

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Peralatan Pendukung**

Dalam penelitian ini dibutuhkan beberapa alat untuk mendukung berjalannya perancangan dan implementasi aplikasi.

##### **3.1.1 Perangkat Lunak**

Perangkat lunak yang digunakan dalam mengembangkan perancangan basis data pada aplikasi Pariwisata JogjaKuy antara lain:

1. *Operating System Windows 8.0*
2. *Netbeans 8.1*
3. *MySql*
4. *Microsoft Visio*

##### **3.1.2 Perangkat Keras**

Perangkat keras yang digunakan dalam mengembangkan perancangan basis data aplikasi jogjakuy berbasis *android* antara lain:

1. *CPU Intel Core i7-3537U 2.00 GHz*
2. *Memory RAM 8 GB*
3. *Nvidia GeForce GT 540M*

#### **3.2 Bahan Penelitian**

Teknik yang digunakan penulis dalam melakukan analisis dan pengumpulan data atau informasi yang dibutuhkan selanjutnya adalah teknik penelitian, yaitu mencari informasi yang berhubungan menggunakan jurnal-jurnal, buku referensi

dan sumber *internet*. Setelah melakukan pencarian dan pengumpulan data kemudian dilakukan analisis pada data dan informasi tersebut yang bertujuan untuk menentukan data dan informasi yang tepat untuk dimasukkan ke dalam *database*.

### **3.3 Analisis Kebutuhan**

#### **3.3.1 Analisis Kebutuhan Fungsional**

Berikut ini adalah kebutuhan-kebutuhan yang memiliki keterkaitan langsung dengan sistem. Kebutuhan fungsional dari aplikasi ini meliputi:

1. Dapat menambahkan, menghapus dan merubah konten objek wisata.
2. Dapat menambahkan, menghapus dan merubah konten Hotel.
3. Dapat menambahkan, menghapus dan merubah konten Jogja sejarah.
4. Dapat menambahkan, menghapus dan merubah konten Kuliner.
5. Dapat menambahkan letak lokasi konten .

#### **3.3.2 Analisis Kebututuhan Non-Fungsional**

Berikut ini adalah kebutuhan yang tidak secara langsung terkait dengan fitur tertentu di dalam sistem. Kebutuhan nonfungsional dari aplikasi ini meliputi:

1. Keamanan: Mengelola sistem data dalam mengganti, menambah dan menghapus data konten hanya dapat dilakukan oleh *Administrator*.
2. Bahasa Indonesia: Aplikasi ditampilkan dengan Bahasa Indonesia

#### **3.3.3 Analisis Kebutuhan Data**

Data yang dikumpulkan berdasarkan 1 Kotamadya 4 Kabupaten yang ada di Propinsi D.I Yogyakarta, yaitu Kabupaten Bantul, Sleman, Kulon Progo, Gunung Kidul dan Kota Yogyakarta. Data yang diambil yaitu:

1. Data koordinat lokasi Wisata, lokasi Sejarah, lokasi kuliner dan Hotel.

2. Gambar lokasi Wisata, lokasi Sejarah, lokasi kuliner dan Hotel.
3. Data Detail Informasi tiap lokasi Wisata, lokasi Sejarah, lokasi kuliner dan Hotel.

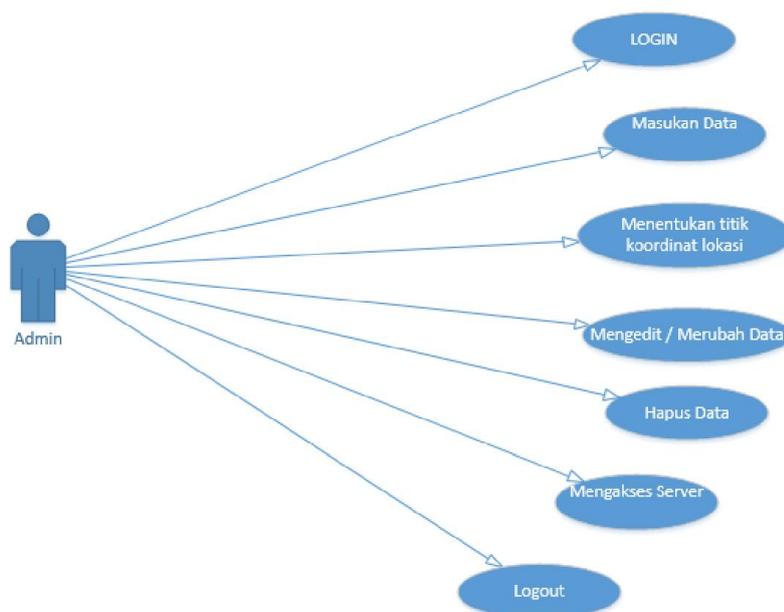
### **3.4 Rancangan Proses**

Pada tahap ini diperlukan untuk membangun desain dari sistem yang akan dikembangkan. Setelah dilakukan analisa, terkumpulah kebutuhan informasi yang dibutuhkan untuk membangun sistem informasi. Untuk pemodelan sistem pada sebuah *database* diperlukan sebuah model analisis terstruktur, yaitu dengan menggunakan *Use Case Diagram*, *Activity Diagram*, dan *ER Diagram*.

Pada sistem informasi ini, model yang dikembangkan adalah:

### 3.4.1 Use Case Diagram

*Use case diagram* merupakan salah satu bagian dari *Unified Markup Language (UML)*, *use case diagram* adalah suatu diagram yang menggambarkan suatu sistem dan bagaimana sistem tersebut bekerja. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Use case diagram* menggambarkan orang yang akan mengoperasikan atau berinteraksi dengan sistem. Berikut adalah gambaran *Use case diagram* Perancangan Basis Data pada aplikasi pariwisata JogjaKuy dapat dilihat pada Gambar 3.1.

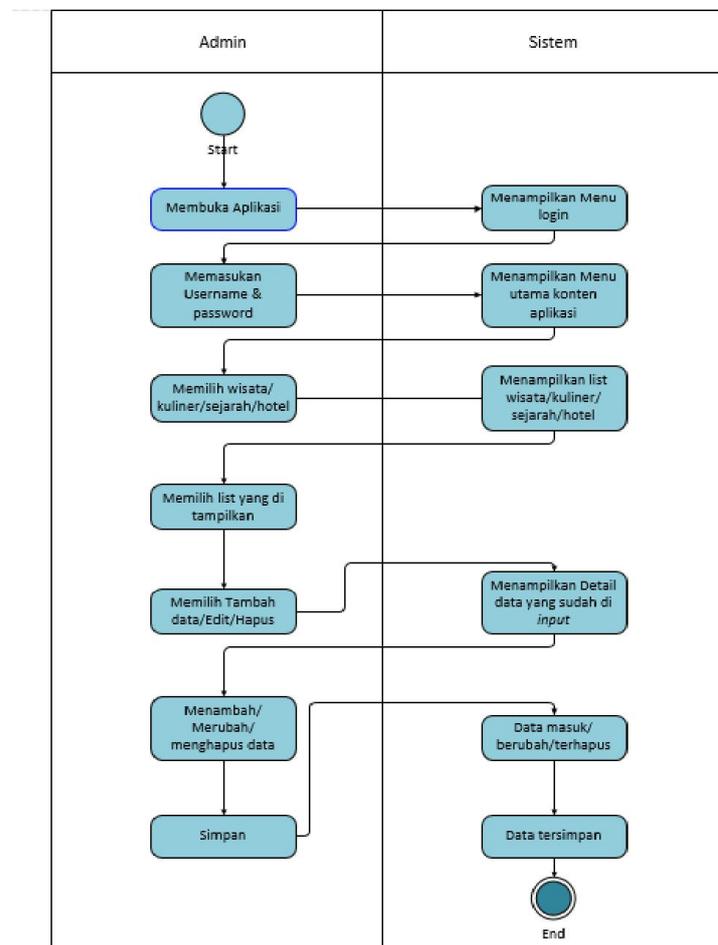


**Gambar 3.1** Use case Diagram (*Administrator*)

*Use case* diatas menunjukkan bahwa *Administrator* dapat membuka aplikasi,menambah data,menghapus data,mengedit data dan menentukan titik koordinat lokasi.

### 3.4.2 Activity Diagram

*Activity diagram* menggambarkan berbagai alur aktivitas yang ada dalam perangkat lunak yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana alur tersebut berakhir. Berikut merupakan *activity diagram* Perancangan Basis Data pada aplikasi pariwisata JogjaKuy dapat dilihat pada Gambar 3.2 :



**Gambar 3.2** Actifity Diagram

Berikut adalah keterangan dari gambar 3.2:

1. Pengguna membuka aplikasi *admin* JogjaKuy.
2. Sistem merespon dengan menampilkan menu *login*.

3. *Admin* harus memasukan *email & password*.
4. Sistem merespon dengan menampilkan menu utama konten aplikasi.
5. *Admin* memilih ingin masuk ke konten wisata, kuliner, sejarah dan hotel.
6. Sistem merespon dengan menampilkan *list* wisata, kuliner, sejarah dan hotel.
7. *Admin* harus mengklik *list* wisata terlebih dahulu.
8. *Admin* memilih tambah data, edit dan hapus.
9. Sistem merespon dengan menampilkan detail data yang sebelumnya sudah di *input*.
10. *Admin* mengisi data pada *box* yang sudah disediakan, tambah data, edit dan hapus.
11. *Admin* mengklik simpan.
12. Sistem merespon dengan data bertambah, data berubah atau data terhapus.
13. Sistem merespon dengan menyimpan data.
14. Selesai.

### **3.5 Metode Perancangan dan Implementasi Database**

Dalam penelitian ini metode perancangan dan implementasi meliputi Perancangan Database konseptual (Perancangan *Entity Relationship Diagram*), Perancangan Database Logikal (Perancangan *Relational Database*), Perancangan Database Fisikal (struktur-struktur penyimpanan pada *file-file database*).

### 3.6 Perancangan *Database Level Konseptual*

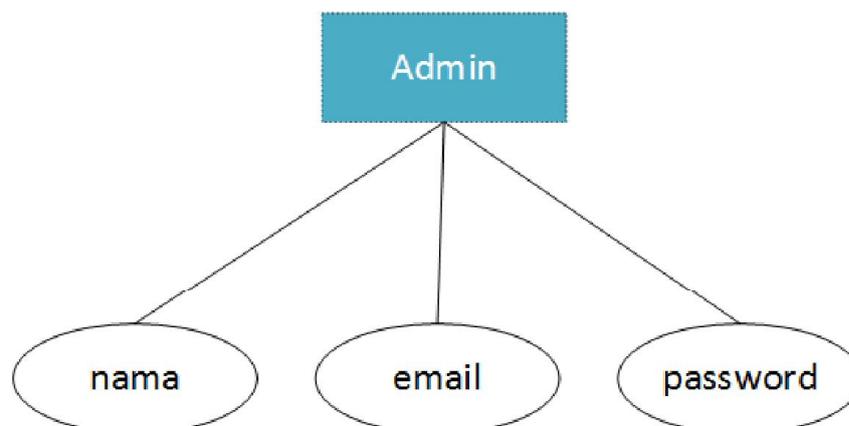
Tahap selanjutnya yang dilakukan dalam melakukan perancangan sebuah *database* adalah metode perancangan *database level* konseptual, perancangan *level* konseptual ini bertujuan untuk mengecek kebutuhan-kebutuhan *user*, Batasan-batasan dan hubungannya. Perancangan skema konseptual ini biasa menggunakan model *ERD (Entity Relationship Diagram)*.

#### 3.6.1 Entitas dan *Attributes*

Pada tahap ini menentukan dan mengidentifikasi entitas yang akan digunakan. Entitas didapatkan berdasarkan data yang telah dikumpulkan dari dokumentasi yang ada. Berikut adalah *entitas* dan *atributes* dari Perancangan basis data pada aplikasi jogjakuy berbasis *android*.

##### 1. Entitas *Admin*

Entitas *Admin* terdiri dari 3 atribut diantaranya adalah *nama*, *email*, dan *password*. Entitas *Admin* tersebut dapat dilihat pada Gambar 3.3.



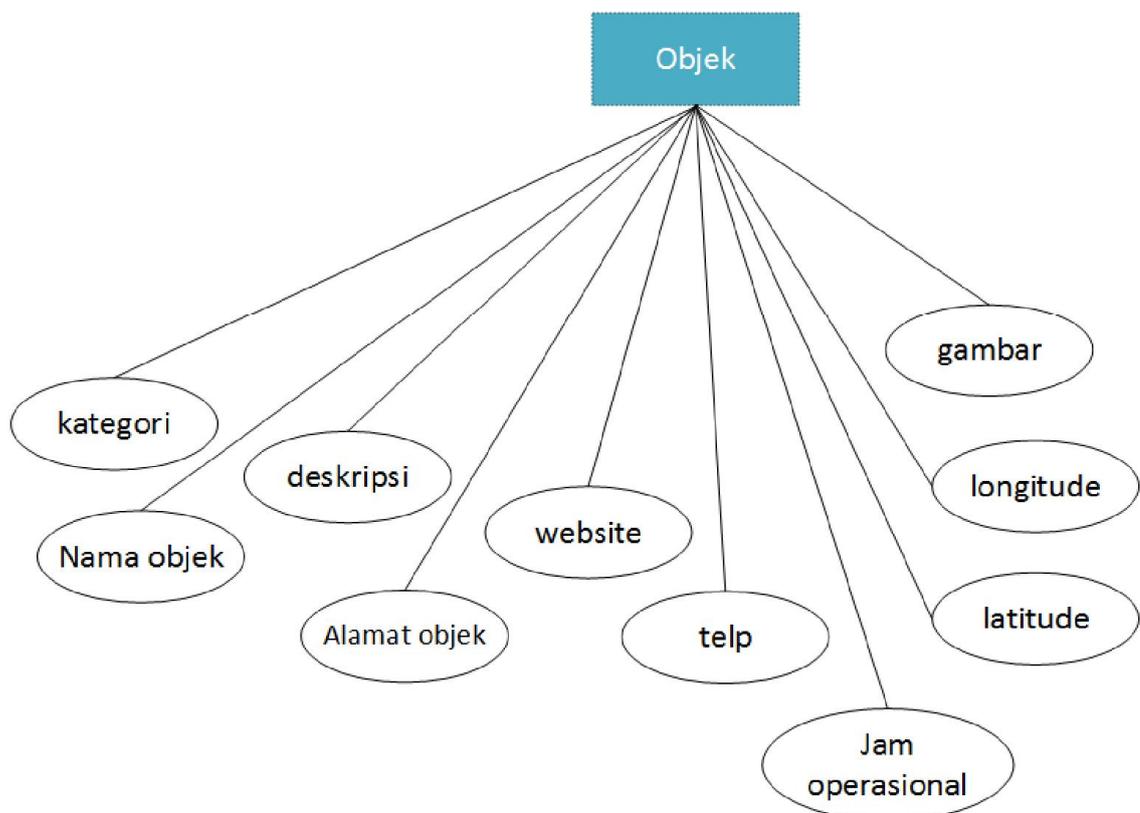
Gambar 3 3 Entitas *Admin*

Berikut penjelasan entitas *Admin* pada Gambar 3.3 adalah sebagai berikut:

1. Atribut nama, *email* dan *password* perlu dimasukkan ke dalam entitas *Admin* karena berfungsi untuk mengetahui identitas dari masing-masing *admin*. Atribut *email* dan *password* perlu dimasukkan ke dalam entitas *Admin* karena berfungsi untuk *login admin*.

## 2. Entitas Objek

Entitas objek terdiri dari 10 atribut diantaranya adalah kategori, nama objek, deskripsi, alamat objek, *website*, telp, jam operasional, *latitude*, *longitude* dan gambar. Entitas objek tersebut dapat dilihat pada Gambar 3.4.



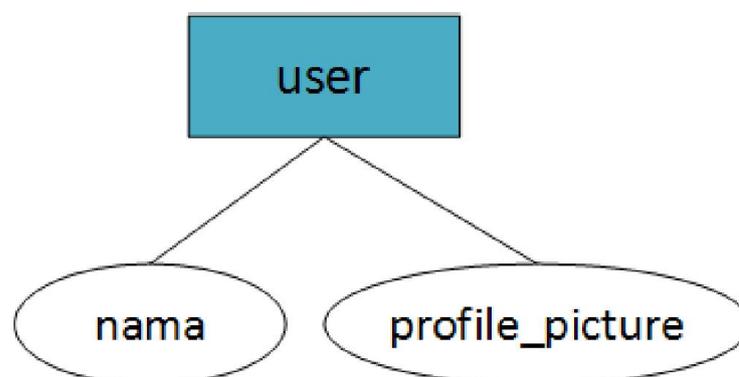
**Gambar 3.4** Entitas objek

Berikut penjelasan entitas objek dalam gambar 3.4 adalah sebagai berikut:

1. atribut kategori, nama objek, deskripsi, alamat objek, *website*, telp, jam operasional, dan gambar perlu dimasukkan ke dalam entitas objek karena berfungsi sebagai penjelasan atau detail dari suatu objek. Atribut *latitude* dan *longitude* perlu di masukan ke dalam entitas objek karena berfungsi sebagai penentu titik *koordinat* letak objek tersebut.

### 3. Entitas *user*

Entitas *user* terdiri dari 2 atribut diantaranya adalah nama dan *profile\_picture*. Entitas *user* tersebut dapat dilihat pada Gambar 3.5.



Gambar 3 5 Entitas *user*

Penjelasan entitas gambar dalam gambar 3.5 adalah sebagai berikut:

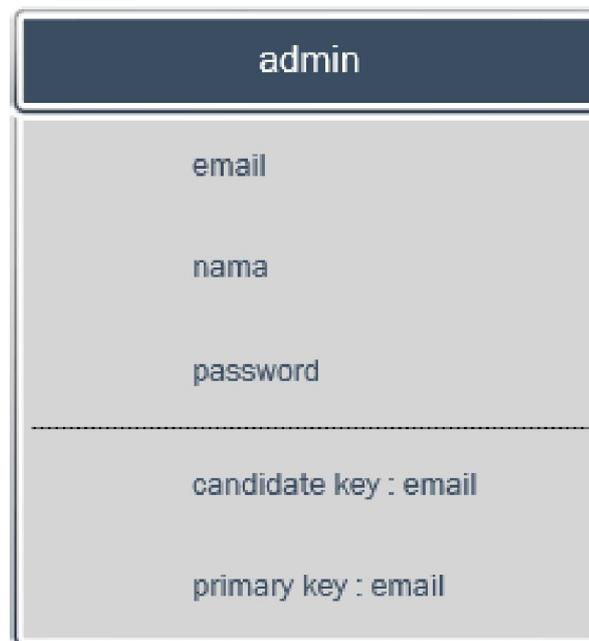
1. Atribut nama dan *profile\_picture* perlu di masukan ke dalam entitas *user* karena berfungsi sebagai identitas *user* tersebut.

### 3.6.2 *Primary Key*

Pada tahap ini menentukan dan mengidentifikasi *primary key* dari setiap entitas. *Primary Key* didapatkan dengan cara mengidentifikasi setiap atribut yang ada dalam setiap entity, apabila terdapat atribut yang memenuhi persyaratan yaitu *unique identifier* dan *non redundancy*, atribut tersebut menjadi *candidate key* yang kemudian menjadi *primary key*. Setelah menentukan *entity* dan *attributes* langkah selanjutnya adalah menentukan *primary key* pada tiap *entity* tersebut. *Primary key* berfungsi untuk memastikan bahwa setiap baris data pada tabel bersifat unik (berbeda antara baris satu dengan baris yang lain). Hal ini sangat berguna pada saat akan menampilkan hasil pencarian data, pengurutan data, dan berbagai operasi query lainnya. Dengan memilih *primary key* proses tersebut dapat berlangsung lebih cepat.

#### 1. *Primary Key Entitas Admin*

Entitas *admin* memiliki *Primary Key* yaitu *id* seperti yang terlihat pada gambar 3.6.



**Gambar 3 6** *Primary Key entitas admin*

Pada entitas *admin* yang ditunjukkan pada gambar 3.3, sudah ada *attributes* yang bersifat unik, maka *attributes* email ini yang akan menjadi *primary key* dari entitas *admin*.

## 2. **Primary key entitas objek**

Entitas objek memiliki *Primary Key* yaitu id seperti yang terlihat pada gambar 3.7.

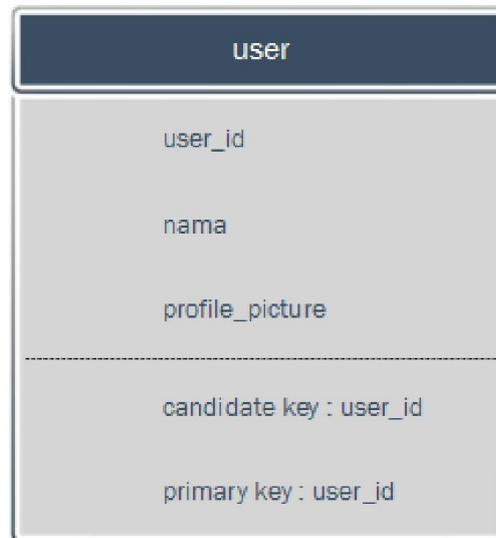


**Gambar 3. 7** *Primary key* entitas objek

Pada entitas objek yang ditunjukkan pada gambar 3.4, tidak ada *attributes* yang bersifat unik, maka perlu ditambahkan satu *attributes* lagi dengan nama id\_objek, *attributes* id\_objek ini yang akan menjadi *primary key* dari entitas *admin*.

### 3. *Primary key* entitas *user*

Entitas *user* memiliki *Primary Key* yaitu *user\_id* seperti yang terlihat pada gambar 3.8.

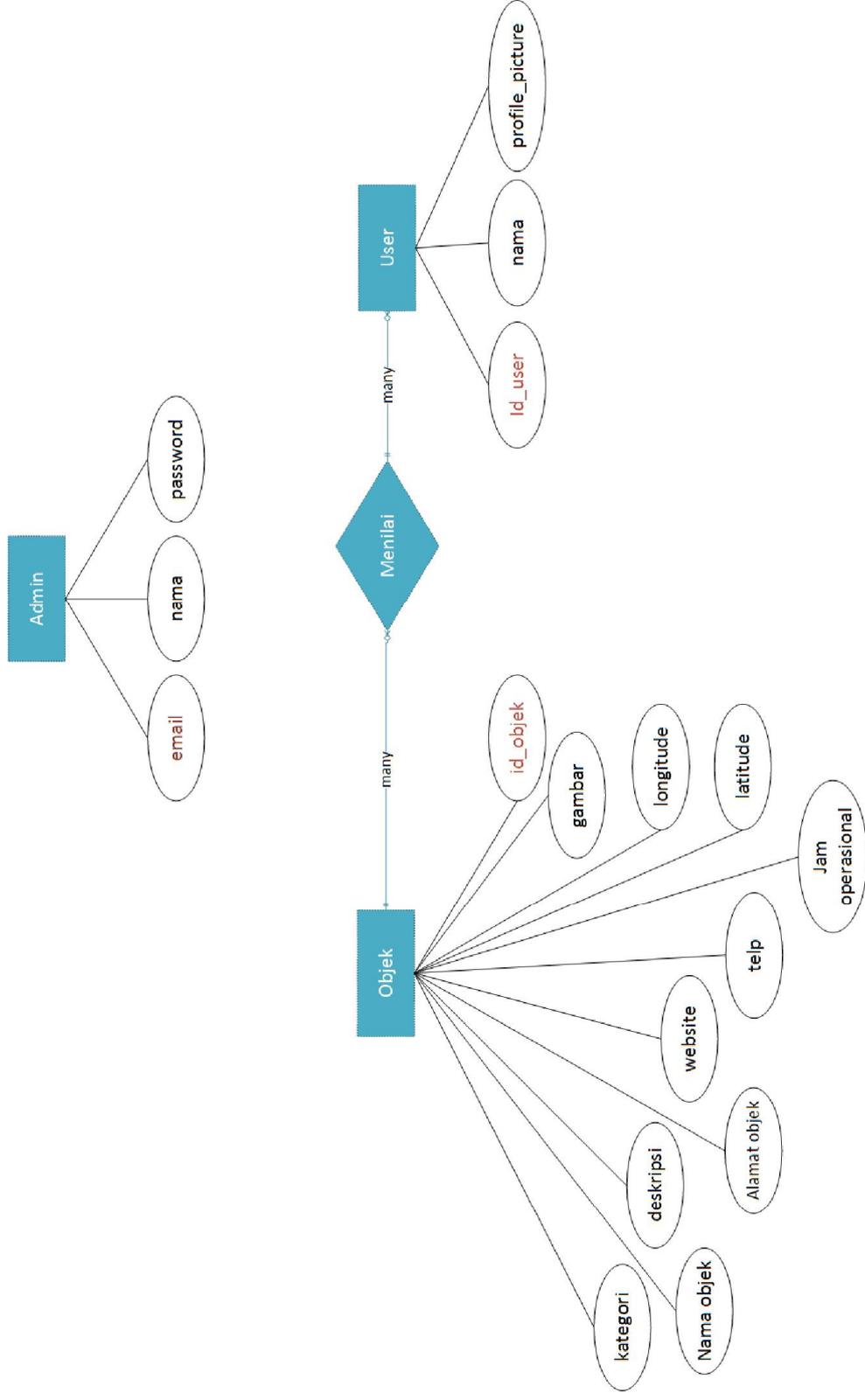


**Gambar 3.8** *Primary key* entitas user

Pada entitas *user* yang ditunjukkan pada gambar 3.5, tidak ada *attributes* yang bersifat unik, maka perlu ditambahkan satu *attributes* lagi dengan nama *user\_id*, attribute *user\_id* ini yang akan menjadi *primary key* dari entitas *user*.

#### 4. ER Diagram

Dalam perancangan *database* pada aplikasi jogjaKuy yang dibuat terdapat *entity admin*, *entity objek*, *entity favorit*, *entity user*, dan *entity rating*, . Dari semua *entity* tersebut membentuk sebuah ERD (*Entity Relationship Diagram*). ERD tersebut dapat dilihat pada gambar 3.9.



Gambar 3.9 ER Diagram

Penjelasan tentang ER *diagram* pada gambar 3.9:

1. Tabel *user* memiliki relasi dengan tabel objek. Relasi antar keduanya yaitu *many to many* yang artinya satu *user* bisa menilai banyak objek dan satu objek bisa dinilai banyak *user*.
2. Tabel *admin* tidak memiliki relasi dengan entitas manapun.

### **3.7 Perancangan *Database* Logikal**

Perancangan *database* level logikal merupakan tahapan untuk memetakan proses perancangan konseptual kedalam model *database* yang akan digunakan. Model yang digunakan dalam perancangan *database* ini adalah model relasi.

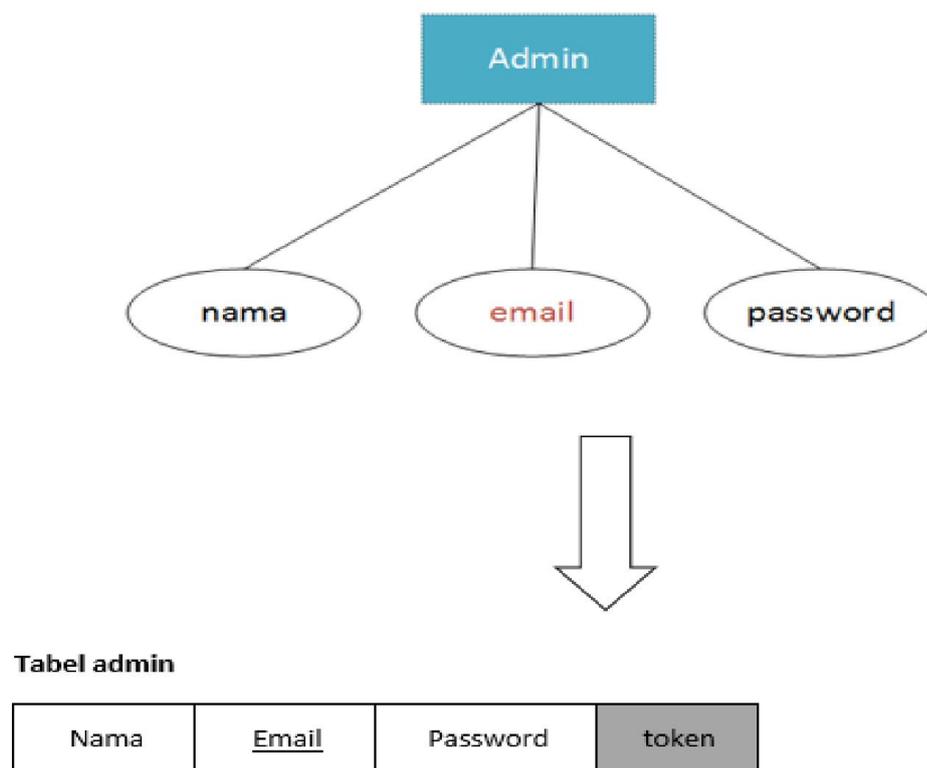
#### **3.7.1 *Konversi* ERD ke dalam Tabel**

*Konversi* ERD ke dalam bentuk tabel adalah proses memasukan *entity* menjadi nama tabel dan *attribute* menjadi nama kolom beserta dengan relasinya.

## 1. Entitas *Admin*

Penjelasan pada gambar 3.10 adalah:

1. Pada proses konversi *entity admin* ke dalam bentuk tabel pada gambar 3.10 dijelaskan bahwa *entity* menjadi nama tabel dan *attribute* menjadi nama kolom. Terdapat penambahan kolom pada tabel *admin* yaitu kolom *token* yang berfungsi sebagai keamanan pada *database*. Entitas *admin* disebut juga dengan nama entitas kuat karena tidak memiliki relasi dengan tabel manapun.



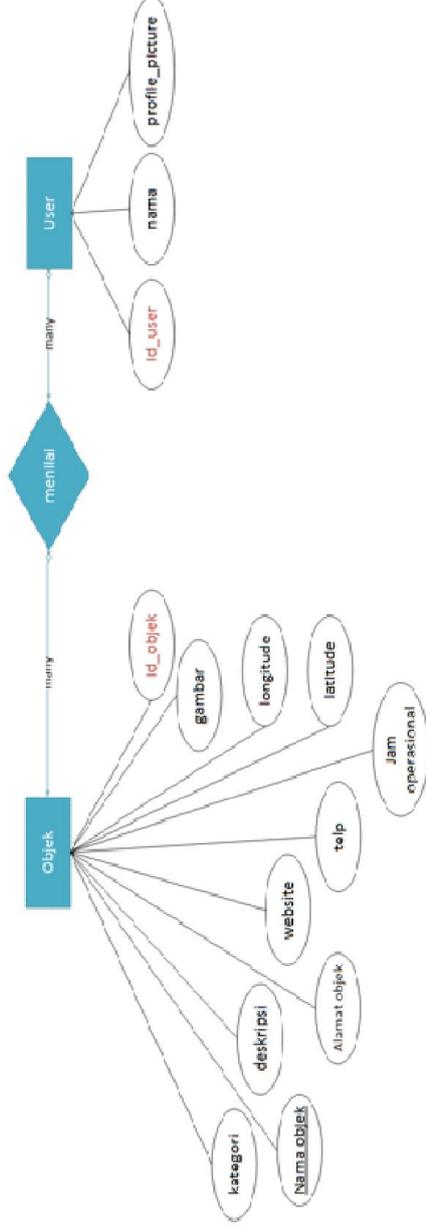
**Gambar 3.10** Konfersi entitas *admin*

## 2. Entitas objek dengan entitas *user*

Penjelasan pada gambar 3.11 adalah:

1. Pada *konversi* antara tabel objek dengan tabel *user* menghasilkan 2 tabel baru yaitu tabel *rating* dan tabel favorit. Fungsi dari tabel *rating* adalah untuk menunjukkan *rating* dari suatu objek yang telah *user* berikan, sedangkan pada tabel favorit berfungsi untuk menyimpan data objek yang telah *user* jadikan sebagai tempat favorit dari *user* itu sendiri.
2. Pada *konversi* antara tabel objek dengan tabel *user* menghasilkan 2 kolom baru pada tabel *user* yaitu kolom *token* dan *join\_date*. Fungsi kolom *token* adalah sebagai keamanan ketika *user login* menggunakan akun *google* yang mana data *token* didapatkan secara otomatis, sedangkan kolom *join\_date* berfungsi untuk mengetahui kapan *user login* ke dalam aplikasi.
3. Satu baris data pada tabel objek dapat dihubungkan ke satu baris atau lebih dari satu baris data pada tabel favorit dan satu baris data pada tabel *user* dapat dihubungkan ke satu atau lebih dari satu baris data pada tabel favorit. maka pada tabel favorit perlu ditambahkan kolom *id\_objek* dan *id\_user* yaitu sebagai *foreign key* dari tabel objek dan tabel *user* yang berfungsi sebagai penghubung antara tabel objek dengan tabel favorit dan tabel *user* dengan tabel favorit.

4. Satu baris data pada tabel objek dapat dihubungkan ke satu baris atau lebih dari satu baris data pada tabel *rating* dan satu baris data pada tabel *user* dapat dihubungkan ke satu atau lebih dari satu baris data pada tabel *rating*. Maka pada tabel *rating* perlu ditambahkan kolom *id\_objek* dan *id\_user* yaitu sebagai *foreign key* dari tabel objek dan tabel *user* yang berfungsi sebagai penghubung antara tabel objek dengan tabel *rating* dan tabel *user* dengan tabel *rating*.



Tabel Objek

<u>id_objek</u>	Kategori	<u>Nama objek</u>	Deskripsi	Alamat objek	Website	Telp	Jam	Latitude	Longitude	Gambar
-----------------	----------	-------------------	-----------	--------------	---------	------	-----	----------	-----------	--------

Tabel user

<u>id_user</u>	Nama user	Profile_picture	Token	login_date
----------------	-----------	-----------------	-------	------------

Tabel rating

<u>Id_rating</u>	rating	komentar	id_user	id_objek	Total_rating	Created_at
------------------	--------	----------	---------	----------	--------------	------------

Tabel Favorit

<u>Id_fav</u>	id_objek	id_user	Total_favorit	Created_at
---------------	----------	---------	---------------	------------

Gambar 3.11 Konversi entitas objek dengan entitas user

### 3.7.2 Kamus data *Database*

Kamus data (*data dictionary*), merupakan salah satu komponen kunci dari DBMS yang mencakup informasi mengenai struktur *database*. Setiap elemen data yang disimpan dalam *database* memiliki catatan kamus data yang mendeskripsikan elemen-elemen tersebut.

#### 1. Tabel *admin*

Pada tabel ini berisi daftar *admin* aplikasi jogjakuy yang sudah terdaftar dan bisa mengakses aplikasi *admin* tersebut.

**Tabel 3.1** Tabel *admin*

Nama Field	Tipe Data	Length field	Keterangan
<i>email</i>	<i>varchar</i>	10	Berfungsi untuk <i>login</i> sebagai email bersifat <i>primary key</i> . Contoh data ( <i>admin@yahoo.com</i> ).
nama	<i>varchar</i>	20	berfungsi sebagai identitas <i>admin</i> . Contoh data (Candria Putrajaya).
<i>password</i>	<i>varchar</i>	15	Berfungsi <i>login</i> sebagai password. Contoh data (qwerty123).
<i>token</i>	<i>varchar</i>	35	Berfungsi untuk keamanan <i>database</i> pada <i>server</i> . Contoh data (8a80f62b335bdb9c9fc84a000d33

## 2. Tabel objek

Pada tabel ini berisi daftar keterangan dari objek aplikasi jogjakuy.yang sudah di *input* ke dalam aplikasi.

*Tabel 3.2 Tabel Objek*

Field	Type	Lebar	Keterangan
id_objek	<i>int</i>	5	Berfungsi sebagai <i>primary key</i> . Contoh data (1,2,3,4,. . .).
kategori	<i>Int</i>	1	Berfungsi sebagai penentu kategori. Contoh data (1,2,3,4).
nama	<i>varchar</i>	35	Berfungsi untuk menyimpan nama dari suatu objek bersifat <i>Unique</i> . Contoh Data (Museum Pusat TNI AU Dirgantara Mandala).
deskripsi	<i>longtext</i>	-	Berfungsi untuk menyimpan deskripsi dari suatu objek.
alamat	<i>varchar</i>	100	Berfungsi untuk menyimpan alamat dari suatu objek. Contoh data (jl. Veteran, XT Square gedung Umar Kayam Lantai Baseman, No.150-151, Pandean, Umbulharjo, Daerah Istimewa Yogyakarta).
website	<i>Varchar</i>	20	Berfungsi untuk menyimpan <i>website</i>

Field	Type	Lebar	Keterangan
			dari salah satu kategori seperti Hotel. Contoh data (www.hotelgrage.com).
telp	<i>varchar</i>	15	Berfungsi untuk menyimpan nomor telfon dari suatu objek. Contoh data (+62274513063)
Jam_operasional	<i>varchar</i>	15	Berfungsi untuk menyimpan jam operasional dari suatu objek. Contoh data (21:00 - Habis, 21:00 – 22:00 WIB).
latitude	<i>double</i>	-	Berfungsi untuk menyimpan <i>latitude</i> koordnat dari suatu objek. Contoh data (-7.784888).
longitude	<i>double</i>	-	Berfungsi untuk menyimpan <i>longitude</i> koordnat dari suatu objek. Contoh data (110.371463).
gambar	<i>varchar</i>	15	Berfungsi untuk menyimpan gambar dari suatu objek. Contoh data (w1502953537553).

### 3. Tabel favorit

Pada tabel ini berisi daftar keterangan dari favorit aplikasi jogjakuy.yang sudah di *input* ke dalam aplikasi.

Tabel 3.3 Tabel Favorit

Field	Type	Lebar	Keterangan
id	<i>int</i>	3	Berfungsi sebagai <i>primary key</i> . Contoh data (1,2,3. . . ).
id_user	<i>varchar</i>	25	Berfungsi sebagai <i>foreign key</i> tabel <i>user</i> . Contoh data (106126458586923555680).
id_objek	<i>int</i>	5	Berfungsi sebagai <i>foreign key</i> tabel objek. Contoh data (1,2,3,4,5,.....).
Created_at	<i>bigint</i>	20	Berfungsi untuk menyimpan data kapan <i>user</i> memberikan favorit ketika memilih suatu objek sebagai tempat favorit. Contoh data (1501820224406).

#### 4. Tabel *user*

Pada tabel ini berisi daftar keterangan dari favorit aplikasi jogjakuy.yang sudah di *input* ke dalam aplikasi.

Tabel 3.4 Tabel *user*

Field	Type	Lebar	Keterangan
<i>user_id</i>	<i>varchar</i>	25	Berfungsi sebagai <i>primary key</i> . Contoh data (101018503486391959211).
nama	<i>varchar</i>	30	Berfungsi untuk menyimpan nama dari <i>user</i> yang telah <i>login</i> aplikasi jogjakuy. Contoh Data (fahri anindika sukarno).
profile_picture	<i>varchar</i>	130	Berfungsi untuk menyimpan foto dari <i>user</i> yang telah <i>login</i> aplikasi jogjakuy. Contoh data ( <a href="https://lh5.googleusercontent.com/-S6aRzhqj4Kg/AAAAAAAAAAI/">https://lh5.googleusercontent.com/-S6aRzhqj4Kg/AAAAAAAAAAI/</a> )

Field	Type	Lebar	Keterangan
			AAAAAAAAAAAA/AMp5VUpN8QTDger3mWyvUurSIDEPE5_Q/s96-c/photo.jpg).
Join_date	bigint	20	Berfungsi untuk mengetahui kapan <i>user</i> tersebut menggunakan aplikasi jogjakuy. Contoh data (1502467968595)
Token	varchar	35	Berfungsi untuk menyimpan <i>token</i> yang google berikan secara otomatis. Contoh data(1075bf22b72dcfc0d87ca2ab4c87f0a1)

## 5. Tabel rating

Pada tabel ini berisi daftar keterangan dari rating aplikasi jogjakuy.yang sudah di *input* ke dalam aplikasi.

Tabel 3.5 Tabel rating

Field	Type	Lebar	Keterangan
id	Int	3	Berfungsi sebagai <i>primary key</i> . Contoh data (1,2,3 s/d 999).
id_user	varchar	25	Berfungsi sebagai <i>foreign key</i> tabel <i>user</i> . Contoh data (106126458586923555680).
id_objek	int	5	Berfungsi sebagai <i>foreign key</i> tabel objek. Contoh data (1,2,3,4,5, . . . ).

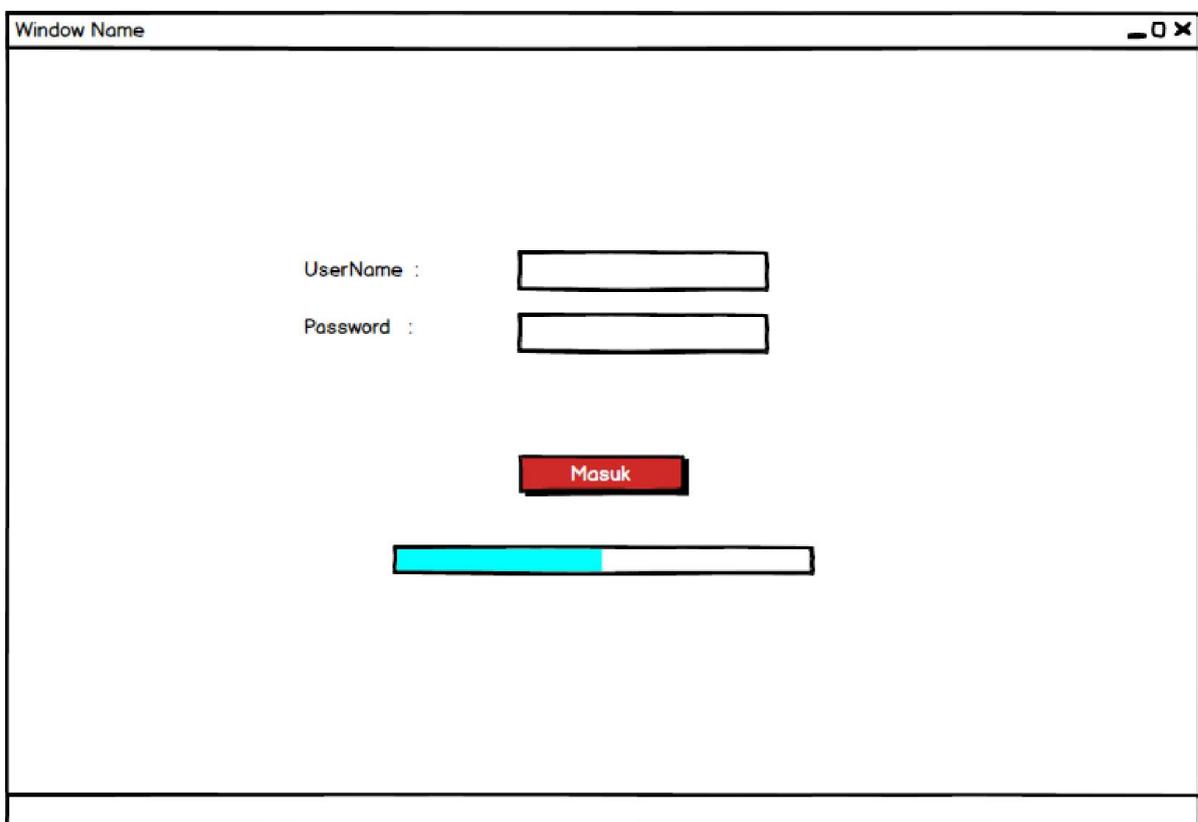
Field	Type	Lebar	Keterangan
rating	<i>Int</i>	1	Berfungsi untuk menyimpan rating yang <i>user</i> berikan pada salah satu objek. Contoh data (1,2,3,4,5)
komentar	longtext	-	Berfungsi untuk menyimpan komentar yang <i>user</i> berikan pada salah satu objek.
Created_at	<i>bigint</i>	20	Berfungsi untuk menyimpan data kapan <i>user</i> memberikan rating ketika memilih suatu objek sebagai tempat favorit. Contoh data (1501820224406).
Total_rating	Float	-	Berfungsi untuk menyimpan rata-rata hasil dari rating yang telah <i>user</i> berikan. Contoh data (3,5555).

### 3.8 Rancangan Antarmuka

Perancangan antarmuka merupakan tampilan dari suatu perangkat lunak yang berperan sebagai media komunikasi antara perangkat lunak dan pengguna. Perancangan ini merupakan sebuah penggambaran, perencanaan dan pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh. Perancangan antarmuka diharapkan memudahkan pengguna dalam melakukan proses *interaksi* terhadap sistem. Berikut adalah perancangan antar muka yang ada pada aplikasi Perancangan Basis Data pada Aplikasi JogjaKuy berbasis *Android* :

### 3.8.1 Antarmuka Menu *Login*

Halaman Menu *Login* merupakan halaman pertama saat pertama kali membuka aplikasi. Pada halaman ini *Admin* membutuhkan *username* dan *password* untuk mengakses aplikasi. Berikut ini merupakan perancangan antarmuka *menu login* pada aplikasi Perancangan Basis Data pada aplikasi *JogjaKuy* berbasis *Android* seperti terlihat pada Gambar 3.12.



The image shows a screenshot of a mobile application window titled "Window Name". The interface is a login screen with the following elements:

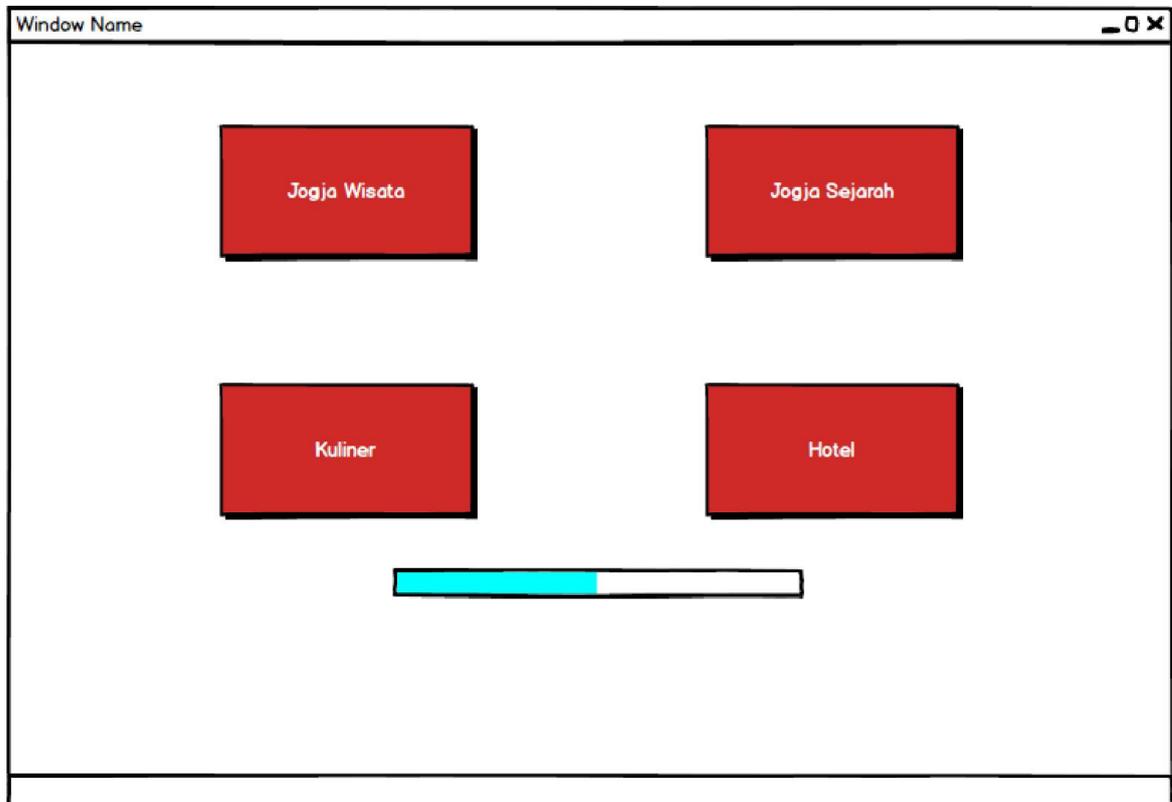
- A label "UserName :" followed by a white rectangular text input field.
- A label "Password :" followed by a white rectangular text input field.
- A red rectangular button with the text "Masuk" in white.
- A horizontal progress bar at the bottom, consisting of a cyan segment on the left and a white segment on the right.

Gambar 3.12 Menu *Login*

### 3.8.2 Antarmuka Menu Utama

Berikut ini adalah rancangan Antarmuka Menu utama pada aplikasi Perancangan Basis Data pada aplikasi *JogjaKuy* berbasis *Android*, terdapat 4

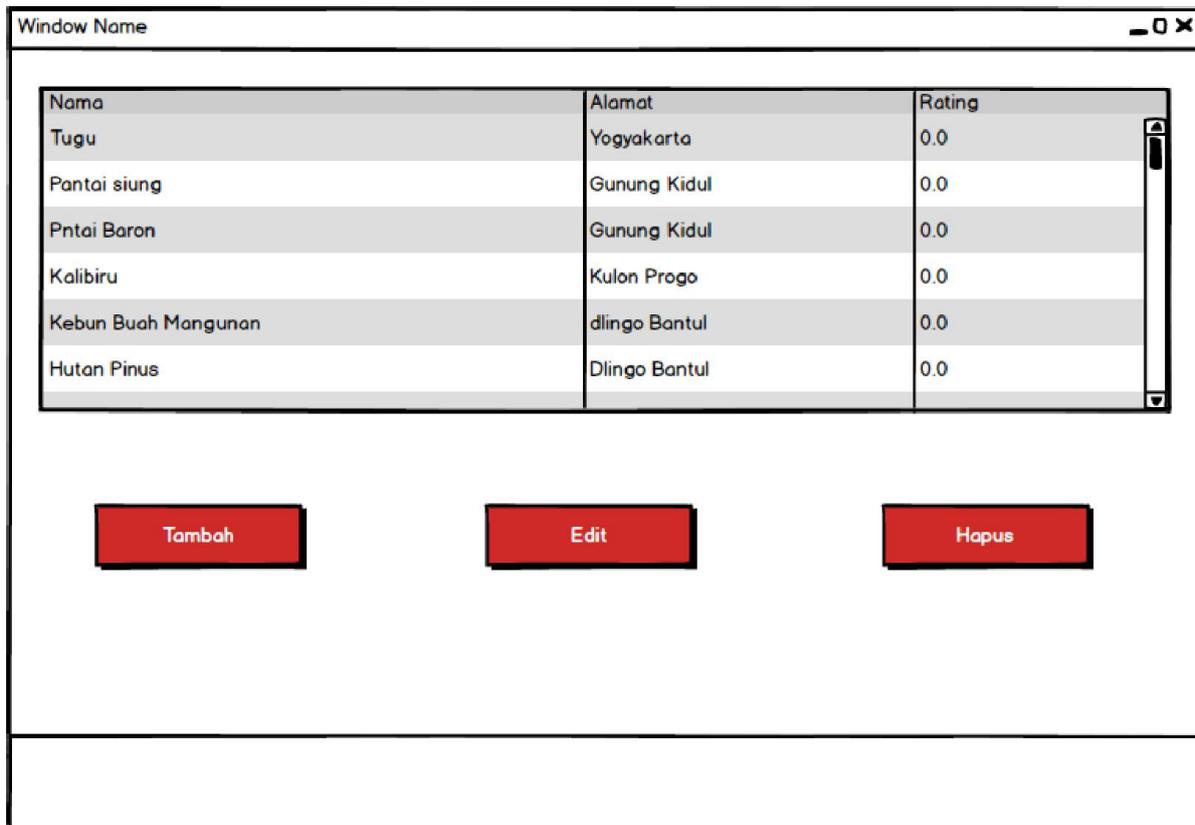
konten pada halaman ini yaitu Jogja wisata, Jogja Sejarah, Kuliner dan Hotel. seperti terlihat pada gambar 3.13.



Gambar 3.13 Menu utama

### 3.8.3 Antarmuka *List* Konten

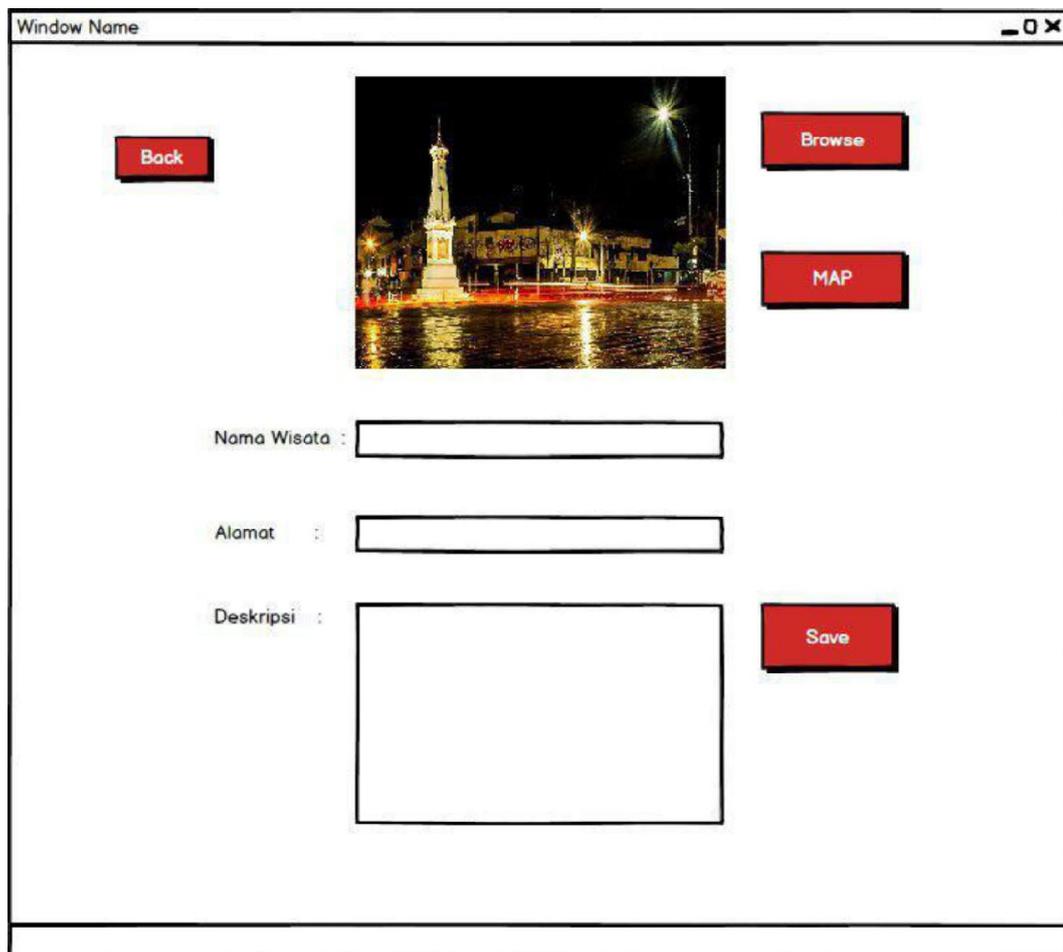
Berikut ini merupakan rancangan antarmuka halaman *List* konten pada aplikasi Perancangan Basis Data pada aplikasi JogjaKuy berbasis *Android*, pada halaman ini *admin* dari aplikasi JogjaKuy dapat menambah, mengedit dan menghapus konten yang ada pada aplikasi. Seperti pada Gambar 3.14.



Gambar 3.14 List konten

### 3.8.4 Antarmuka *Edit* Konten

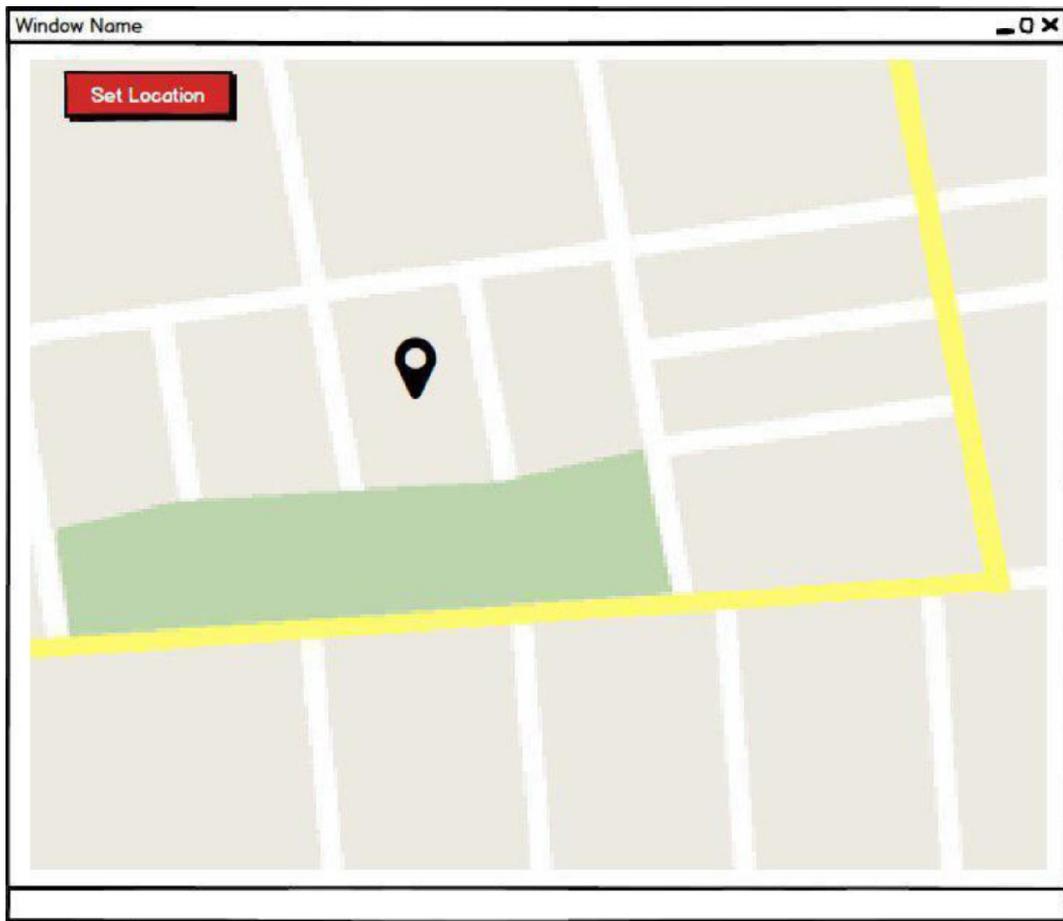
Berikut ini merupakan rancangan antarmuka halaman *Edit* Konten pada aplikasi Perancangan Basis Data pada aplikasi JogjaKuy berbasis *Android*, pada halaman ini *admin* aplikasi JogjaKuy bisa menambahkan konten, menambah gambar, dan menentukan titik koordinat lokasi seperti pada Gambar 3.15.



Gambar 3.15 Menu Edit Konten

### 3.8.5 Antarmuka Peta Lokasi

Berikut ini merupakan perancangan antarmuka Peta Lokasi yang akan muncul jika pengguna memilih *button* "Map" pada halaman edit Konten. Perancangan peta lokasi dapat dilihat pada Gambar 3.16.



Gambar 3.16 Peta Lokasi