kemudian inputan data yang dimasukkan akan disimpan dalam basis data.



**Gambar 1. Arsitektur Perangkat Lunak JOKER**

## 2.2 Fungsi Produk

Fungsi produk perangkat lunak JOKER adalah sebagai berikut:

1. Fungsi *Login* Administrator(**SKPL-JOKER-001**).  
Merupakan fungsi yang digunakan administrator untuk dapat masuk ke dalam sistem JOKER.
2. Fungsi *Pengelolaan Data Venue* (**SKPL-JOKER-002**)  
Merupakan fungsi yang digunakan administrator untuk melakukan pengelolaan *venue* berupa data lokasi tempat penginapan, tempat makan, dan tempat transportasi umum.

Fungsi Pengelolaan Data Venue mencakup :

- a. Fungsi *Tambah Data Venue* (**SKPL-JOKER-002-01**)

Merupakan fungsi yang digunakan untuk memasukkan data *venue* yang baru.

b. Fungsi *Tampil Data Venue* (**SKPL-JOKER-002-02**)

Merupakan fungsi yang digunakan untuk menampilkan data *venue* yang telah ada sebelumnya.

c. Fungsi *Tampil Peta* (**SKPL-JOKER-002-03**)

Merupakan fungsi yang digunakan untuk menampilkan *venue* pada peta.

3. Fungsi *Login User* (**SKPL-JOKER-003**)

Merupakan fungsi yang digunakan pengguna untuk dapat masuk ke dalam sistem JOKER.

4. Fungsi *Pencarian Posisi Sekarang* (**SKPL-JOKER-004**)

Merupakan fungsi yang digunakan untuk memperoleh posisi pengguna saat itu juga.

5. Fungsi *Tampil Venue Berdasarkan Katagori* (**SKPL-JOKER-005**)

Merupakan fungsi yang digunakan untuk melakukan pencarian lokasi *venue* berdasarkan katagori. Adapun katagori yang tersedia yaitu lokasi tempat menginap, lokasi tempat makan, dan lokasi tempat transportasi umum.

Fungsi Pencarian *Venue* Berdasarkan Katagori meliputi :

a. Fungsi *Pencarian Lokasi Sekarang* (**SKPL-JOKER-005-01**)

Merupakan fungsi yang digunakan untuk memperoleh posisi pengguna saat itu.

b. Fungsi *Tampil Detail Venue* (**SKPL-JOKER-005-02**)

Merupakan fungsi yang digunakan untuk memperoleh informasi selengkapnya tentang *venue* yang dipilih.

c. Fungsi *Tampil Navigasi Venue* (**SKPL-JOKER-005-03**)

Merupakan fungsi yang digunakan untuk menampilkan rute navigasi yang akan dilalui pengguna untuk sampai pada lokasi *venue* yang dipilih.

d. Fungsi *Lihat dan Tambah Komentar* (**SKPL-JOKER-005-04**)

Merupakan fungsi untuk memperoleh data komentar antarpengguna pada *venue* yang dipilih pengguna dan menambahkan komentar.

6. Fungsi *Pencarian Venue Terdekat*(**SKPL-JOKER-006**).

Merupakan fungsi yang digunakan untuk mencari lokasi *venue* dalam radius 1 kilometer, dan menampilkannya dalam daftar *venue* terdekat.

Fungsi *Pencarian Venue* terdekat meliputi :

a. Fungsi *Pencarian Lokasi Sekarang* (**SKPL-JOKER-006-01**)

Merupakan fungsi yang digunakan untuk memperoleh posisi pengguna saat itu.

b. Fungsi *Tampil Detail Venue* (**SKPL-JOKER-006-02**)

Merupakan fungsi yang digunakan untuk memperoleh informasi selengkapnya tentang *venue* yang dipilih.

c. Fungsi *Tampil Navigasi Venue* (**SKPL-JOKER-006-03**)

Merupakan fungsi yang digunakan untuk menampilkan rute navigasi yang akan dilalui pengguna untuk sampai pada lokasi *venue* yang dipilih.

d. Fungsi *Tambah Komentar* (**SKPL-JOKER-006-04**)

Merupakan fungsi untuk memperoleh data komentar antarpengguna pada venue yang dipilih pengguna dan menambahkan komentar.

### 2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak JOKER adalah sebagai berikut :

1. Administrator.
  - a. Memahami pengoperasian komputer.
  - b. Mengerti tentang internet.
  - c. Memahami pengoperasian *web browser*.
2. Pengguna
  - a. Memahami pengoperasian perangkat mobile berbasis Android.

### 2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak JOKER tersebut adalah :

1. Kebijakan Umum  
Berpedoman pada tujuan dari pengembangan perangkat lunak JOKER.
2. Keterbatasan perangkat keras  
Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

### 2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada perangkat mobile yang menggunakan sistem operasi Android minimal dengan versi 2.2 (Froyo).

### 3 Kebutuhan khusus

#### 3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak JOKER meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

##### 3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form dan gambar.

##### 3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak JOKER adalah:

1. Perangkat *mobile* dengan sistem operasi Android.
2. Layar Sentuh
3. aGPS

##### 3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak Joker di sisi server adalah sebagai berikut :

1. Nama : MySQL  
Sumber : MySQL

Sebagai database management system (DBMS) yang digunakan untuk menyimpan data di sisi server.

2. Nama : Google Android  
Sumber : Google

Sebagai sistem operasi untuk perangkat mobile.

3. Nama : Apache  
Sumber : Apache

Sebagai web server lokal.

4. Nama : Google API v.8

Sumber : Google

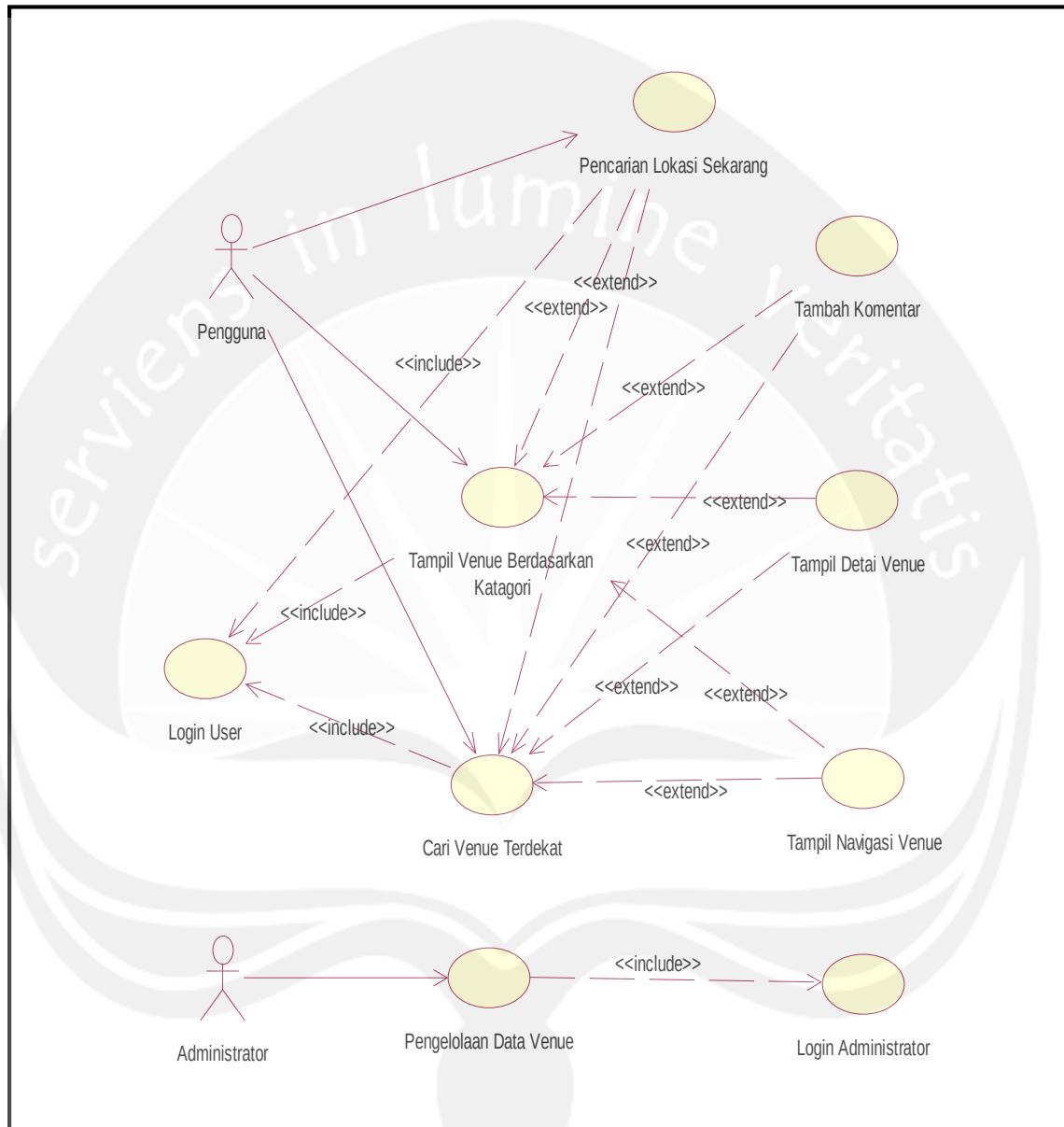
Sebagai framework tambahan.

#### **3.1.4 Antarmuka Komunikasi**

Antarmuka komunikasi perangkat lunak JOKER terhubung menggunakan protokol HTTP.

## 3.2 Kebutuhan fungsionalitas Perangkat Lunak

### 3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram

## **4 Spesifikasi Rinci Kebutuhan**

### **4.1 Spesifikasi Kebutuhan Fungsionalitas**

#### **4.1.1 Use case Spesification: Login Administrator**

##### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem aplikasi.

##### **2. Primary Actor**

Administrator

##### **3. Supporting Actor**

none

##### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor memasukkan id login dan password.
4. Sistem memeriksa id login dan password yang diinputkan aktor.

E-1 Password dan id login user tidak sesuai.

5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

##### **5. Alternative Flow**

none

##### **6. Error Flow**

E-1 Password dan id login user tidak sesuai

1. Sistem menampilkan peringatan bahwa password dan id login user tidak sesuai.
2. Kembali ke Basic Flow langkah ke-3.

##### **7. PreConditions**

1. Sistem berjalan dengan baik dan lancar.
2. Aktor memiliki koneksi internet sehingga dapat terhubung dengan sistem.

3. Sistem berjalan dengan baik dan lancar.
4. Aktor telah terdaftar pada sistem.
5. Aktor telah membuka halaman antarmuka login.

#### **8. PostConditions**

1. Aktor dapat memasuki sistem aplikasi.

#### **4.1.2 Use case Spesification: Pengelolaan Data Venue**

##### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan venue

##### **2. Primary Actor**

Administrator

##### **3. Supporting Actor**

none

##### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan venue. Pengelolaan venue meliputi tambah data venue, edit data venue, tampil data venue dan hapus data venue .
2. Sistem menampilkan antarmuka pengelolaan venue.
3. Aktor memilih untuk melakukan tambah data venue.
  - A-1 Aktor memilih untuk melakukan tampil data venue.
  - A-2 Aktor memilih untuk melakukan tampil venue pada peta.
4. Sistem menampilkan antarmuka untuk melakukan tambah data venue.
5. Aktor memasukkan data venue.
6. Aktor menyimpan data venue ke database.
  - E-1 Sambungan internet terputus.

7. Sistem menampilkan data venue yang telah terupdate.

8. Use Case ini selesai.

#### **5. Alternative Flow**

A-1 Aktor memilih untuk melakukan tampil data venue

1. Sistem menampilkan antarmuka untuk melakukan tampil data venue.

2. Lanjut ke basic flow langkah ke-2.

A-2 Aktor memilih untuk melakukan tampil venue pada peta.

1. Sistem menampilkan antarmuka untuk melakukan tampil venue pada peta.

2. Lanjut ke basic flow langkah ke-2.

#### **6. Error Flow**

E-1 Sambungan internet terputus.

1. Sistem menampilkan peringatan bahwa data yang diinputkan tidak tersimpan.

2. Kembali ke Basic Flow langkah ke-5.

#### **7. PreConditions**

1. Sistem berjalan dengan baik dan lancar.

2. Aktor telah membuka halaman antarmuka pengelolaan venue.

#### **8. PostConditions**

1. Aktor telah terdaftar sebagai pengguna dari sistem.

2. Aktor telah memiliki akun untuk memperoleh akses ke sistem.

### **4.1.3 Use case Spesification: Pencarian Posisi Sekarang**

#### **1. Brief Description**

|                                                                                                                                                                                                                    |            |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------|
| Program Studi Teknik Informatika                                                                                                                                                                                   | SKPL-JOKER | 19/ 26 |
| Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika |            |        |

Use Case ini digunakan oleh aktor untuk memperoleh posisi aktor saat itu juga. Dengan diketahuinya posisi ini maka dapat dilakukan aksi selanjutnya yaitu pencarian venue berdasarkan katagori dan pencarian venue terdekat.

## **2. Primary Actor**

Pengguna JOKER

## **3. Supporting Actor**

none

## **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pencarian posisi dirinya dalam peta Google.
2. Sistem menampilkan proses pencarian lokasi.  
E-1 Sambungan internet terputus.  
E-2 Koordinat lokasi tidak ditemukan.
3. Sistem menyimpan koordinat lokasi aktor.
4. Use Case selesai.

## **5. Alternative Flow**

none

## **6. Error Flow**

E-1 Sambungan internet terputus.

1. Sistem memberikan pesan peringatan bahwa sambungan internet terputus.
2. Kembali ke Basic Flow langkah ke-2.

E-2 Koordinat lokasi tidak ditemukan.

1. Sistem memberikan pesan bahwa lokasi tidak ditemukan.
2. Kembali ke Basic Flow langkah ke-2.

## **7. PreConditions**

1. Use Case Login telah dilakukan.

|                                                                                                                                                                                                                    |            |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------|
| Program Studi Teknik Informatika                                                                                                                                                                                   | SKPL-JOKER | 20/ 26 |
| Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika |            |        |

2. Aktor telah memasuki sistem.

#### **8. PostConditions**

1. Aktor telah menyimpan data lokasi sementara aktor dalam sistem.

#### **4.1.4 Use case Spesification: Login User**

##### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem aplikasi.

##### **2. Primary Actor**

1. Pengguna JOKER

##### **3. Supporting Actor**

none

##### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor memasukkan id login dan password.
4. Sistem memeriksa id login dan password yang diinputkan aktor.

E-1 Password dan id login user tidak sesuai.

5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

##### **5. Alternative Flow**

none

##### **6. Error Flow**

E-1 Password dan id login user tidak sesuai

1. Sistem menampilkan peringatan bahwa password dan id login user tidak sesuai.
2. Kembali ke Basic Flow langkah ke-3.

##### **7. PreConditions**

|                                                                                                                                                                                                                    |            |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------|
| Program Studi Teknik Informatika                                                                                                                                                                                   | SKPL-JOKER | 21/ 26 |
| Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika |            |        |

1. Sistem berjalan dengan baik dan lancar.
2. Aktor memiliki koneksi internet sehingga dapat terhubung dengan sistem.
3. Sistem berjalan dengan baik dan lancar.
4. Aktor telah terdaftar pada sistem.
5. Aktor telah membuka halaman antarmuka login.

#### **8. PostConditions**

1. Aktor dapat memasuki sistem aplikasi.

#### **4.1.5 Use case Spesification: Lihat Detail Venue**

##### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk melihat detail informasi venue yang dipilih aktor.

##### **2. Primary Actor**

1. Pengguna JOKER
2. Administrator

##### **3. Supporting Actor**

none

##### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan lihat detail venue.
2. Sistem mencari detail venue berdasarkan yang dipilih aktor.

E-1 Sambungan internet terputus.

3. Sistem menampilkan detail venue.
4. Use Case selesai.

##### **5. Alternative Flow**

none

##### **6. Error Flow**

E-1 Data Sambungan internet terputus.

1. Sistem memberikan pesan peringatan bahwa sambungan internet terputus.

2. Kembali ke Basic Flow langkah ke-2

#### **7. PreConditions**

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

3. Use Case Pencarian venue berdasarkan katagori dan pencarian venue terdekat sudah dilakukan.

#### **8. PostConditions**

1. Aktor telah melihat detail informasi venue.

### **4.1.6 Use case Spesification: Tampil Navigasi Venue**

#### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk melihat navigasi rute yang harus dilalui aktor untuk sampai pada lokasi venue yang dipilih.

#### **2. Primary Actor**

Pengguna JOKER

#### **3. Supporting Actor**

none

#### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan tampil navigasi venue.

2. Sistem memasukkan koordinat lokasi venue dan lokasi keberadaan aktor.

3. Sistem menampilkan navigasi Google.

E-1 Sambungan internet terputus.

4. Use Case selesai.

#### **5. Alternative Flow**

none

#### **6. Error Flow**

|                                                                                                                                                                                                                    |            |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------|
| Program Studi Teknik Informatika                                                                                                                                                                                   | SKPL-JOKER | 23/ 26 |
| Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika |            |        |

E-1 Sambungan internet terputus

1. Sistem menampilkan pemberitahuan bahwa sambungan internet terputus.
2. Kembali ke Basic Flow langkah ke-3.

#### **7. PreConditions**

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.
3. Use Case Pencarian venue berdasarkan katagori dan pencarian venue terdekat sudah dilakukan.

#### **8. PostConditions**

1. Aktor memulai navigasi Google dari posisi aktor hingga lokasi venue.

#### **4.1.7 Use case Spesification : Tambah Komentar**

##### **1. Brief Description**

Use Case ini digunakan oleh aktor untuk menampilkan dan menambahkan komentar pada suatu venue yang dipilih aktor.

##### **2. Primary Actor**

Pengguna JOKER

##### **3. Supporting Actor**

none

##### **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan tampil dan tambah komentar untuk venue yang dipilih.
2. Sistem mencari data komentar venue yang dipilih oleh aktor.
3. Sistem menampilkan komentar venue.

E-1 Sambungan internet terputus.

4. Sistem menyediakan menu tambah komentar.

|                                                                                                                                                                                                                    |            |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------|
| Program Studi Teknik Informatika                                                                                                                                                                                   | SKPL-JOKER | 24/ 26 |
| Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika |            |        |

A-1 Aktor memilih untuk menambah komentar.

5. Use Case selesai.

#### **5. Alternative Flow**

A-1 Aktor memilih untuk menambah komentar.

1. Aktor memasukkan komentar.
2. Aktor memilih menambahkan komentar ke sistem.
3. Sistem menambahkan komentar aktor pada venue.
4. Kembali ke basic flow langkah ke-5.

#### **6. Error Flow**

E-1 Sambungan internet terputus.

1. Sistem memberikan pesan peringatan bahwa sambungan internet terputus.
2. Kembali ke Basic Flow Langkah ke-3.

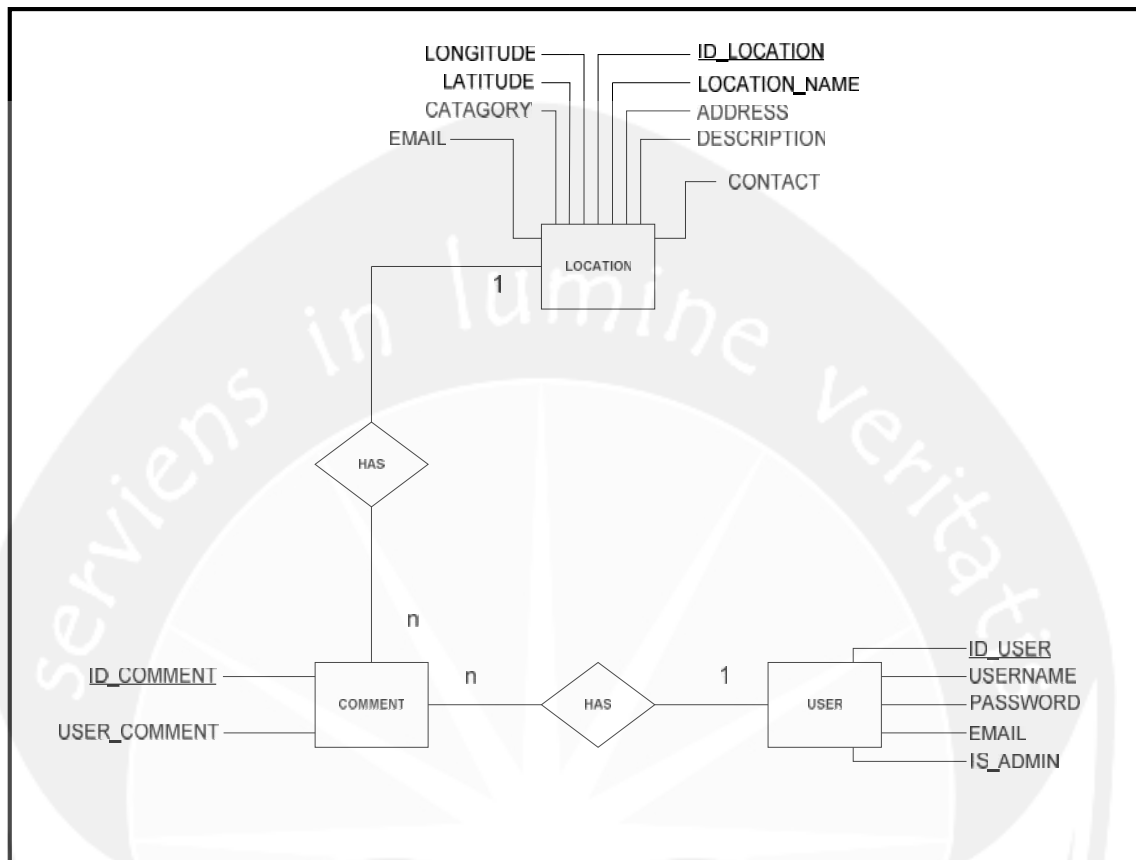
#### **7. PreConditions**

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.
3. Use Case Pencarian venue berdasarkan kategori dan pencarian venue terdekat sudah dilakukan.

#### **8. PostConditions**

1. Aktor berhasil menambahkan komentar pada venue yang dipilih aktor.

## 5 Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram

# DPPL

## DESKRIPSI PERANCANGAN PERANGKAT LUNAK

### JOKER (Jogja Backpacker)

Dipersiapkan oleh:

Yulius Dharmayanto / 08 07 05720

Program Studi Teknik Informatika – Fakultas Teknologi  
Industri

Universitas Atma Jaya Yogyakarta

|                                                                                     |                                                                             |                     |  |             |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|---------------------|--|-------------|
|  | Program Studi<br>Teknik<br>Informatika<br>Fakultas<br>Teknologi<br>Industri | Nomor Dokumen       |  | Halaman     |
|                                                                                     |                                                                             | <b>DPPL - JOKER</b> |  | <b>1/38</b> |
|                                                                                     |                                                                             | Revisi              |  |             |

## DAFTAR PERUBAHAN

| Revisi   | Deskripsi |
|----------|-----------|
| <b>A</b> |           |
| <b>B</b> |           |
| <b>C</b> |           |
| <b>D</b> |           |
| <b>E</b> |           |
| <b>F</b> |           |

| INDEX<br>TGL       | -  | A | B | C | D | E | F | G |
|--------------------|----|---|---|---|---|---|---|---|
| Ditulis<br>oleh    | YD |   |   |   |   |   |   |   |
| Diperik<br>sa oleh |    |   |   |   |   |   |   |   |
| Disetuj<br>ui oleh |    |   |   |   |   |   |   |   |

## Daftar Halaman Perubahan

| Halaman                                                                             | Revisi | Halaman | Revisi |
|-------------------------------------------------------------------------------------|--------|---------|--------|
|  |        |         |        |

# Daftar Isi

|                                                       |           |
|-------------------------------------------------------|-----------|
| <b>1 PENDAHULUAN.....</b>                             | <b>7</b>  |
| 1.1 Tujuan.....                                       | 7         |
| 1.2 Ruang Lingkup.....                                | 7         |
| 1.3 Definisi dan Akronim.....                         | 7         |
| 1.4 Referensi.....                                    | 8         |
| <b>2 PERANCANGAN SISTEM.....</b>                      | <b>9</b>  |
| 2.1 Perancangan Arsitektur.....                       | 9         |
| 2.2 Perancangan RinciSequence Diagram.....            | 10        |
| 2.2.1.1 Login Administrator .....                     | 10        |
| 2.2.1.2 Mengelola Data Lokasi.....                    | 11        |
| 2.2.1.3 Login User .....                              | 13        |
| 2.2.2 Class Diagram .....                             | 16        |
| 2.2.3 Class Diagram Specific Descriptions .....       | 17        |
| 2.2.3.1 <i>Specific Design Class</i> loginAdmin ....  | 17        |
| 2.2.3.2 <i>Specific Design Class</i> submitLocation   | 17        |
| 2.2.3.3 <i>Specific Design Class</i> locationList ..  | 17        |
| 2.2.3.4 <i>Specific Design Class</i> loginUser .....  | 17        |
| 2.2.3.5 <i>Specific Design Class</i> searchLocation   | 18        |
| 2.2.3.6 <i>Specific Design Class</i> addComment ..... | 18        |
| 2.2.3.7 <i>Specific Design Class</i> getDirrection .  | 18        |
| 2.2.3.8 <i>Specific Design Class</i> getMyLocation .  | 18        |
| 2.2.3.9 <i>Specific Design Class</i> viewOnMap .....  | 19        |
| 2.2.3.10 <i>Specific Design Class</i> userMgr .....   | 19        |
| 2.2.3.11 <i>Specific Design Class</i> commentMgr .... | 19        |
| 2.2.3.12 <i>Specific Design Class</i> locationMgr ... | 19        |
| 2.2.3.13 <i>Specific Design Class</i> users .....     | 20        |
| 2.2.3.14 <i>Specific Design Class</i> locations ..... | 21        |
| 2.2.3.15 <i>Specific Design Class</i> comments .....  | 22        |
| 2.2.3.16 <i>Specific Design Class</i> googleApi ..... | 23        |
| <b>3 PERANCANGAN DATA.....</b>                        | <b>24</b> |
| 3.1 Dekomposisi Data.....                             | 24        |
| 3.1.1 Deskripsi Entitas users .....                   | 24        |
| 3.1.2 Deskripsi Entitas locations .....               | 24        |
| 3.1.3 Deskripsi Entitas comments .....                | 24        |
| 3.2 Physical Data Model.....                          | 25        |
| <b>4 PERANCANGAN ANTARMUKA.....</b>                   | <b>26</b> |
| 4.1 Antarmuka Halaman Login Admin.....                | 26        |
| 4.2 Antarmuka Beranda Admin.....                      | 27        |
| 4.3 Antarmuka Submit Location.....                    | 28        |
| 4.4 Antarmuka List Lokasi.....                        | 29        |
| 4.5 Antarmuka Deskripsi Lokasi (admin).....           | 30        |
| 4.6 Antarmuka Halaman Utama User.....                 | 32        |
| 4.7 Antarmuka Halaman Deskripsi Lokasi (User).....    | 34        |

|                                     |    |
|-------------------------------------|----|
| 4.8 Antarmuka Login User.....       | 36 |
| 4.9 Antarmuka Sign Up.....          | 37 |
| 4.10 Antarmuka Tambah Komentar..... | 38 |



## Daftar Gambar

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Gambar 1. Perancangan Arsitektur .....                            | 9  |
| Gambar 2. Sequence Diagram : Login Administrator ....             | 10 |
| Gambar 3. Sequence Diagram : <i>Add Venue</i> .....               | 11 |
| Gambar 4. Sequence Diagram : <i>Display Info Location</i> ..      | 11 |
| Gambar 5. Sequence Diagram : <i>View on Map</i> .....             | 12 |
| Gambar 6. Sequence Diagram : Login User .....                     | 13 |
| Gambar 7. Sequence Diagram : <i>Display Info Location</i> ..      | 13 |
| Gambar 8. Sequence Diagram : <i>Add Comment</i> .....             | 14 |
| Gambar 10. Sequence Diagram : <i>Get Dirrection</i> .....         | 15 |
| Gambar 11. Sequence Diagram : <i>Search Location</i> .....        | 15 |
| Gambar 12. Class Diagram .....                                    | 16 |
| Gambar 13. Physical Data Model .....                              | 25 |
| Gambar 14. Rancangan Antarmuka Login Admin .....                  | 26 |
| Gambar 15 Rancangan Antarmuka Beranda .....                       | 27 |
| Gambar 16 Rancangan Antarmuka Submit .....                        | 28 |
| Gambar 17 Rancangan Antarmuka List Lokasi .....                   | 29 |
| Gambar 18 Rancangan Antarmuka Deskripsi Lokasi .....              | 30 |
| Gambar 19 Rancangan Antarmuka detail lokasi dalam peta<br>.....   | 31 |
| Gambar 20 Perancangan Antarmuka Halaman Utama User ..             | 32 |
| Gambar 21 Perancangan Antarmuka Peta My Location ....             | 33 |
| Gambar 22 Perancangan Antarmuka Halaman Deskripsi<br>Lokasi ..... | 34 |
| Gambar 23 Rancangan Antarmuka Get Direction .....                 | 35 |
| Gambar 24 Perancangan Antarmuka Login User .....                  | 36 |
| Gambar 25 Perancangan Antarmuka Sign Up .....                     | 37 |
| Gambar 26 Perancangan Antarmuka Add Comment .....                 | 38 |

## 1 Pendahuluan

### 1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

### 1.2 Ruang Lingkup

Perangkat Lunak JOKER dikembangkan dengan tujuan untuk :

1. Menangani pencarian posisi pengguna.
2. Menangani pencarian venue terdekat dari posisi pengguna.
3. Menangani penyimpanan data lokasi venue.
4. Menangani informasi seputar venue yang ditampilkan.
5. Menangani navigasi rute perjalanan dari posisi pengguna sekarang hingga sampai pada lokasi venue.
6. Menangani pemberian komentar pada suatu venue.

Dan berjalan pada lingkungan dengan *platform* Android.

### 1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

| Keyword/Phrase | Definisi                                                                                                                                                                 |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DPPL           | Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan. |

|          |                                                                                                                                                                                                                                                           |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| JOKER    | Merupakan akronim dari Jogja Backpacker yang merupakan aplikasi untuk menampilkan informasi dan memberikan navigasi pada suatu venue.                                                                                                                     |
| Venue    |                                                                                                                                                                                                                                                           |
| Database | Merupakan tempat penyimpanan data.                                                                                                                                                                                                                        |
| LBS      | Merupakan singkatan dari Location Based Service yang merupakan bentuk layanan informasi yang diakses oleh perangkat mobile melalui jaringan mobile dan memiliki kemampuan untuk memanfaatkan posisi geografis dari perangkat mobile tersebut.             |
| Location | Merupakan posisi spesifik dari suatu ruang secara fisik.                                                                                                                                                                                                  |
| Internet | Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web. |

#### 1.4 Referensi

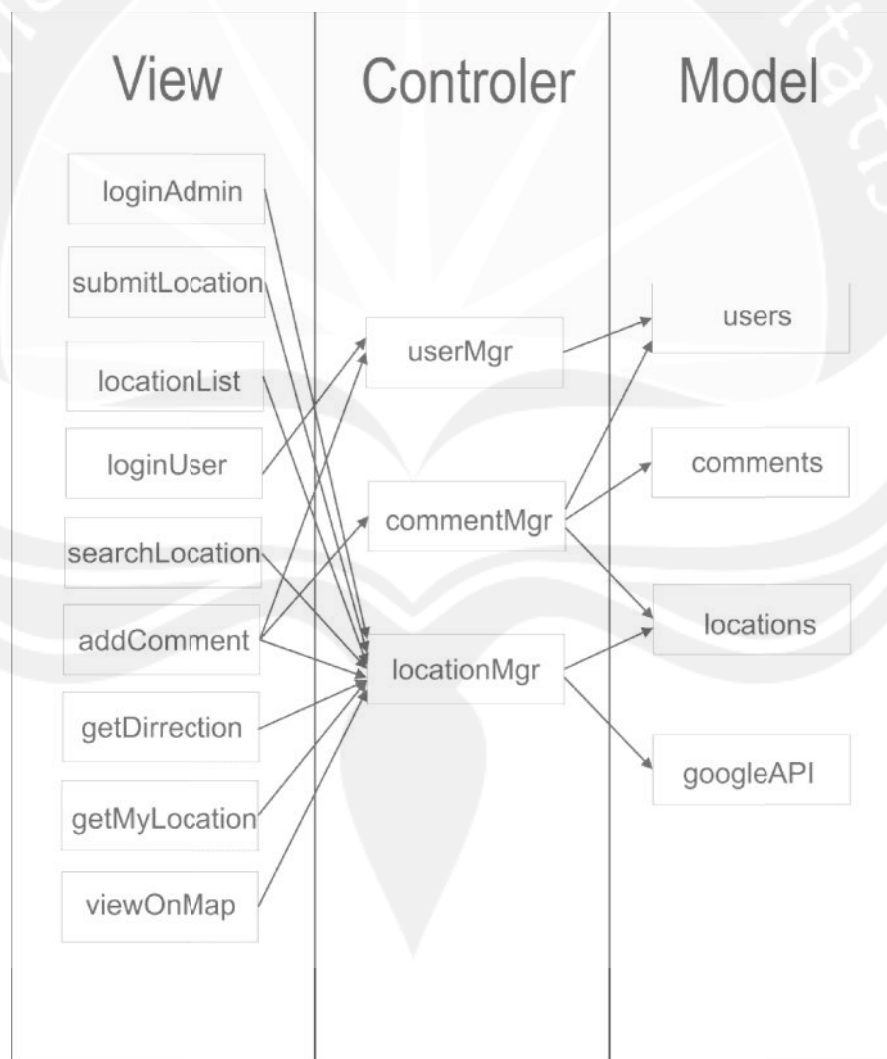
Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Dharmayanto, Yulius. *Spesifikasi Kebutuhan Perangkat Lunak Joker*, Universitas Atma Jaya Yogyakarta, 2012.

2. Dharmayanto, Yulius. *Deskripsi Perancangan Perangkat Lunak AMS*, Universitas Atma Jaya Yogyakarta, 2011.
3. Vialliresa, Emmanuel. *Deskripsi Perancangan Perangkat Lunak SIMPONI UAJY*, Universitas Atma Jaya Yogyakarta, 2011.

## 2 Perancangan Sistem

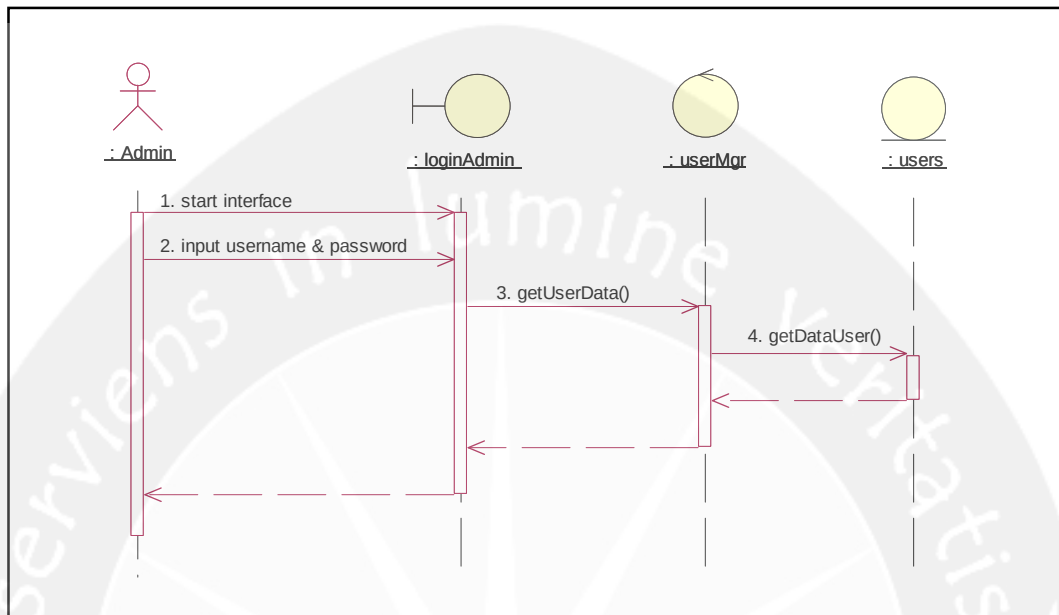
### 2.1 Perancangan Arsitektur



**Gambar 1. Perancangan Arsitektur**

## 2.2 Perancangan RinciSequence Diagram

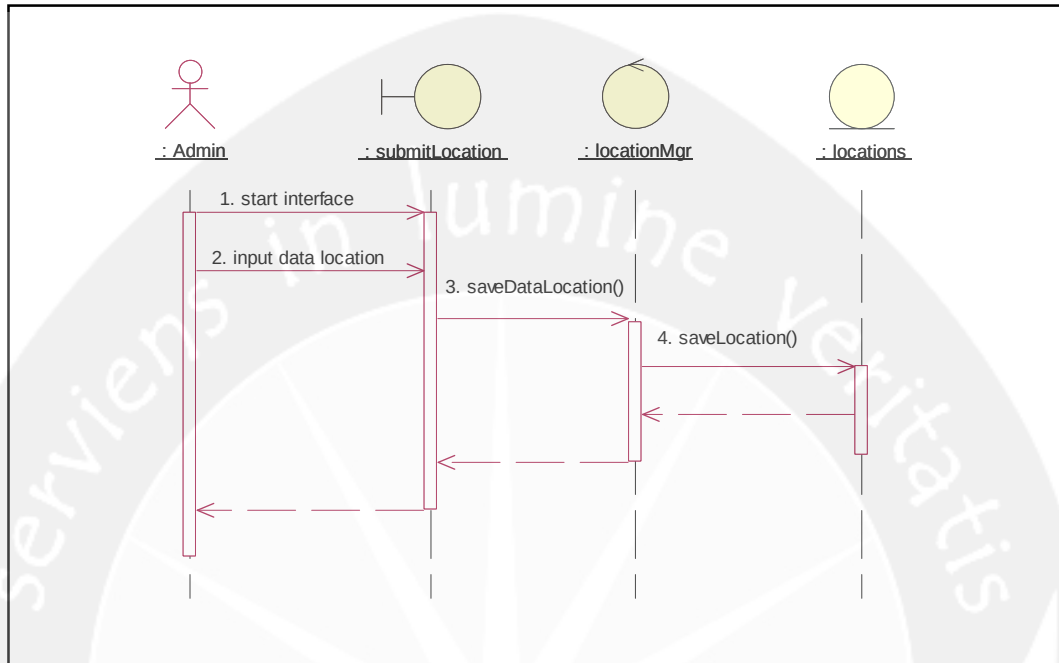
### 2.2.1.1 Login Administrator



Gambar 2. Sequence Diagram : Login Administrator

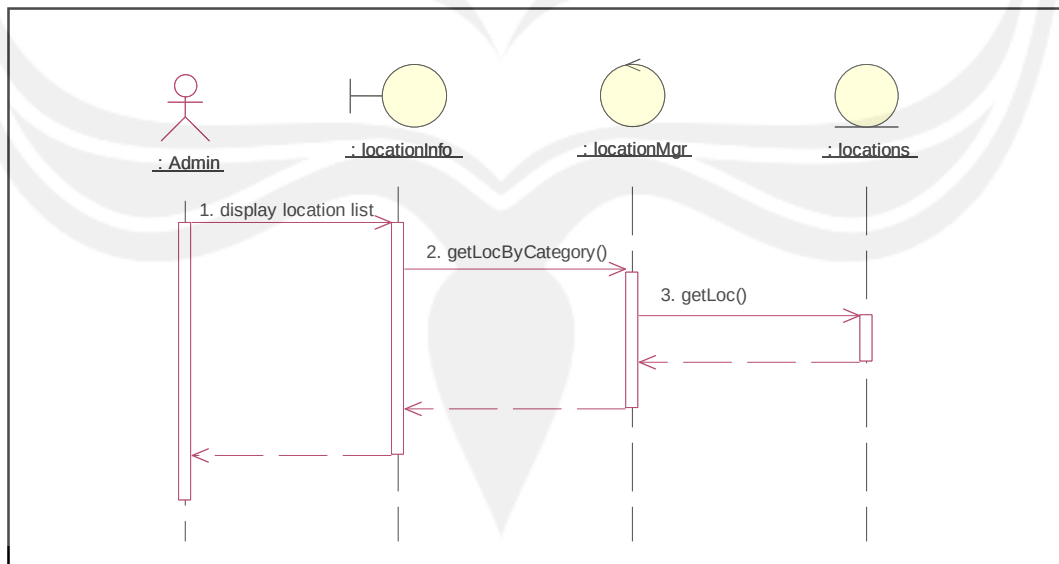
## 2.2.1.2 Mengelola Data Lokasi

### 2.2.1.2.1 Add Venue



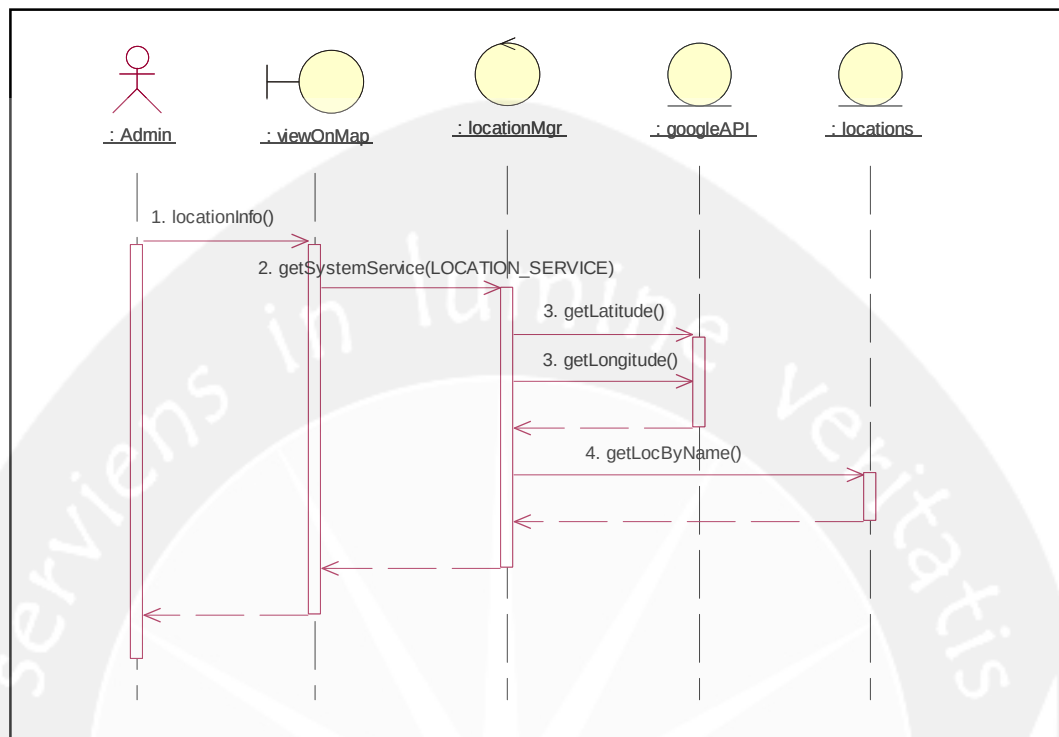
Gambar 3.Sequence Diagram : *Add Venue*

### 2.2.1.2.2 Display Info Location



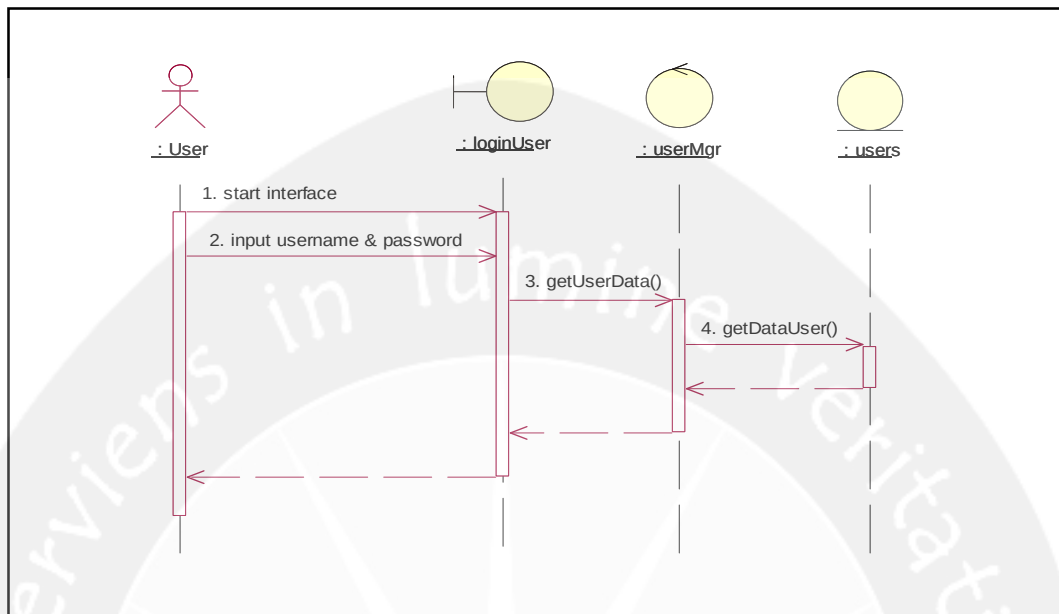
Gambar 4. Sequence Diagram : *Display Info Location*

### 2.2.1.2.3 View on Map



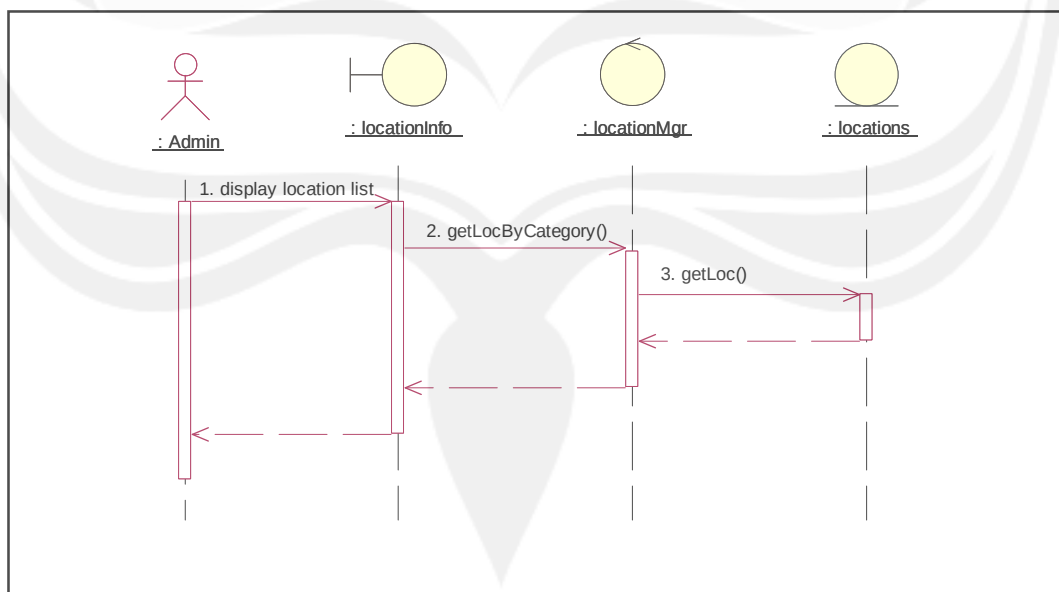
Gambar 5. Sequence Diagram : View on Map

### 2.2.1.3 Login User



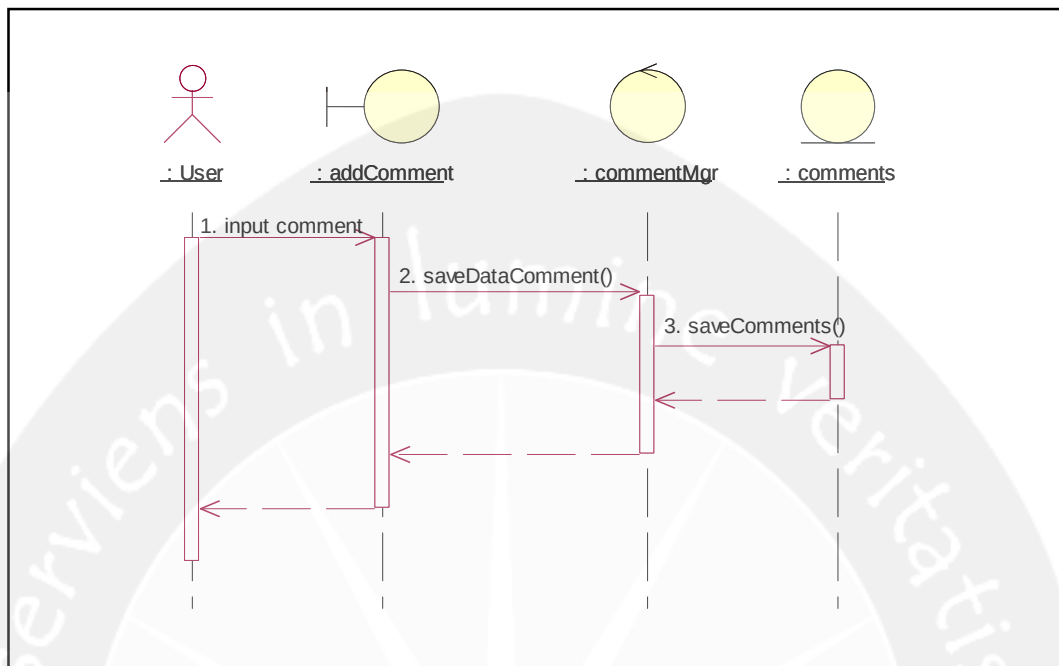
Gambar 6. Sequence Diagram : Login User

### 2.2.1.4 Display Info Location



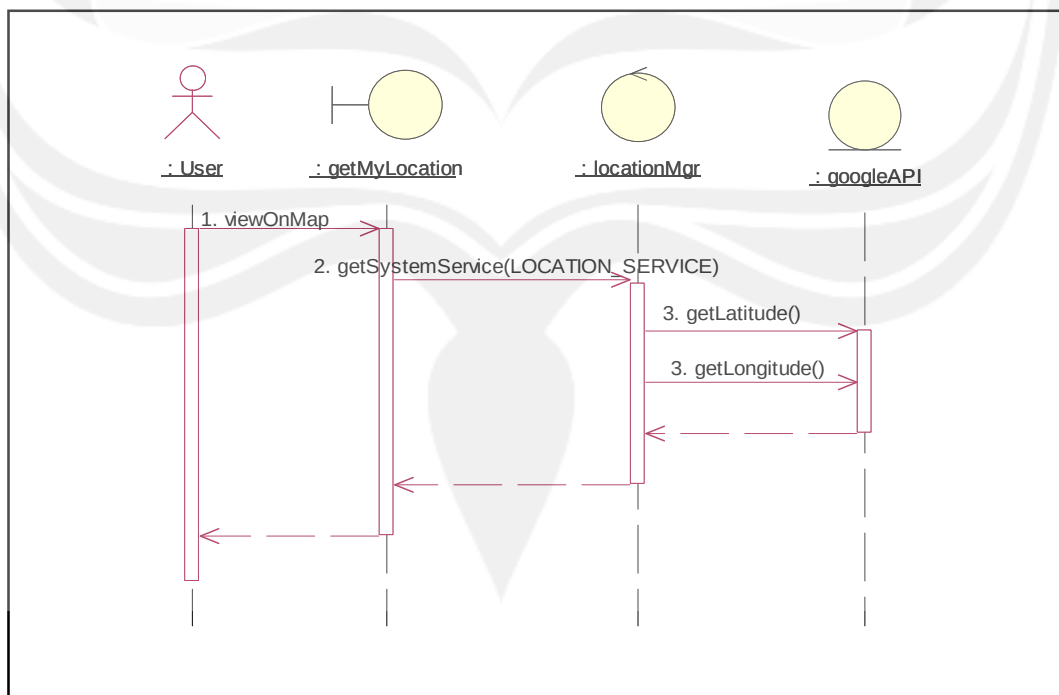
Gambar 7. Sequence Diagram : *Display Info Location*

#### 2.2.1.5 Add Comment



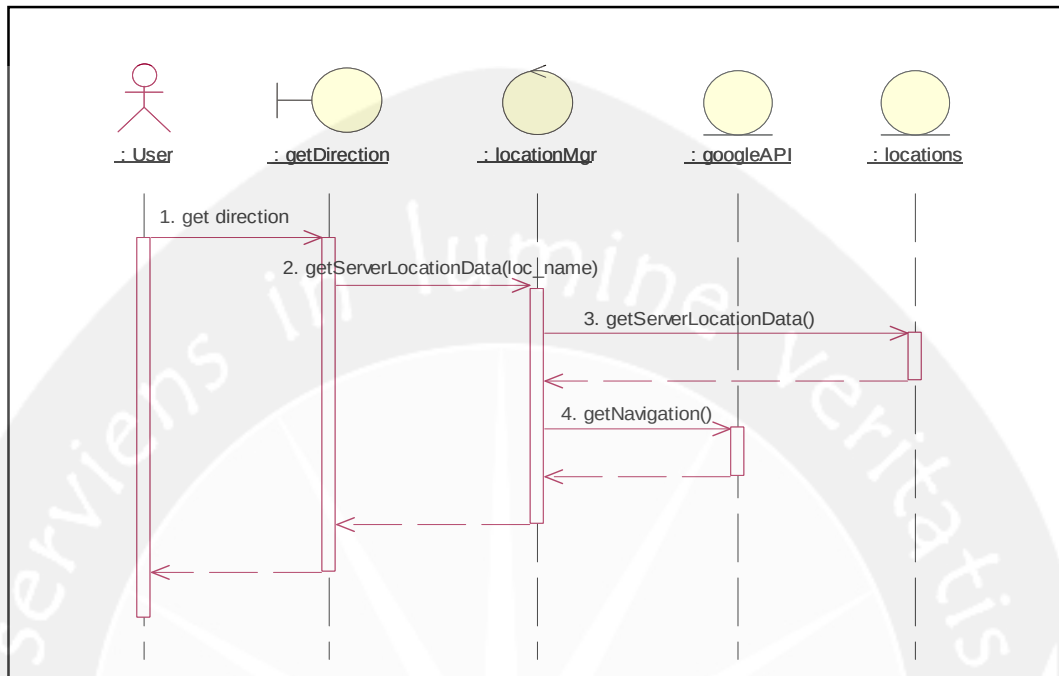
Gambar 8. Sequence Diagram : Add Comment

#### 2.2.1.6 Get My Location



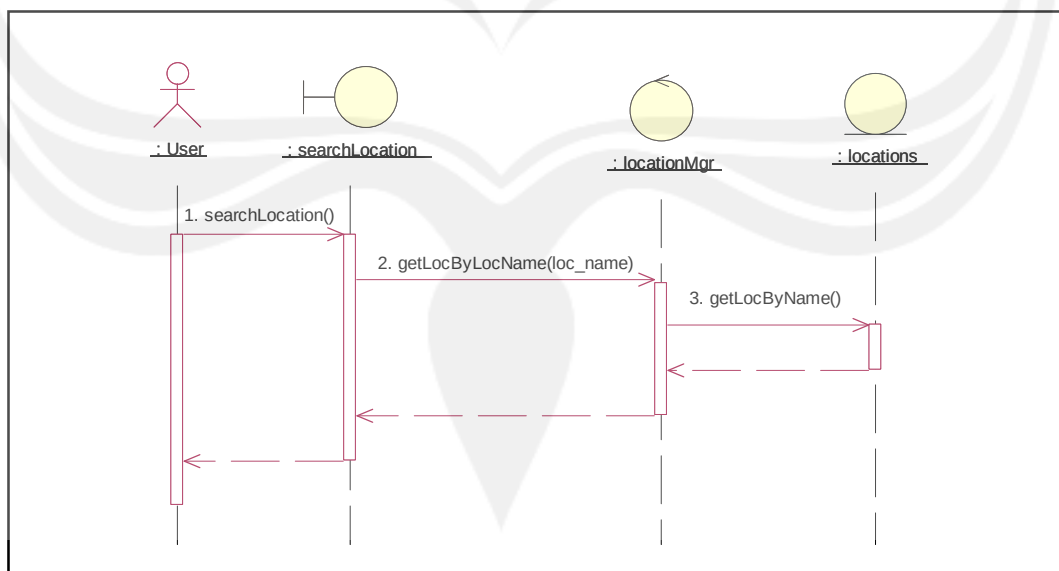
Gambar 9. Sequence Diagram : Get My Location

### 2.2.1.7 Get Dirrection



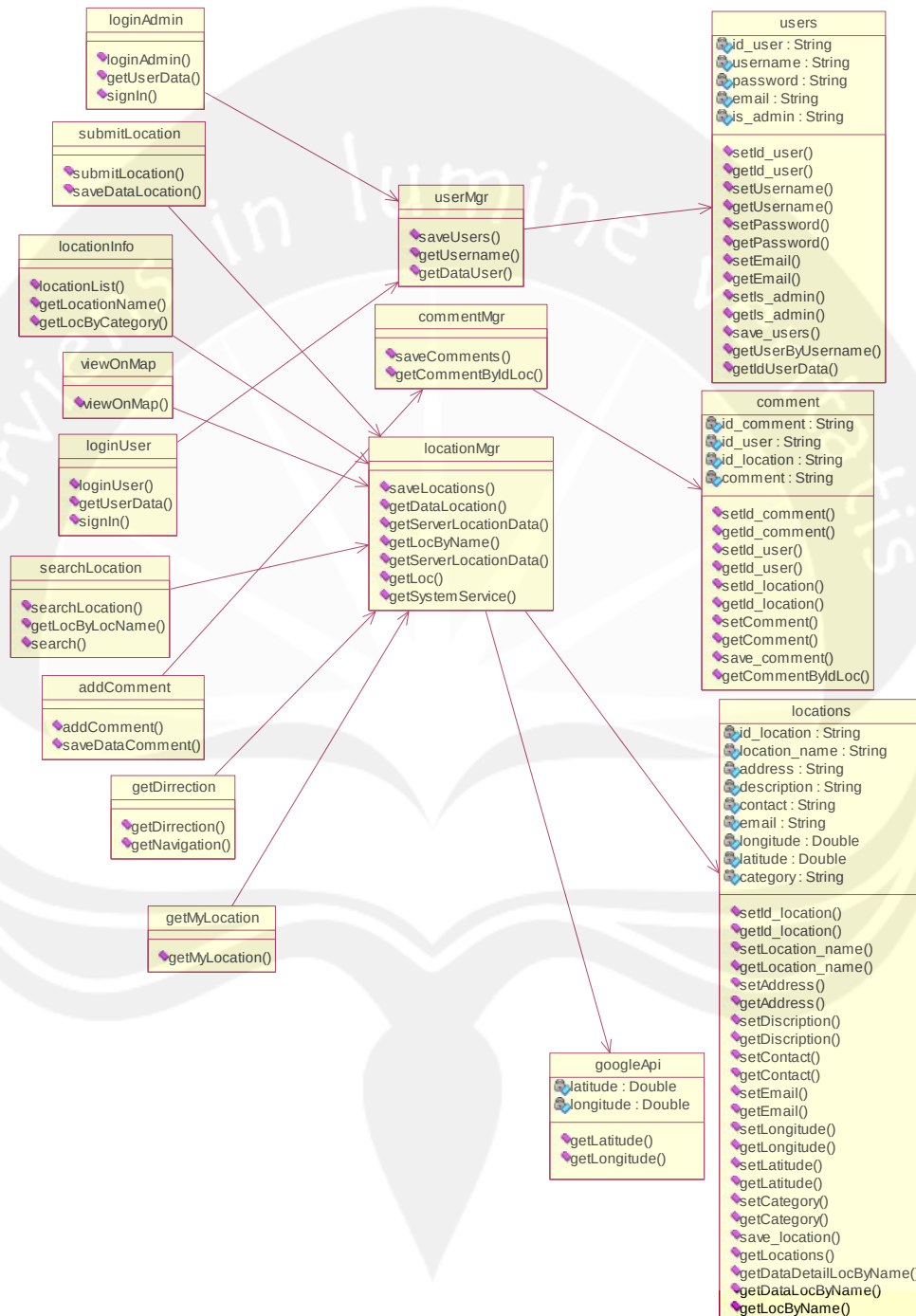
Gambar 10. Sequence Diagram : *Get Dirrection*

### 2.2.1.8 Search Location



Gambar 11. Sequence Diagram : *Search Location*

### 2.2.2 Class Diagram



**Gambar 12. Class Diagram**

### 2.2.3 Class Diagram Specific Descriptions

#### 2.2.3.1 Specific Design Class loginAdmin

|                                                                                                             |                                 |
|-------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>loginAdmin</b>                                                                                           | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+loginAdmin()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas login. |                                 |
| <b>+getUserData(username,pass) : String</b><br>Prosedur ini digunakan untuk mencari data admin.             |                                 |
| <b>+signIn()</b><br>Prosedur ini digunakan untuk melakukan proses masuk ke dalam aplikasi.                  |                                 |

#### 2.2.3.2 Specific Design Class submitLocation

|                                                                                                                     |                                 |
|---------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>submitLocation</b>                                                                                               | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+ submitLocation ()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas manager. |                                 |
| <b>+saveDataLocation()</b><br>Prosedur untuk melakukan perintah penyimpanan data lokasi.                            |                                 |

#### 2.2.3.3 Specific Design Class locationList

|                                                                                                                      |                                 |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>locationInfo</b>                                                                                                  | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+locationInfo()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas locationList. |                                 |
| <b>+getLocationName()</b><br>Prosedur untuk melakukan perintah pengambilan nama lokasi data lokasi.                  |                                 |
| <b>+getLocByCategory</b><br>Prosedur untuk melakukan perintah pengambilan data lokasi berdasarkan kategori.          |                                 |

#### 2.2.3.4 Specific Design Class loginUser

|                                                                                                            |                                 |
|------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>loginUser</b>                                                                                           | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+loginUser()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas login. |                                 |
| <b>+getUserData(username,pass) : String</b><br>Prosedur ini digunakan untuk mencari data pengguna.         |                                 |
| <b>+signIn()</b>                                                                                           |                                 |

Prosedur ini digunakan untuk melakukan proses masuk ke dalam aplikasi.

#### 2.2.3.5 *Specific Design Class searchLocation*

|                                                                                                                          |                                 |
|--------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>searchLocation</b>                                                                                                    | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+searchLocation()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas searchLocation. |                                 |
| <b>+getLocByLocName()</b><br>Prosedur ini digunakan untuk mencari data lokasi berdasarkan nama lokasi.                   |                                 |
| <b>+search()</b><br>Prosedur ini digunakan untuk melakukan proses pencarian.                                             |                                 |

#### 2.2.3.6 *Specific Design Class addComment*

|                                                                                                                  |                                 |
|------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>addComment</b>                                                                                                | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+addComment()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas addComment. |                                 |
| <b>+saveDataComment()</b><br>Prosedur untuk melakukan perintah penyimpanan data komentar.                        |                                 |

#### 2.2.3.7 *Specific Design Class getDirrection*

|                                                                                                                         |                                 |
|-------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>getDirrection</b>                                                                                                    | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+getDirrection ()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas getDirrection. |                                 |
| <b>+getNavigation()</b><br>Prosedur untuk melakukan perintah pencarian rute.                                            |                                 |

#### 2.2.3.8 *Specific Design Class getMyLocation*

|                                                                                                                          |                                 |
|--------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>getMyLocation</b>                                                                                                     | <b>&lt;&lt;boundary&gt;&gt;</b> |
| <b>+ getMyLocation ()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas getMyLocation. |                                 |

### 2.2.3.9 Specific Design Class viewOnMap

|                                                                                                                 |                                 |
|-----------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>viewOnMap</b>                                                                                                | <b>&lt;&lt;boundary&gt;&gt;</b> |
|                                                                                                                 |                                 |
| <b>+viewOnMap ()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas viewOnMap. |                                 |

### 2.2.3.10 Specific Design Class userMgr

|                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <b>userMgr</b>                                                                                                                                                                                                                                                                                                                                                                                                                   | <b>&lt;&lt;control&gt;&gt;</b> |
|                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                |
| <b>+userMgr()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas userMgr.<br><b>+saveUsers()</b><br>Prosedur ini digunakan untuk menyimpan data pengguna.<br><b>+getUsername()</b><br>Prosedur ini digunakan untuk mendapatkan username dari basis data.<br><b>+getDataUser()</b><br>Prosedur ini digunakan untuk mendapatkan data user yang digunakan dalam verifikasi ketika melakukan login. |                                |

### 2.2.3.11 Specific Design Class commentMgr

|                                                                                                                                                                                                                                                                                                                                                              |                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <b>commentMgr</b>                                                                                                                                                                                                                                                                                                                                            | <b>&lt;&lt;control&gt;&gt;</b> |
|                                                                                                                                                                                                                                                                                                                                                              |                                |
| <b>+commentMgr()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas commentMgr.<br><b>+saveComments()</b><br>Prosedur ini digunakan untuk menyimpan data komentar dari pengguna.<br><b>+getCommentByIdLoc()</b><br>Prosedur ini digunakan untuk mendapatkan data komentar yang akan ditampilkan dalam location description. |                                |

### 2.2.3.12 Specific Design Class locationMgr

|                                                                                                                                                                                                                                                                                                                                              |                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <b>locationMgr</b>                                                                                                                                                                                                                                                                                                                           | <b>&lt;&lt;control&gt;&gt;</b> |
|                                                                                                                                                                                                                                                                                                                                              |                                |
| <b>+ locationMgr()</b><br>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas locationMgr.<br><b>+saveLocations()</b><br>Prosedur ini digunakan untuk menyimpan data lokasi.<br><b>+getDataLocation()</b><br>Prosedur ini digunakan untuk mendapatkan data lokasi yang akan ditampilkan dalam location description. |                                |

**+getServerLocationData()**

Prosedur ini digunakan untuk mendapatkan data id lokasi, longitude dan latitude dari basis data.

**+getLocByName()**

Prosedur ini digunakan untuk mendapatkan data lokasi berdasarkan nama lokasi.

**+getLoc()**

Prosedur ini digunakan untuk mendapatkan data lokasi yang akan ditampilkan dalam location description.

**2.2.3.13 Specific Design Class users**

| users                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <<entity>> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>-id_user : String</b><br>Atribut ini untuk menyimpan data id pengguna<br><b>-username : String</b><br>Atribut ini untuk menyimpan data username pengguna<br><b>-password : String</b><br>Atribut ini untuk menyimpan data password pengguna<br><b>-email : String</b><br>Atribut ini untuk menyimpan data email pengguna<br><b>-is_admin : String</b><br>Atribut ini untuk menyimpan data status pengguna                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |
| <b>+users()</b><br>Default konstruktor dari kelas user<br><b>+save_user() : String</b><br>Untuk menambah data pengguna ke dalam basis data<br><b>+getUserByUsername() : String</b><br>Untuk memanggil data pengguna yang ada di dalam basis data, sesuai dengan masukan username<br><b>+getUserData() : String</b><br>Untuk memanggil data pengguna yang ada di dalam basis data<br><b>-setUsername()</b><br>Prosedur untuk menginisialisasi atribut username<br><b>-setPassword()</b><br>Prosedur untuk menginisialisasi atribut password<br><b>-setEmail()</b><br>Prosedur untuk menginisialisasi atribut email<br><b>-setId_User()</b><br>Prosedur untuk menginisialisasi atribut idUser<br><b>-setIs_Admin()</b><br>Prosedur untuk menginisialisasi atribut isAdmin<br><b>-getUsername : String</b><br>Fungsi untuk mendapatkan nilai dari atribut username<br><b>-getPassword : String</b><br>Fungsi untuk mendapatkan nilai dari atribut password<br><b>-getEmail : String</b><br>Fungsi untuk mendapatkan nilai dari atribut email<br><b>-getId_User : String</b><br>Fungsi untuk mendapatkan nilai dari atribut idUser |            |

**-getIs\_Admin : String**

Fungsi untuk mendapatkan nilai dari atribut isAdmin

#### 2.2.3.14 *Specific Design Class locations*

| locations                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <<entity>> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <p><b>-id_location : String</b><br/>Atribut ini untuk menyimpan data id lokasi</p> <p><b>-location_name : String</b><br/>Atribut ini untuk menyimpan data nama lokasi</p> <p><b>-address : String</b><br/>Atribut ini untuk menyimpan data alamat lokasi</p> <p><b>-description : String</b><br/>Atribut ini untuk menyimpan data deskripsi dari lokasi</p> <p><b>-contact</b><br/>Atribut ini untuk menyimpan data kontak dari lokasi</p> <p><b>-email</b><br/>Atribut ini untuk menyimpan data email dari lokasi</p> <p><b>-category : String</b><br/>Atribut ini untuk menyimpan data kategori lokasi lokasi tersebut</p> <p><b>-latitude: String</b><br/>Atribut ini untuk menyimpan data latitude lokasi</p> <p><b>-longitude : String</b><br/>Atribut ini untuk menyimpan data longitude lokasi</p>                                                                                                                                                                                                                                     |            |
| <p><b>+locations()</b><br/>Default konstruktor dari kelas locations</p> <p><b>+save_location()</b><br/>Untuk menyimpan data lokasi ke dalam basis data</p> <p><b>+getLocByName()</b><br/>Untuk mengambil data lokasi dari basis data berdasarkan nama lokasi</p> <p><b>+getDataLocByName()</b><br/>Untuk mengambil data lokasi dari basis data berdasarkan nama lokasi</p> <p><b>+getDataDetailLocByName()</b><br/>Untuk mengambil data detail lokasi dari basis data berdasarkan nama lokasi</p> <p><b>+getLocations()</b><br/>Untuk mengambil data lokasi dari basis data</p> <p><b>-setId_location()</b><br/>Prosedur untuk menginisialisasi atribut id_location</p> <p><b>-setLocation_name()</b><br/>Prosedur untuk menginisialisasi atribut location_name</p> <p><b>-setAddress</b><br/>Prosedur untuk menginisialisasi atribut address</p> <p><b>-setDescription</b><br/>Prosedur untuk menginisialisasi atribut description</p> <p><b>-setContact</b><br/>Prosedur untuk menginisialisasi atribut contact</p> <p><b>-setEmail</b></p> |            |

Prosedur untuk menginisialisasi atribut email  
**-setCategory**  
 Prosedur untuk menginisialisasi atribut category  
**-setLatitude**  
 Prosedur untuk menginisialisasi atribut latitude  
**-setLongitude**  
 Prosedur untuk menginisialisasi atribut longitude  
**-getId\_location()**  
 Fungsi untuk mendapatkan nilai dari atribut id\_location  
**-getLocation\_name()**  
 Fungsi untuk mendapatkan nilai dari atribut location\_name  
**-getAddress**  
 Fungsi untuk mendapatkan nilai dari atribut address  
**-getDescription**  
 Fungsi untuk mendapatkan nilai dari atribut description  
**-getContact**  
 Fungsi untuk mendapatkan nilai dari atribut contact  
**-getEmail**  
 Fungsi untuk mendapatkan nilai dari atribut email  
**-getCategory**  
 Fungsi untuk mendapatkan nilai dari atribut category  
**-getLatitude**  
 Fungsi untuk mendapatkan nilai dari atribut latitude  
**-getLongitude**  
 Fungsi untuk mendapatkan nilai dari atribut longitude

#### 2.2.3.15 *Specific Design Class comments*

| comments                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <<entity>> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>-id_comment : String</b><br>Atribut ini untuk menyimpan data id komentar<br><b>-id_location : String</b><br>Atribut ini untuk menyimpan data id lokasi<br><b>-id_user : String</b><br>Atribut ini untuk menyimpan data id pengguna<br><b>-comment : String</b><br>Atribut ini untuk menyimpan data komentar                                                                                                                                                                                                    |            |
| <b>+comments()</b><br>Default konstruktor dari kelas comments<br><b>+save_comment()</b><br>Untuk menyimpan data komentar ke dalam basis data<br><b>+getCommentByIdLoc()</b><br>Untuk mengambil data komentar dari basis data berdasarkan id lokasi<br><b>-setId_comment</b><br>Prosedur untuk menginisialisasi atribut id_comment<br><b>-setId_location</b><br>Prosedur untuk menginisialisasi atribut id_location<br><b>-setId_user</b><br>Prosedur untuk menginisialisasi atribut id_user<br><b>-setComment</b> |            |

Prosedur untuk menginisialisasi atribut comment  
**-getId\_comment**  
 Fungsi untuk mendapatkan nilai dari atribut id\_comment  
**-getId\_location**  
 Fungsi untuk mendapatkan nilai dari atribut id\_location  
**-getId\_user**  
 Fungsi untuk mendapatkan nilai dari atribut id\_user  
**-getComment**  
 Fungsi untuk mendapatkan nilai dari atribut comment

#### 2.2.3.16 *Specific Design Class googleApi*

|                                                                                    |                               |
|------------------------------------------------------------------------------------|-------------------------------|
| <b>googleApi</b>                                                                   | <b>&lt;&lt;entity&gt;&gt;</b> |
| <b>-latitude : Double</b><br>Atribut untuk menyimpan data latitude suatu lokasi    |                               |
| <b>-longitude : Double</b><br>Atribut untuk menyimpan data longitude suatu lokasi  |                               |
| <b>+getLatitude() : Double</b><br>Untuk memanggil data koordinat latitude lokasi   |                               |
| <b>+getLongitude() : Double</b><br>Untuk memanggil data koordinat longitude lokasi |                               |

### 3 Perancangan Data

#### 3.1 Dekomposisi Data

##### 3.1.1 Deskripsi Entitas users

| Nama      | Tipe    | Panjang | Keterangan                              |
|-----------|---------|---------|-----------------------------------------|
| ID_USER   | Integer | -       | Id pengguna, <i>Primary key, Unique</i> |
| USER_NAME | Varchar | 35      | Username dari pengguna                  |
| PASSWORD  | Varchar | 10      | Password dari pengguna                  |
| EMAIL     | Varchar | 25      | Email dari pengguna                     |
| IS_ADMIN  | Boolean | -       | Status role                             |

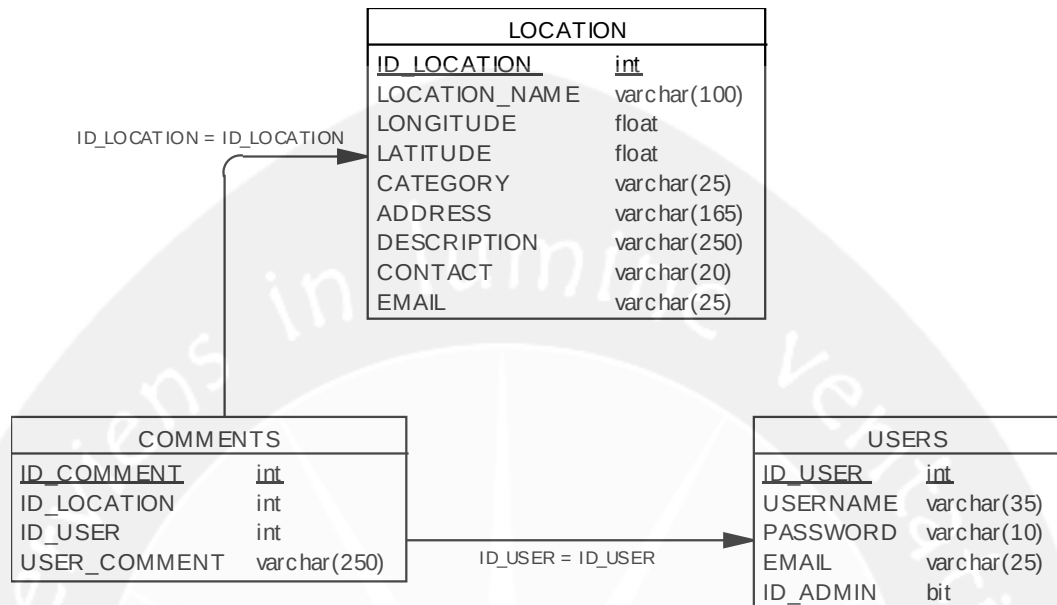
##### 3.1.2 Deskripsi Entitas locations

| Nama          | Tipe    | Panjang | Keterangan                            |
|---------------|---------|---------|---------------------------------------|
| ID_LOCATION   | Integer | -       | Id lokasi, <i>Primary key, Unique</i> |
| LOCATION_NAME | Varchar | 100     | Nama lokasi                           |
| ADDRESS       | Varchar | 165     | Alamat lokasi                         |
| DESCRIPTION   | Varchar | 250     | Deskripsi lokasi                      |
| CONTACT       | Varchar | 20      | Kontak dari lokasi/venue              |
| EMAIL         | Varchar | 25      | Email dari lokasi/venue               |
| LONGITUDE     | Float   | -       | Koordinat longitude lokasi            |
| LATITUDE      | Float   | -       | Koordinat latitude lokasi             |
| CATEGORY      | Varchar | 25      | Kategori dari lokasi yang ada         |

##### 3.1.3 Deskripsi Entitas comments

| Nama        | Tipe    | Panjang | Keterangan                              |
|-------------|---------|---------|-----------------------------------------|
| ID_COMMENT  | Integer | -       | Id komentar, <i>Primary Key, Unique</i> |
| ID_LOCATION | Integer | -       | Id lokasi, <i>Foreign Key</i>           |
| ID_USER     | Integer | -       | Id Pengguna, <i>Foreign Key</i>         |
| COMMENT     | Varchar | 250     | Komentar terhadap lokasi                |

### 3.2 Physical Data Model



Gambar 13. Physical Data Model

## 4 Perancangan Antarmuka

### 4.1 Antarmuka Halaman Login Admin

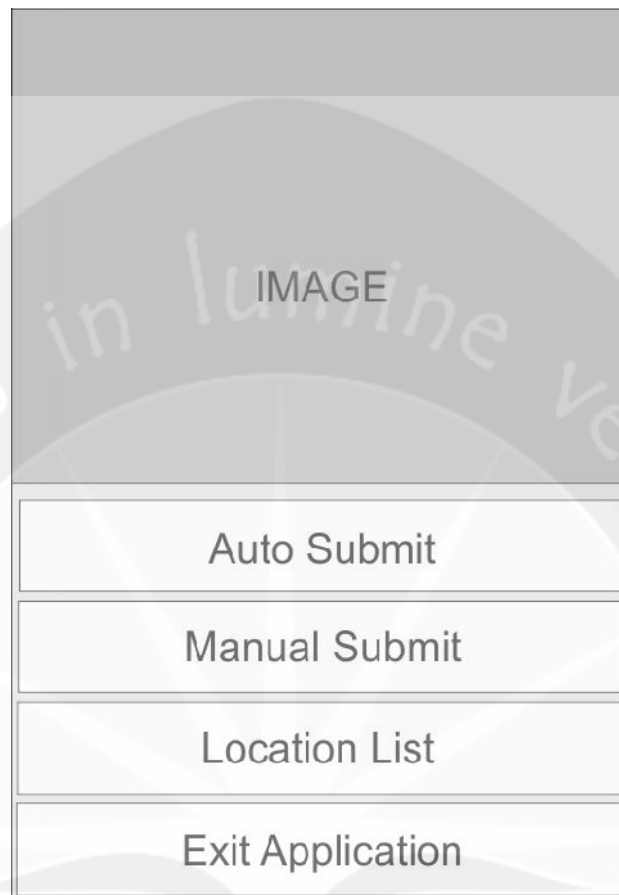


The image shows a mockup of an admin login page. It features a title bar at the top with the text "LOGIN ADMIN". Below the title bar, there are two text input fields. The first field is labeled "username" and the second field is labeled "password". Below these fields is a button labeled "Sign In". The entire form is set against a background that includes a large, faint watermark of a university crest with the motto "serviens lumine veritatis".

**Gambar 14. Rancangan Antarmuka Login Admin**

Antarmuka gambar 14 adalah rancangan antarmuka ketika administrator ingin melakukan login kedalam aplikasi pengelolaan Joker. Administrator diminta mengisi username dan passwordnya pada textfields yang tersedia untuk bisa login. Jika username dan password benar, maka pengguna akan masuk ke form beranda. Jika salah, maka sistem akan menampilkan pesan kesalahan.

#### 4.2 Antarmuka Beranda Admin



**Gambar 15 Rancangan Antarmuka Beranda**

Antarmuka gambar 15 adalah rancangan antarmuka beranda, ketika administrator masuk pertama kali kedalam sistem. Dari antarmuka ini, administrator bisa mulai memilih menu-menu yang disediakan aplikasi. Menu yang disediakan di halaman beranda antara lain checkin submit dan manual submit untuk menambahkan data lokasi. Sedangkan menu location list digunakan untuk menampilkan list data lokasi yang telah ada sebelumnya.

#### 4.3 Antarmuka Submit Location

location name

address

description

longitude

latitude

email

contact

category

☐ RESTAURANTS

☐ ATTRACTIONS

☐ HOTEL

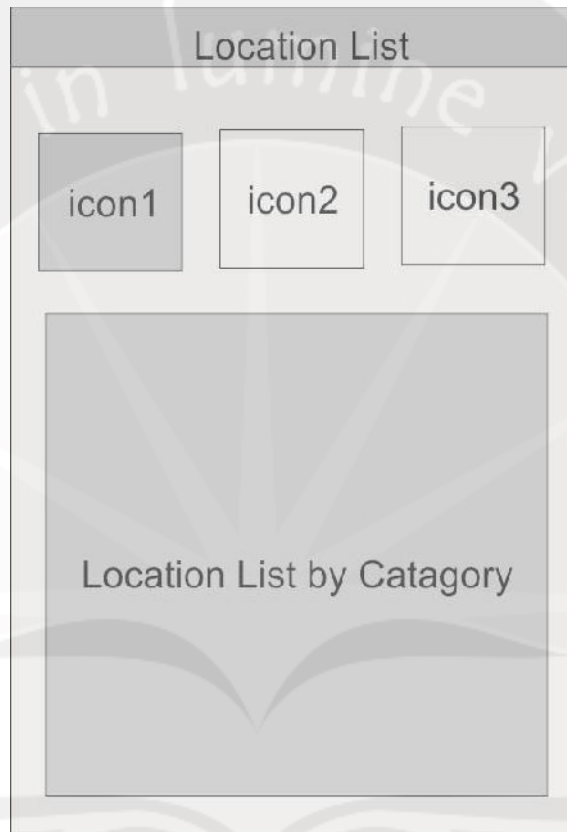
ADD RETRY CANCEL

**Gambar 16 Rancangan Antarmuka Submit**

Antarmuka gambar 16 adalah rancangan antarmuka submit yang dilakukan administrator. Pada halaman submit ini administrator dapat memilih untuk memasukkan data koordinat(longitude dan latitude) secara otomatis maupun manual. Jika menghendaki memasukkan data koordinat lokasi secara otomatis maka pada halaman

beranda admin harus masuk ke menu check in submit sedangkan jika ingin memasukkan data secara manual maka pada halam beranda administrator harus memilih menu manual submit.

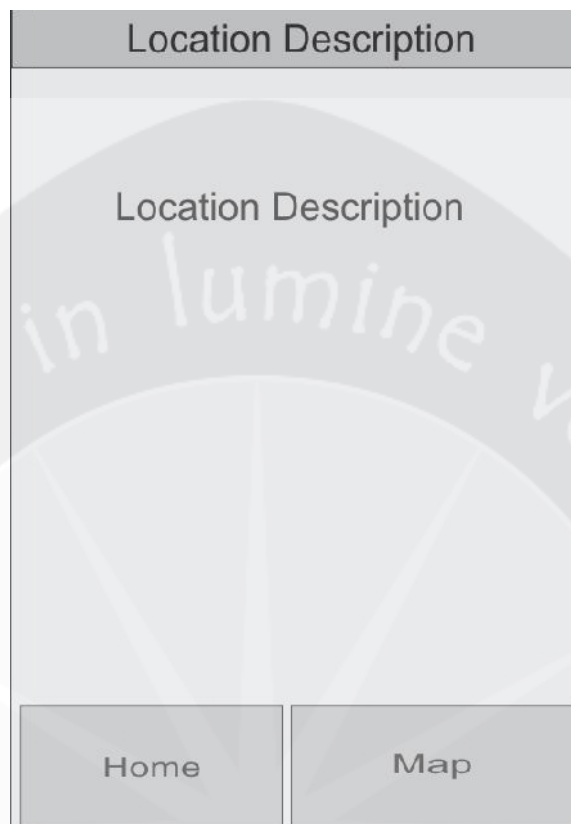
#### 4.4 Antarmuka List Lokasi



**Gambar 17 Rancangan Antarmuka List Lokasi**

Antarmuka gambar 17 ini adalah rancangan antarmuka untuk melihat daftar lokasi yang telah dikelompokkan berdasarkan katagori. Katagori yang ada adalah hotels/tempat menginap, attractions/tempat wisata, dan restaurants/tempat makan.

#### 4.5 Antarmuka Deskripsi Lokasi (admin)



**Gambar 18 Rancangan Antarmuka Deskripsi Lokasi**

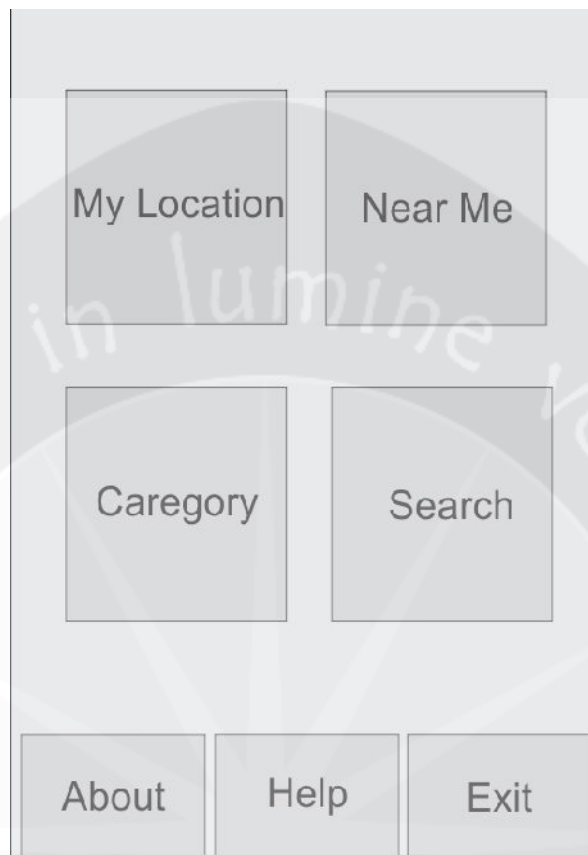
Antarmuka gambar 18 ini adalah rancangan antarmuka untuk menampilkan detail venue yang telah tersimpan dalam basis data. Pada halaman deskripsi lokasi ini terdapat 4 menu yaitu:

1. Home, menu ini sebagai *shortcut* untuk menuju halaman beranda tanpa harus kembali ke halaman location list.
2. Map, menu ini akan menampilkan detail lokasi yang disajikan dalam peta google. Rancangan antarmuka untuk menampilkan peta tersebut adalah seperti gambar 19:



**Gambar 19 Rancangan Antarmuka detail lokasi dalam peta**

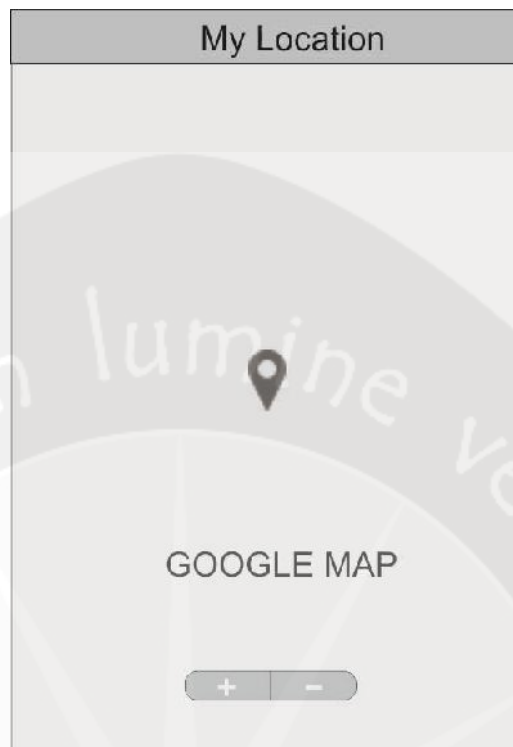
#### 4.6 Antarmuka Halaman Utama User



**Gambar 20 Perancangan Antarmuka Halaman Utama User**

Antarmuka gambar 20 adalah rancangan antarmuka halaman utama yang menampilkan 4 buah menu. Menu yang disediakan adalah:

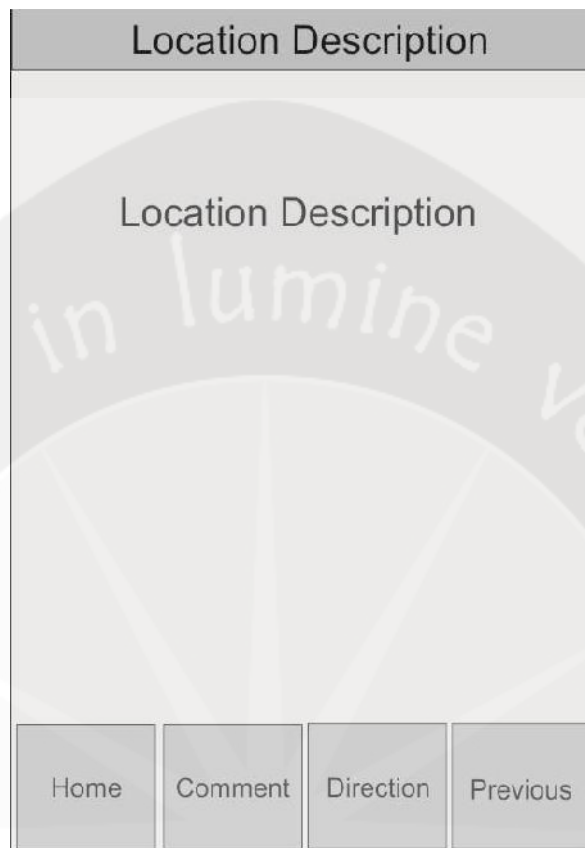
1. Catagories, menu ini memungkinkan user untuk melihat data lokasi venue yang telah dikelompokkan berdasarkan katagori. Rancangan antarmuka untuk menampilkan category list tersebut adalah seperti gambar 17.
2. My Location, menu ini memungkinkan user untuk melihat lokasi keberadaannya dalam peta google. Rancangan antarmuka untuk menampilkan peta keberadaan user adalah sebagai berikut,



**Gambar 21 Perancangan Antarmuka Peta My Location**

3. Help, menu ini digunakan untuk membantu user yang baru pertama kali menggunakan aplikasi joker. Pada halaman ini berisi tutorial menggunakan aplikasi Joker.
4. Credit, menu ini akan menampilkan data seputarpembangunan aplikasi Joker.

#### 4.7 Antarmuka Halaman Deskripsi Lokasi (User)

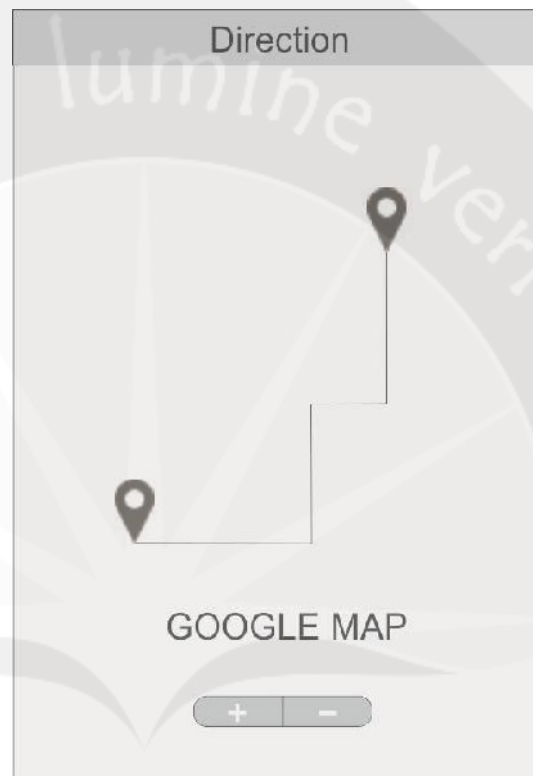


**Gambar 22 Perancangan Antarmuka Halaman Deskripsi Lokasi**

Antarmuka gambar 22 adalah ini adalah rancangan antarmuka untuk menampilkan detail venue yang telah tersimpan dalam basis data. Pada halaman deskripsi lokasi ini terdapat 4 menu yaitu.

1. Home, menu ini sebagai *shortcut* untuk menuju halaman beranda tanpa harus kembali ke halaman location list.
2. Comment, menu ini akan menampilkan form yang memungkinkan user untuk menambahkan komentar pada lokasi yang telah diseleksi. Untuk menambahkan komentar user harus login terlebih dahulu.

3. Direction, menu ini memungkinkan user untuk melihat lokasi keberadaannya dalam peta dan memandu dalam mencapai lokasi yang diinginkan. Rancangan antarmuka direction adalah sebagai berikut,



**Gambar 23 Rancangan Antarmuka Get Direction**

4. Previous, menu ini .memungkinkan user untuk kembali ke halaman sebelumnya.

#### 4.8 Antarmuka Login User

LOGIN USER

username

password

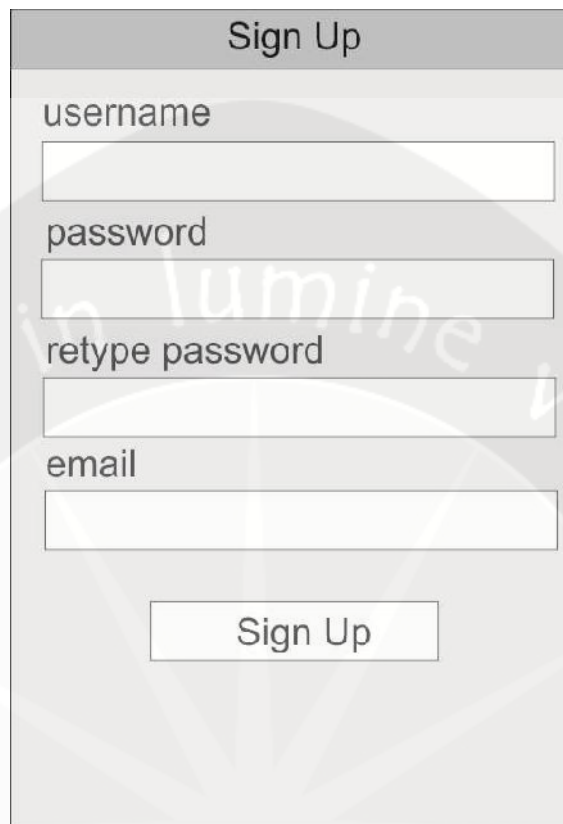
Sign Up

Sign In

**Gambar 24 Perancangan Antarmuka Login User**

Antarmuka gambar 24 adalah rancangan antarmuka ketika user ingin menambahkan komentar ke dalam lokasi yang telah diseleksi. Antarmuka aplikasi meminta masukan pengguna berupa username dan password. Setelah pengguna memasukan data-data yang diperlukan, sistem memeriksa bilamana terjadi kesalahan pada inputan. Jika terjadi, maka sistem akan menampilkan pesan kesalahan dan meminta pengguna menginputkan ulang data yang dimaksud. Jika username dan password tidak ditemukan di dalam basis data maka user akan diminta untuk sign up terlebih dahulu. Jika semua data valid, maka sistem akan membuka form komentar untuk diisi user.

#### 4.9 Antarmuka Sign Up



The image shows a 'Sign Up' form interface. It has a title bar at the top labeled 'Sign Up'. Below the title bar, there are four input fields: 'username', 'password', 'retype password', and 'email'. Each field is represented by a text label followed by a rectangular input box. At the bottom of the form, there is a 'Sign Up' button. The entire form is set against a light gray background with a faint watermark of a circular emblem containing the text 'serviens in lumine veritatis'.

**Gambar 25 Perancangan Antarmuka Sign Up**

Antarmuka gambar 25 adalah rancangan antarmuka ketika user belum mempunyai akun untuk login. Antarmuka aplikasi meminta masukan user berupa username, password, dan email. Setelah pengguna memasukkan data-data yang diperlukan, sistem memeriksa bilamana terjadi kesalahan pada inputan. Jika terjadi, maka sistem akan menampilkan pesan kesalahan dan meminta pengguna menginputkan ulang data yang dimaksud. Jika semua data valid, maka sistem akan menyimpan ke dalam basis data. Setelah data user tersimpan, maka login ke dalam aplikasi untuk menambahkan komentar pada lokasisudah dapat dilakukan.

#### 4.10 Antarmuka Tambah Komentar



The image shows a user interface for adding a comment. It consists of a header bar with the text "Add Comment". Below the header is a label "Comment" followed by a large rectangular text input area. At the bottom of the input area is a button labeled "Add Comment".

**Gambar 26 Perancangan Antarmuka Add Comment**

Antarmuka gambar 26 adalah rancangan antarmuka ketika user ingin menambahkan komentar pada lokasi yang telah dipilih. Untuk menambahkan komentar user harus memiliki akun dan login dahulu ke dalam sistem.