

## **BAB V**

### **IMPLEMENTASI DAN PENGUJIAN SISTEM**

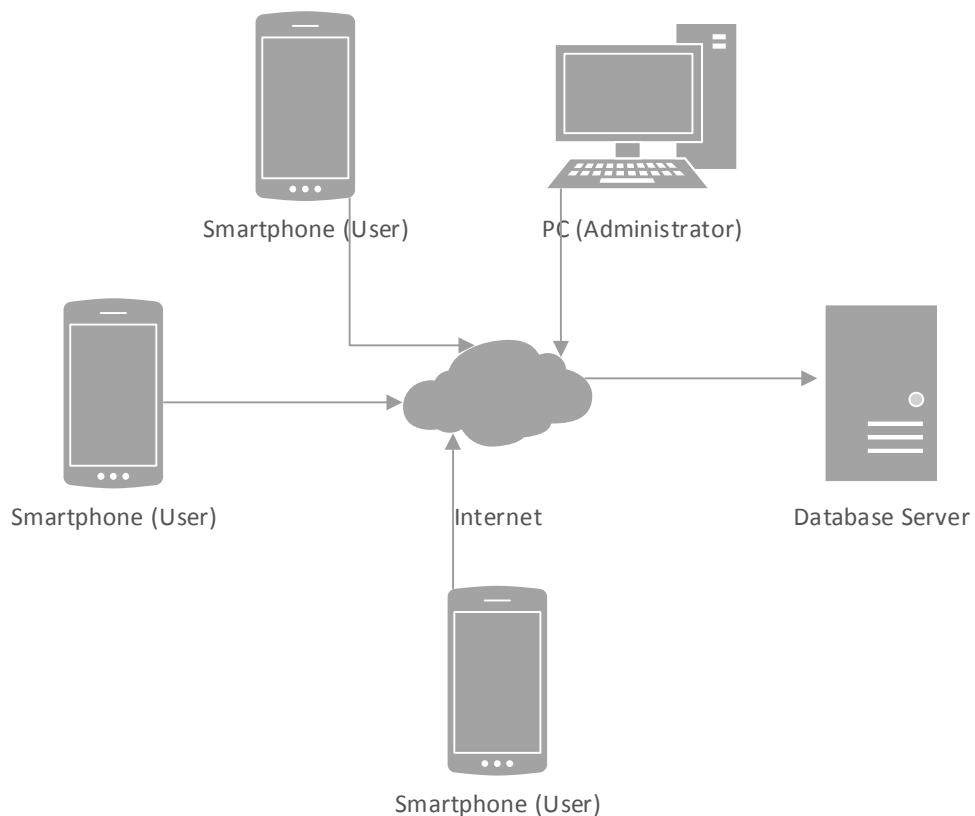
Pada bab implementasi dan pengujian perangkat lunak ini akan dijelaskan mengenai definisi sistem, implementasi sistem, dan hasil pengujian sistem.

#### **5.1 Definisi Perangkat lunak**

Aplikasi TandaTanya memiliki dua bagian yang saling terintegrasi. Bagian pertama yang dipergunakan internal dari TandaTanya yang berbasis web sebagai admin yang memiliki fungsi untuk menambah, merubah maupun menghapus data kategori tempat wisata, tempat wisata, pertanyaan, jawaban, *badge*, dan foto. Bagian kedua dari aplikasi ini yaitu dipergunakan oleh pengguna dalam hal ini pengguna bias menggunakan fitur yang tersedia dan tidak bisa merubah data yang sudah ada kecuali data pribadi pengguna. Pada bagian ini pengguna dapat bermain kuis, *check in*, melihat sejarah pengguna, melihat teman Facebook pengguna yang menggunakan aplikasi TandaTanya, mendapatkan skor dan *badge*.

Perangkat lunak ini berjalan pada perangkat bergerak berupa *Smartphone* yang memiliki system operasi Android yang terhubung dengan koneksi internet, aplikasi ini dibuat menggunakan bahasa pemrograman Java dan *tools* yang digunakan adalah Eclipse Juno, sedangkan pada *web service* menggunakan *framework* Codeigniter dan menggunakan bahasa pemrograman PHP, HTML, dengan *tools* yang digunakan adalah Sublime Text 2.

Arsitektur perangkat lunak berupa *client-server*, di mana semua data disimpan di sebuah *web server* seperti pada gambar 5.1. Administrator dapat melakukan pengelolaan data yang ada di *server* dengan menggunakan PC yang terhubung ke *server* melalui *web service*. Pengguna dapat bermain dan melihat informasi tempat wisata melalui perangkat Android yang terhubung ke *server* melalui *web service* dengan menggunakan jaringan internet yang tersedia.



**Gambar 5.1** Arsitektur Perangkat Lunak TANDATANYA

Tabel 5.1 sampai dengan tabel 5.4 berisi semua file beserta ukuran dan keterangan yang dibuat untuk membangun TandaTanya-Web :

Tabel 5.1 Tabel File Views TandaTanya-Web

No.	Nama File	Ukuran	Keterangan
1	login.php	2 KB	Digunakan sebagai halaman login
2.	home.php	14 KB	Digunakan sebagai halaman beranda sekaligus pengelolaan data pertanyaan
3.	Category_manager.php	11 KB	Digunakan sebagai halaman pengelolaan data kategori tempat wisata
4.	Place_manager.php	17 KB	Digunakan sebagai halaman pengelolaan data tempat wisata
5.	Answer_view.php	12 KB	Digunakan sebagai halaman pengelolaan data pertanyaan
6.	Foto_manager.php	8 KB	Digunakan sebagai halaman pengelolaan data foto
7.	Badge_manager.php	25 KB	Digunakan sebagai halaman pengelolaan data badge.

**Tabel 5.2 Tabel File Controllers TandaTanya-Web**

No.	Nama File	Ukuran	Keterangan
1	tandatanya.php	10 KB	Controller <i>Web Service</i> Aplikasi TANDATANYA
2.	Tandmind.php	15 KB	Controller <i>web</i> TANDATANYA Admin

**Tabel 5.3 Tabel File Models TandaTanya-Web**

No.	Nama File	Ukuran	Keterangan
1	Tandatanya_model.php	12 KB	Model TANDATANYA-Web & Aplikasi

Tabel 5.5 sampai dengan tabel 5.8 berisi semua file beserta ukuran dan keterangannya yang dibuat untuk membangun TandaTanya-Android:

**Tabel 5.4 Tabel File Activity dan File Pendukung Aplikasi TandaTanya-Android**

No.	Nama File	Ukuran	Keterangan
1	AboutUsFragment.java	1 KB	Fragment untuk menampilkan tentang aplikasi TANDATANYA
2.	CallbackBadges.java	1 KB	Kelas <i>interface</i> yang



			digunakan mengelola data balikan dari data <i>badge</i>
3.	CallbackFriends.java	1 KB	Kelas <i>interface</i> yang digunakan mengelola data balikan dari data teman
4.	CallbackHistory.java	1 KB	Kelas <i>interface</i> yang digunakan mengelola data balikan dari data sejarah user
5.	CallbackPlaceProfile.java	1 KB	Kelas <i>interface</i> yang digunakan mengelola data balikan dari data profil tempat wisata
6.	CallBackPlaces.java	1 KB	Kelas

			<i>interface</i> yang digunakan mengelola data balikan dari data tempat wisata
7.	CallbackUser.java	1 KB	Kelas <i>interface</i> yang digunakan mengelola data balikan dari data user
8.	FriendsFragment.java	6 KB	Fragment untuk menampilkan teman Facebook pengguna
9.	HistoryFragment.java	4 KB	Fragment untuk menampilkan sejarah pengguna
10.	HistoryListAdapter.java	3 KB	Kelas pembentuk list sejarah pengguna
11.	FriendListAdapter.java	2 KB	Kelas

			pembentuk list teman Facebook pengguna
12.	LoginActivity.java	7 KB	Activity untuk menampilkan form Login Facebook
13.	MainActivity.java	7 KB	Activity dasar
14.	MapActivity.java	6 KB	Activity untuk menampilkan peta saat bermain
15.	MemoryCache.java	3 KB	Kelas untuk menyimpan data ( <i>cache</i> ) pada <i>platform</i>
16.	NavDrawerItem.java	2 KB	Kelas <i>entity</i> dari list menu item
17.	NavDrawerListAdapter.java	3 KB	Kelas pembentuk list menu
18.	PlaceActivity.java	2 KB	Activity untuk menampilkan info tempat wisata

19.	ProfileFragment.java	4 KB	Fragment untuk menampilkan profil pengguna
20.	QuestionCallback.java	5 KB	Kelas <i>interface</i> yang digunakan mengelola data balikan dari data pertanyaan
21.	QuizActivity.java	5 KB	Activity untuk menampilkan kuis
22.	QuizResultActivity.java	2 KB	Activity untuk menampilkan hasil kuis
23.	TandatanyaRestClient.java	1 KB	Penyedia layanan akses <i>Web Service</i>

**Tabel 5.5 Tabel File Controller Aplikasi TandaTanya-Android**

No.	Nama File	Ukuran	Keterangan
1	BadgeController.java	1 KB	Controller data <i>Badge</i>
2.	CheckPointController.java	1 KB	Controller data

			<i>checkpoint</i>
3.	HistoryController.java	1 KB	Controller data sejarah pengguna
4.	PlaceController.java	1 KB	Controller data tempat wisata
5.	QuizController.java	1 KB	Controller data kuis
6.	UserController.java	2 KB	Controller data pengguna

**Tabel 5.6** Tabel File Model Aplikasi TandaTanya-Android

No.	Nama File	Ukuran	Keterangan
1	BadgeModel.java	3 KB	Model data <i>Badge</i>
2.	CheckpointModel.java	1 KB	Model data <i>check point</i>
3.	HistoryModel.java	3 KB	Model data sejarah pengguna
4.	PlaceModel.java	5 KB	Model data tempat wisata
5.	QuizModel.java	4 KB	Model data kuis
6.	UserModel.java	4 KB	Model data pengguna

**Tabel 5.7** Tabel File Entitiy Aplikasi TandaTanya-Android

No.	Nama File	Ukuran	Keterangan
1	Answer.java	1 KB	Entity data jawaban
2.	Badge.java	2 KB	Entity data <i>Badge</i>
3.	Checkpoint.java	1 KB	Entity data <i>Check</i>

			<i>point</i>
4.	History.java	2 KB	Entity data sejarah
5.	Place.java	2 KB	Entity data tempat wisata
6.	Question.java	1 KB	Entity data pertanyaan
7.	Users.java	1 KB	Entitiy data pengguna

## 5.2 Implementasi Sistem

Referensi kelas dan antarmuka mengacu pada dokumen DPPL-TANDATANYA (Deskripsi Perancangan Perangkat Lunak TANDATANYA) yang disertakan sebagai lampiran

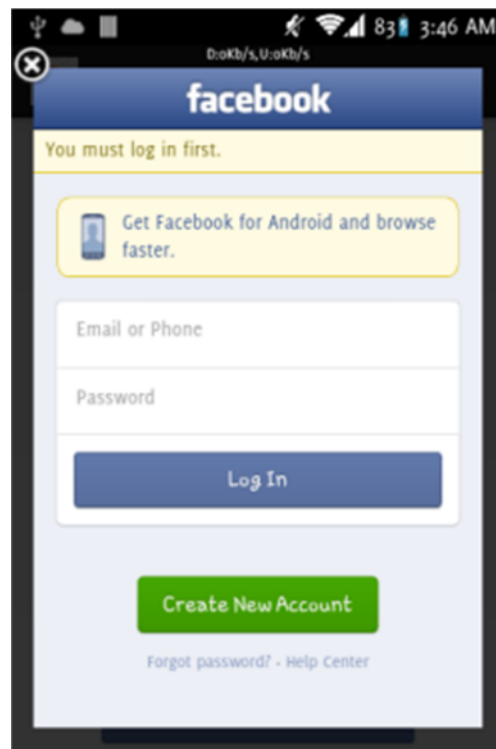
### 5.2.1 Antarmuka TANDATANYA-Android

#### 5.2.1.1 Antarmuka Login



Gambar 5.2 Antarmuka Log In

Antarmuka gambar 5.2 digunakan sebagai halaman awal dari aplikasi TANDATANYA. Ketika aplikasi dijalankan, LoginActivity merupakan antarmuka pertama yang ditampilkan ke user. Tombol 'Login with Facebook' digunakan untuk memanggil antarmuka login akun Facebook. Antarmuka login akun Facebook dapat dilihat pada gambar 5.3.



**Gambar 5.3 Antarmuka Log In Facebook**

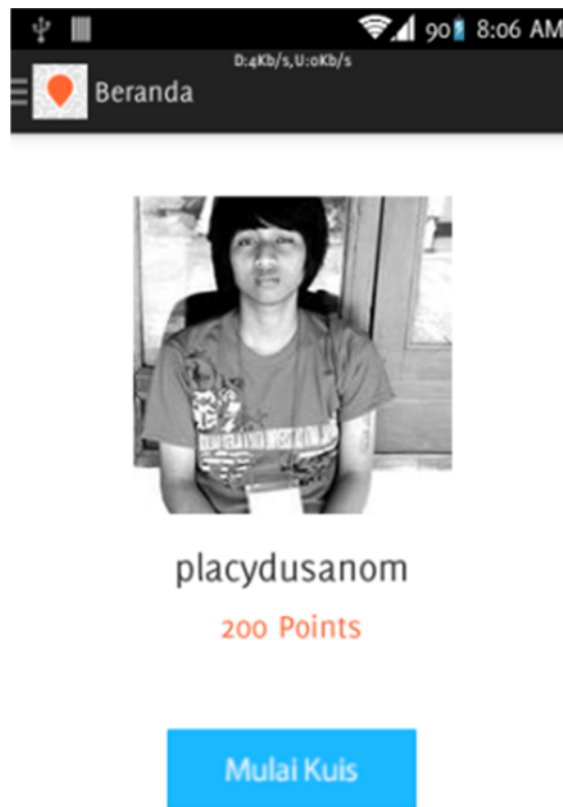
Antarmuka gambar 5.3 digunakan *user* untuk mengisi *username* dan *password* akun Facebook. Setelah *user* memasukkan *username* dan *password* yang valid, system akan mengirimkan data ke *server* Facebook untuk autentifikasi akun yang kemudian *server* Facebook akan mengirimkan data balikan ke aplikasi, jika data autentifikasi valid maka aplikasi akan menampilkan

antarmuka utama aplikasi. Proses ini dapat dilihat pada gambar 5.4.



Gambar 5.4. Proses Login

#### 5.2.1.2 Antarmuka Home

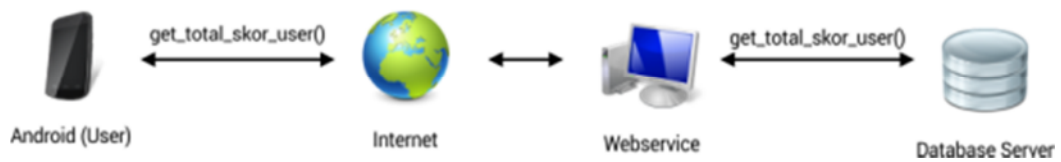


Gambar 5.5 Antarmuka Home

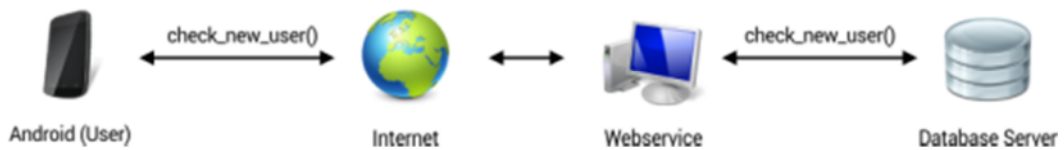
Antarmuka gambar 5.5 digunakan untuk menampilkan antarmuka utama aplikasi TANDATANYA sekaligus menampilkan profil *user*. Ketika antarmuka Home ini ditampilkan, maka aplikasi akan menjalankan dua buah proses yaitu pengambilan data profil dan pengecekan



user baru. Data user yang ditampilkan terdiri dari foto profil, nama, dan total skor dimana foto profil dan nama didapatkan langsung melalui proses autentifikasi ketika login Facebook sedangkan untuk data total skor diambil dari basis data oleh aplikasi. Proses ini dapat dilihat pada gambar 5.6. Pengecekan user baru dilakukan dengan mengirim user id ke basis data lalu server akan mengecek user id tersebut sudah ada dalam basis data atau belum, jika belum maka server otomatis akan mendaftarkan user tersebut. Proses ini dapat dilihat pada gambar 5.7.

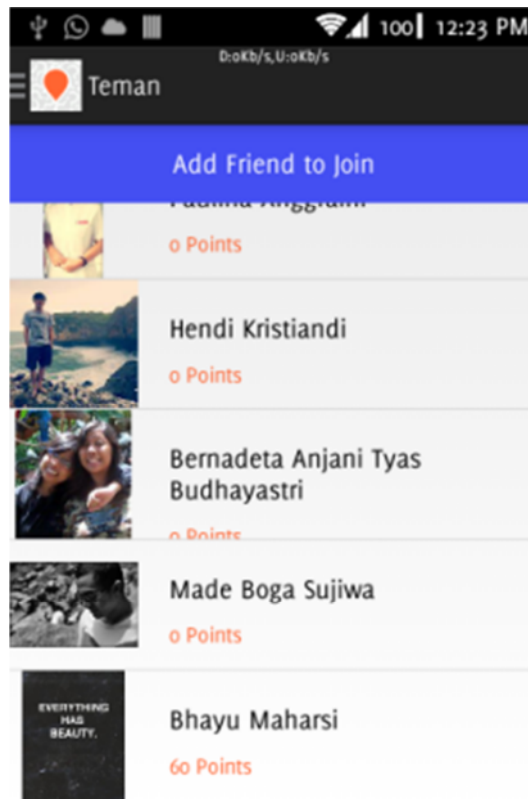


**Gambar 5.6 Proses Mengambil Total Skor**



**Gambar 5.7 Proses Cek User Baru**

### 5.2.1.3 Antarmuka Friends



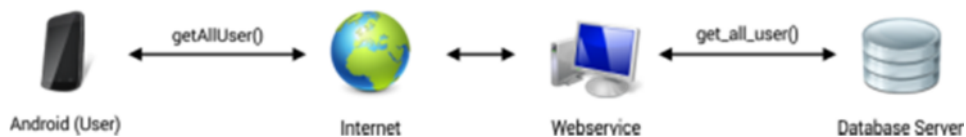
Gambar 5.8 Antarmuka Friends

Antarmuka gambar 5.8 digunakan untuk menampilkan teman Facebook *user* yang juga menggunakan aplikasi TANDATANYA beserta perolehan skor setiap teman berupa list. Data yang ditampilkan diperoleh dari dua buah server yaitu dari Facebook *server* dan *server* TANDATANYA. Pertama aplikasi akan mengambil data teman Facebook yang menggunakan aplikasi TANDATANYA dengan mengirimkan *query* beserta *id user* yang sedang aktif, sehingga Facebook *server* akan memberi data balikan berupa *json* yang kemudian diambil *id user*-nya, kemudian aplikasi mengambil semua *user* yang ada pada basis data dengan memanggil fungsi `getAllUser()`, setelah kedua data didapatkan, aplikasi mencocokkan *id user*, jika ada *id user* yang sama maka info *user* tersebut dimasukkan

kedalam list. Kemudian setelah semua data diperoleh aplikasi akan membentuk list teman yang menampilkan foto profil, nama, dan perolehan skor setiap teman. Proses ini dapat dilihat pada gambar 5.9.



**Gambar 5.9 Proses Mengambil Data Teman Facebook**



**Gambar 5.10 Proses Mengambil Data Semua User**

Pada gambar 5.10 terdapat proses pengambilan data semua user dimana aplikasi memanggil fungsi `get_all_user()`, fungsi ini akan memanggil fungsi `get_all_user()` pada `web service controller tandatanya.php` yang akan memberikan data semua user info. Fungsi `get_all_user()` dapat dilihat pada gambar 5.11.

```

public function get_all_user(){
    $data['users'] = $this->tandatanya_model->get_all_user();
    echo json_encode($data);
}
  
```

**Gambar 5.11 Fungsi `get_all_user()` tandatanya.php**

Fungsi `get_all_user()` pada gambar 5.11 akan memanggil fungsi pada `tandatanya_model.php` `get_all_user()` guna mengambil data pada basis data. Fungsi ini dapat dilihat pada gambar 5.12.

```

public function get_all_user(){
    $this->db->select('*');
    $this->db->from('users');
}
  
```

```
return $this->db->get()->result_array();  
}
```

**Gambar 5.12 Fungsi get\_all\_user() tandatanya-model.php**

Fungsi pada gambar 5.12 mengambil semua data user yang ada pada basis data dan membentuk data keluaran bertipe array sehingga dapat dilihat data json pada gambar 5.13.

```
{"users": [{"USER_ID": "100007370382223", "USERNAME": "bhayu.maharsi", "TOTAL_POINT": "60"}, {"USER_ID": "1504697984", "USERNAME": "placydusanom", "TOTAL_POINT": "200"}]}
```

**Gambar 5.13 JSON Output Fungsi get\_all\_user()**

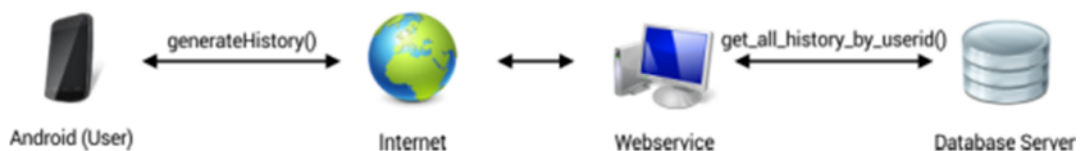
#### 5.2.1.4 Antarmuka History



**Gambar 5.14 Antarmuka History**

Antarmuka gambar 5.14 digunakan untuk menampilkan sejarah user. Sejarah user menampilkan data tempat wisata dan perolehan skor user pada tempat wisata tersebut. Sehingga ketika antarmuka gambar 5.14 diakses oleh user maka aplikasi akan mengambil data sejarah

user dari basis data dengan mengirimkan id user aktif, maka server akan memberikan data balikan sejarah user berupa json yang berisi alamat url foto profil tempat wisata dan skor yang diperoleh user yang kemudian data-data tersebut ditampilkan oleh aplikasi ke dalam sebuah list. Ketika list sejarah user sudah terbentuk, user juga dapat menekan list item untuk menampilkan profil tempat wisata yang akan dibahas pada antarmuka tempat wisata. Proses ini dapat dilihat pada gambar 5.15.



**Gambar 5.15 Proses Mengambil Data Sejarah User**

Pada gambar 5.15 aplikasi memanggil fungsi `get_all_history_by_userid()` yang fungsi tersebut memanggil fungsi pada controller `web service tandatanya.php` `get_all_history_by_userid()` yang dapat dilihat pada gambar 5.16.

```

public function get_all_history_by_userid(){
    $userid = $this->input->get('userid');
    $data['history'] = $this->tandatanya_model->get_all_history_and_place_by_userid($userid);
    echo json_encode($data['history']);
}
  
```

**Gambar 5.16 Fungsi `get_all_history_by_userid()` pada `tandatanya.php`**

Pada gambar 5.16 fungsi `get_all_history_byuserid()` mula-mula mengambil `userid` yang dilemparkan dari aplikasi menggunakan fungsi `$this->input->get('userid')`, kemudian fungsi ini akan mengambil data

sejarah user pada fungsi `$this->tandatanya_model->get_all_history_and_place_by_userid($userid)` yang fungsi tersebut dapat dilihat pada gambar 5.17.

```
public function get_all_history_and_place_by_userid($userid){
    $sql = "SELECT * FROM history JOIN places ON
history.PLACE_ID = places.PLACE_ID JOIN photos ON
places.PLACE_ID = photos.PLACE_ID WHERE history.USER_ID
=".$userid." GROUP BY places.PLACE_ID" ;
    $query = $this->db->query($sql);
    return $query->result_array();
}
```

**Gambar 5.17 Fungsi `get_all_history_and_place_by_userid($userid)` pada `tandatanya-model.php`**

Pada gambar 5.17 fungsi `get_all_history_and_place_by_userid($userid)` akan mengambil semua data 'history' dan 'place' dari basis data berdasarkan 'userid' dan membentuk hasil datanya bertipe *array*, sehingga data akan dibalikkan dalam bentuk json yang dapat dilihat pada gambar 5.18.

```
[{"HISTORY_ID":"6","PLACE_ID":"2","USER_ID":"1504697984","POINT":"30","DATE_PLAY":null,"CAT_PLACE_ID":"3","PLACE_NAME":"Monjali","PLACE_ADDRESS":"Jl. Lingkar Utara, Yogyakarta 55581, Indonesia","PLACE_DESCRIPTION":"Museum Monumen Yogya Kembali, adalah sebuah museum sejarah perjuangan kemerdekaan Republik Indonesia yang ada di kota Yogyakarta dan dikelola oleh Departemen Kebudayaan dan Pariwisata. Museum yang berada di bagian utara kota ini banyak dikunjungi oleh","PLACE_LAT":"-7.749945","PLACE_LON":"110.369593","PHOTO_ID":"7","PHOTO_URL":"http://localhost/tandatanya/img_place/30c0a12920cb86badde5cf1e05daa441.jpg"},{"HISTORY_ID":"12","PLACE_ID":"10","USER_ID":"1504697984","POINT":"20","DATE_PLAY":null,"CAT_PLACE_ID":"3","PLACE_NAME":"Museum Affandi","PLACE_ADDRESS":"Jl. Laksda Adisucipto 167, Yogyakarta 55281, Indonesia","PLACE_DESCRIPTION":"Museum affandi adalah seluruh bagian dari kehidupan Affandi sebagai maestro seni lukis. Di wilayah tepi sungai Gajah Wong itu, Affandi hidup, berkarya, mentransformasikan ilmunya dan bersemayam di rumah abadinya.","PLACE_LAT":"-7.783154","PLACE_LON":"110.396154","PHOTO_ID":"10","PHOTO_URL":"http://localhost/tandatanya/img_place/bb8270a6eb6cc776438a461e58cb923d.jpg"},{"HISTORY_ID":"11","PLACE_ID":"11","USER_ID":"1504697984","POINT":"10","DATE_PLAY":null,"CAT_PLACE_ID":"4","PLACE_NAME":"Tugu","PLACE_ADDRESS":"","PLACE_DESCRIPTION":"Tu
```

```

gu Jogja merupakan landmark Kota Yogyakarta yang paling
terkenal. Monumen ini berada tepat di tengah perempatan Jalan
Pangeran Mangkubumi, Jalan Jendral Soedirman, Jalan A.M Sangaji
dan Jalan Diponegoro. Tugu Jogja yang berusia hampir 3 abad
memili", "PLACE_LAT": "-
7.782931", "PLACE_LON": "110.366998", "PHOTO_ID": "13", "PHOTO_URL":
"http://localhost/tandatanya/img_place/371e3d0da8c947ae6f8
435402e6f2b3b.jpg"}]

```

**Gambar 5.18 JSON Output Fungsi  
get\_all\_history\_and\_place\_by\_userid()**

Setelah data JSON didapatkan dari server seperti gambar 5.18 maka aplikasi akan mengolah data tersebut untuk digunakan dalam membentuk list sejarah user yang fungsinya dapat dilihat pada gambar 5.19.

```

try {
JSONArray jaHistory = new JSONArray(arg0);
for (int i = 0; i < jaHistory.length(); i++) {
History history = new History();
JSONObject joHistory = jaHistory
.getJSONObject(i);
history.setHistory_id(Integer.valueOf(joHistory.getString("HIS
TORY_ID")));
history.setPlace_id(Integer.valueOf(joHistory.getString("PLACE
_ID")));
history.setPlace_name(joHistory.getString("PLACE_NAME"));
history.setPlace_photo_url((joHistory.getString("PHOTO_URL").r
eplace("\\/", "").replace("http://localhost/",
"http://192.168.234.1/"));
history.setPoint(joHistory.getString("POINT"));
arlHistory.add(history);
}
} catch (JSONException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
Log.d("DATA HISTORY", arg0);
historyFragment.onHistoryDataReceived(arlHistory);

```

**Gambar 5.19 Fungsi OnSucces() pada  
get\_all\_history\_and\_place\_by\_userid()**

#### 5.2.1.5 Antarmuka About Us



**Gambar 5.20 Antarmuka About Us**

Antarmuka gambar 5.20 digunakan untuk menampilkan tentang aplikasi TANDATANYA. Antarmuka gambar 5.55 tidak memerlukan akses data ke server karena data yang ditampilkan tersimpan pada aplikasi.



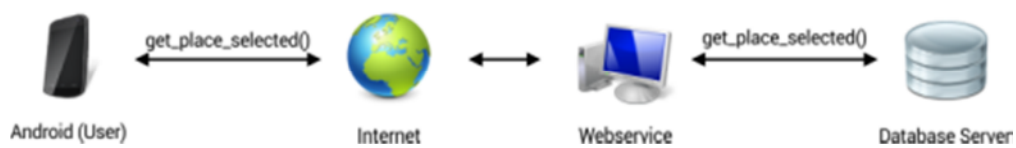
#### 5.2.1.6 Antarmuka Play



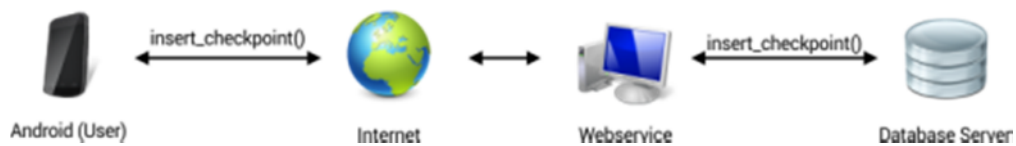
Gambar 5.21 Antarmuka Play

Antarmuka gambar 5.21 digunakan untuk menampilkan peta yang merupakan tampilan awal ketika user memilih untuk bermain kuis. Antarmuka gambar 5.21 menampilkan sebuah peta bersama marker-marker tempat wisata yang dimana posisi tengah peta diatur pada marker tempat wisata yang harus dituju oleh user. Sehingga pada proses ini aplikasi akan meminta data tempat wisata pada server, sehingga server akan mengambil semua data tempat wisata yang belum pernah dikunjungi oleh user (tidak ada pada sejarah user), kemudian server juga akan memilih salah satu tempat wisata sebagai tempat wisata yang harus dituju oleh user secara acak, setelah data siap maka server akan memberikan data balikan kepada aplikasi berupa data json. Proses ini dapat dilihat pada gambar 5.22. Setelah salah satu tempat

wisata terpilih sebagai tempat wisata tujuan *user* maka aplikasi akan menyimpan id tempat wisata dan id *user* aktif ke dalam basis data sebagai data *checkpoint*. Proses ini dapat dilihat pada gambar 5.23. Kemudian antarmuka gambar 5.21 juga memiliki sebuah tombol 'Check In' yang digunakan *user* untuk mengunjungi tempat wisata dan kemudian kuis akan dimulai.



**Gambar 5.22 Proses Mengambil Data Tempat Wisata Terpilih**



**Gambar 5.23 Proses Mengisi Data *Checkpoint***

Pada gambar 5.22 terdapat fungsi `get_place_selected()` pada aplikasi yang dimana akan memanggil fungsi `initilize_play_game()` pada controller tandatanya.php *web service*, fungsi ini dapat dilihat pada gambar 5.24.

```

public function initilize_play_game(){
    $userid = $this->input->get('userid');
    $places_are_valid['place'] = $this->tandatanya_model-
>get_all_places_where_not_in_history_by_user_id($userid);
    $place_from_checkpoint = $this->tandatanya_model-
>get_check_point_by_user_id($userid);

    if(empty($places_are_valid['place'])){
        echo "empty";
    }elseif($place_from_checkpoint->num_rows()>0){
        foreach ($place_from_checkpoint->result_array() as $row)
    {
        $placeid_selected = $row['PLACE_ID'];
    }
    $place_select['select'] = $this->tandatanya_model-
>get_json_place_by_id($placeid_selected);

```

```

        $places_are_valid['select'] = $place_select['select'][0];
    }else{
        $count = count($places_are_valid['place']);
        $rand = rand(0,$count-1);
        $places_are_valid['select'] =
    $places_are_valid['place'][$rand];
    }

    echo json_encode($places_are_valid);
}

```

**Gambar 5.24 Fungsi `intialize_play_game()` pada controller `tandatanya.php`**

Pada gambar 5.24 mula-mula fungsi akan mengambil data `userid` parameter yang dilempar dari aplikasi menggunakan fungsi `$this->input->get('userid');` kemudian aplikasi mengambil semua data tempat wisata yang *valid* atau data tempat wisata yang tidak ada pada data `'history'` dari basis data menggunakan fungsi pada gambar 5.25. Setelah mendapatkan data tempat wisata yang tidak ada pada data `'history'` maka akan dipanggil fungsi untuk mengambil data tempat wisata yang ada pada table `'checkpoint'` menggunakan fungsi pada gambar 5.26. Setelah itu dilakukan pengecekan apakah data tempat wisata *valid* ada atau kosong, jika kosong maka *web service* akan melemparkan data balikan `'empty'`, jika data tidak kosong maka akan dilakukan pengecekan kembali data tempat wisata yang didapat dari table `cekpoint` ada atau kosong, jika kosong maka data tempat wisata yang digunakan adalah data dari data tempat wisata *valid*, sehingga data tempat wisata *valid* tersebut akan dipilih salah satu sebagai tempat wisata yang harus dituju oleh *user* secara acak, begitu juga jika data *checkpoint* tidak kosong. Setelah proses selesai maka *web service* akan melempar data balikan berupa json seperti pada gambar 5.27.

```

public function get_all_history_and_place_by_userid($userid){
    $sql = "SELECT * FROM history JOIN places ON
history.PLACE_ID = places.PLACE_ID JOIN photos ON
places.PLACE_ID = photos.PLACE_ID WHERE history.USER_ID
=".$userid." GROUP BY places.PLACE_ID" ;
    $query = $this->db->query($sql);
    return $query->result_array();
}

```

**Gambar 5.25 Fungsi `get_all_history_and_place_by_userid($userid)` pada tandatanya-model.php**

```

public function get_check_point_by_user_id($userid){
    $this->db->select('*');
    $this->db->from('checkpoint');
    $this->db->where('USER_ID',$userid);
    return $this->db->get();
}

```

**Gambar 5.26 Fungsi `get_check_point_by_userid($userid)` pada tandatanya-model.php**

```

{"place":[{"PLACE_ID":"1","CAT_PLACE_ID":"1","PLACE_NAME":"Candi
Prambanan","PLACE_ADDRESS":"Jalan Raya Jogja - Solo Km 16, Prambanan,
Sleman, Yogyakarta 55571, Indonesia","PLACE_DESCRIPTION":"Candi
Prambanan adalah mahakarya kebudayaan Hindu dari abad ke-10.
Bangunannya yang langsing dan menjulang setinggi 47 meter membuat
kecantikan arsitekturnya tak tertandingi.","PLACE_LAT":-
7.75205","PLACE_LON":"110.491966"}],"select":{"PLACE_ID":"1","CAT_PLA
CE_ID":"1","PLACE_NAME":"Candi Prambanan","PLACE_ADDRESS":"Jalan Raya
Jogja - Solo Km 16, Prambanan, Sleman, Yogyakarta 55571,
Indonesia","PLACE_DESCRIPTION":"Candi Prambanan adalah mahakarya
kebudayaan Hindu dari abad ke-10. Bangunannya yang langsing dan
menjulang setinggi 47 meter membuat kecantikan arsitekturnya tak
tertandingi.","PLACE_LAT":-7.75205","PLACE_LON":"110.491966"}}

```

**Gambar 5.27 JSON Output fungsi `get_all_history_and_place_by_userid()` pada tandatanya.php**

Setelah data JSON didapatkan dari server seperti gambar 5.27 maka aplikasi akan mengolah data tersebut untuk digunakan dalam menampilkan tempat-tempat wisata pada peta yang fungsinya dapat dilihat pada gambar 5.28.

```

Log.d("DATA JSON", arg0);
try {

```

```

        JSONObject joRoot = new JSONObject(arg0);
        JSONObject joPlaceSelected = joRoot.getJSONObject("select");
        placeSelected.setPlace_id(Integer.valueOf(joPlaceSelected.getString("PLACE_ID")));

        placeSelected.setPlace_name(joPlaceSelected.getString("PLACE_NAME"));

        placeSelected.setPlace_description(joPlaceSelected.getString("PLACE_ADDRESS"));

        placeSelected.setPlace_lat(Double.valueOf(joPlaceSelected.getString("PLACE_LAT")));

        placeSelected.setPlace_lon(Double.valueOf(joPlaceSelected.getString("PLACE_LON")));

        JSONArray jaPlaces = joRoot.getJSONArray("place");
        for (int i = 0; i < jaPlaces.length(); i++) {
            Place place = new Place();
            JSONObject joPlace = jaPlaces.getJSONObject(i);
            place.setPlace_name(joPlace
                .getString("PLACE_NAME"));
            place.setPlace_description(joPlace
                .getString("PLACE_DESCRIPTION"));
            place.setPlace_address(joPlace
                .getString("PLACE_ADDRESS"));

            place.setPlace_lat(Double.valueOf(joPlace
                .getString("PLACE_LAT")));
            place.setPlace_lon(Double.valueOf(joPlace
                .getString("PLACE_LON")));

            arlPlaces.add(place);
            Log.d("PLACE ALL", place.getPlace_name());
        }
        Log.d("PLACE SELECT", placeSelected.getPlace_name());

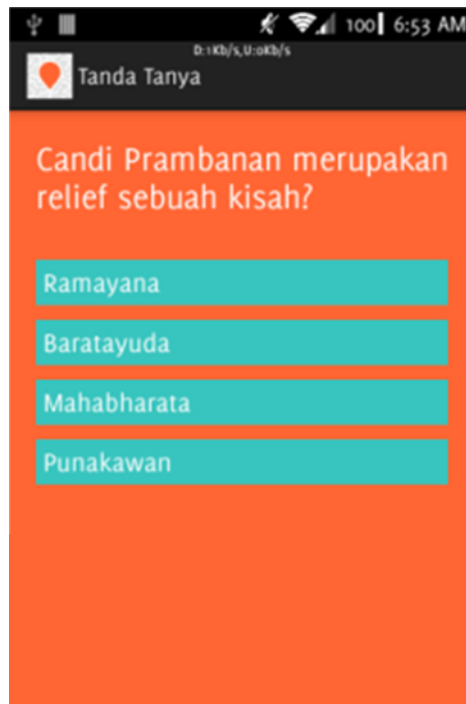
        mapAct.OnPlaceSelectedReceive(placeSelected, arlPlaces);

    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

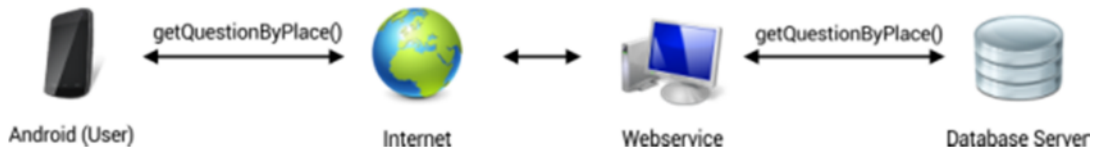
**Gambar 5.28 Fungsi OnSucces() pada get\_place\_selected()**

#### 5.2.1.7 Antarmuka Quiz

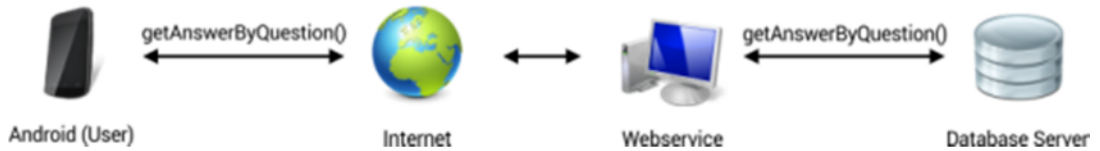


Gambar 5.29 Antarmuka Quiz

Antarmuka gambar 5.29 digunakan untuk menampilkan kuis. Aplikasi menampilkan antarmuka kuis yang terdiri dari pertanyaan dengan jawaban-jawaban dari pertanyaan. Ketika antarmuka gambar 5.29 diakses oleh *user* maka mula-mula aplikasi akan mengambil seluruh data pertanyaan dan jawaban-jawaban sesuai tempat wisata yang dikunjungi oleh *user*, kemudian aplikasi menampilkannya satu persatu ke layar. Ketika pertanyaan dan pilihan jawaban tertampil, *user* dapat memilih jawaban dengan cara menekan jawaban yang dianggap benar yang kemudian aplikasi akan mengecek dan menambahkan 10 poin setiap jawaban benar yang dipilih oleh *user*. Proses ini dapat dilihat pada gambar 5.30 dan gambar 5.31.



Gambar 5.30 Proses Mengambil Data Pertanyaan



Gambar 5.31 Proses Mengambil Data Jawaban

Pada gambar 5.30 terdapat fungsi `getQuestionByPlace()` pada aplikasi yang dimana fungsi tersebut akan memanggil fungsi `get_all_question_by_placeid()` pada controller `tandatanya.php` di *web service* yang fungsi tersebut dapat dilihat pada gambar 5.32.

```

public function get_all_question_by_placeid(){
    $placeid = $this->input->get('placeid');
    echo json_encode($this->tandatanya_model->get_question_by_id($placeid));
}
  
```

Gambar 5.32 Fungsi `get_all_question_by_placeid()` pada `tandatanya.php`

Fungsi pada gambar 5.32 mula-mula akan mengambil data `'placeid'` parameter yang dilempar dari aplikasi menggunakan `$this->input->get('placeid');` kemudian data `'placeid'` tersebut digunakan untuk mengambil semua data pertanyaan berdasarkan id tempat wisata (`placeid`) dari basis data menggunakan fungsi `$this->tandatanya_model->get_question_by_id($placeid)` seperti tampak pada gambar 5.33 yang kemudian data pertanyaan yang didapat

akan dibalikkan pada aplikasi berupa data json yang tampak pada gambar 5.34.

```
public function get_question_by_id($idIcon){
    $this->db->select('*');
    $this->db->from('questions');
    $this->db->where('PLACE_ID',$idIcon);
    $query = $this->db->get();
    return $query->result_array();
}
```

**Gambar 5.33 Fungsi `get_all_question_by_placeid()` pada `tandatanya.php`**

```
[{"QUESTION_ID":"0","PLACE_ID":"1","QUESTION":"Candi Prambanan merupakan relief sebuah kisah?"}, {"QUESTION_ID":"1","PLACE_ID":"1","QUESTION":"Terletak di manakah candi prambanan?"}, {"QUESTION_ID":"19","PLACE_ID":"1","QUESTION":"Pada abad ke berapa Candi Prambanan dibangun?"}, {"QUESTION_ID":"20","PLACE_ID":"1","QUESTION":"Siapakah nama laki-laki yang menjadi tokoh di dalam balik cerita Candi Prambanan?"}, {"QUESTION_ID":"21","PLACE_ID":"1","QUESTION":"Siapakah nama seorang perempuan yang menjadi tokoh di balik kisah Candi Prambanan?"}, {"QUESTION_ID":"22","PLACE_ID":"1","QUESTION":"Candi Prambanan memiliki 3 candi utama, yaitu?"}, {"QUESTION_ID":"23","PLACE_ID":"1","QUESTION":"Berapa ruangan yang terdapat di dalam Candi Siwa?"}]
```

**Gambar 5.34 JSON Output fungsi `get_question_by_id($idIcon)`**

Setelah data pertanyaan didapatkan dari *web service* maka aplikasi akan mengolah data balikan json tersebut menggunakan fungsi seperti pada gambar 5.35.

```
try {
    JSONArray jaQuestion = new JSONArray(arg0);
    for (int i = 0; i < jaQuestion.length(); i++) {
        Question question = new Question();
        JSONObject joQuestion = jaQuestion
            .getJSONObject(i);
        question.setQuestion_id(joQuestion
            .getString("QUESTION_ID"));
        question.setPlace_id(joQuestion
            .getString("PLACE_ID"));
        question.setQuestion(joQuestion
            .getString("QUESTION"));
        arlQuestion.add(question);
    }
} catch (JSONException e) {
```



```

// TODO Auto-generated catch block
e.printStackTrace();
}

Log.d("DATA QUESTION", arg0);
quizAct.OnQuestionDataReceived(arlQuestion);

```

**Gambar 5.35 Fungsi OnSucces() pada get\_all\_question\_by\_placeid()**

Kemudian aplikasi akan memanggil fungsi `getAnswerByQuestion()` yang dimana fungsi ini akan memanggil fungsi `get_all_answer_by_questionid()` pada controller `tandatanya.php` di *web service* dimana fungsi ini dapat dilihat pada gambar 5.36.

```

public function get_all_answer_by_questionid(){
    $questionid = $this->input->get('questionid');
    echo json_encode($this->tandatanya_model-
>get_answer_by_question_id($questionid));
}

```

**Gambar 5.36 Fungsi get\_all\_answer\_by\_questionid() pada tandatanya.php**

Fungsi pada gambar 5.36 mula-mula akan mengambil data parameter id pertanyaan yang dilempar dari aplikasi menggunakan fungsi `$this->input->get('questionid');` yang digunakan untuk mengambil data jawaban berdasarkan id pertanyaan tersebut menggunakan fungsi `$this->tandatanya_model->get_answer_by_question_id($questionid)`, fungsi ini lalu akan merujuk pada fungsi `get_answer_by_question_id($questionid)` pada model `tandatanya_model.php` yang fungsinya dapat dilihat pada gambar 5.37.

```

public function get_answer_by_question_id($id){
    $this->db->select('*');
    $this->db->from('answers');
    $this->db->where('QUESTION_ID',$id);
    $query = $this->db->get();
}

```

```

        return $query->result_array();
    }

```

**Gambar 5.37 Fungsi `get_answer_by_questionid($id)` pada `tandatanya_model.php`**

Fungsi pada gambar 5.37 akan mengambil semua data jawaban berdasarkan id pertanyaan dan membentuk data dalam tipe `array`. Setelah data jawaban didapatkan maka fungsi `get_all_answer_by_questionid()` selanjutnya akan membalikkan data kepada aplikasi berupa data json yang dapat dilihat pada gambar 5.38.

```

[{"ANSWER_ID":"73","QUESTION_ID":"0","ANSWER":"Ramayana","CORRECT":"true"}, {"ANSWER_ID":"74","QUESTION_ID":"0","ANSWER":"Baratayuda","CORRECT":"false"}, {"ANSWER_ID":"75","QUESTION_ID":"0","ANSWER":"Mahabharata","CORRECT":"false"}, {"ANSWER_ID":"76","QUESTION_ID":"0","ANSWER":"Punakawan","CORRECT":"false"}]

```

**Gambar 5.38 JSON Output fungsi `get_all_answer_by_questionid()`**

Setelah mendapat data jawaban balikan dari *web service* maka aplikasi akan mengolah data tersebut menggunakan fungsi seperti pada gambar 5.39 yang dimana data tersebut akan ditampilkan.

```

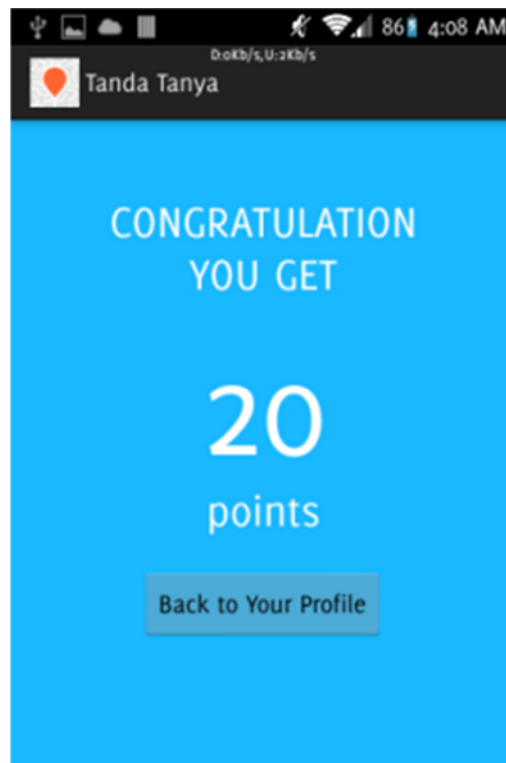
try {
    JSONArray jaAnswer = new JSONArray(arg1);
    for (int i = 0; i < jaAnswer.length(); i++) {
        Answer answer = new Answer();
        JSONObject joAnswer = jaAnswer.getJSONObject(i);
        answer.setQuestion_id(joAnswer
            .getString("QUESTION_ID"));
        answer.setAnswer(joAnswer.getString("ANSWER"));
        answer.setCorrect(joAnswer.getString("CORRECT"));
        arlAnswer.add(answer);
    }
} catch (Exception e) {
    // TODO: handle exception
}

Log.d("GET ANSWER STAT", arg1);
quizAct.OnAnswerDataReceived(arlAnswer);

```

**Gambar 5.39 Fungsi `OnSucces()` pada `get_all_answer_by_questionid()`**

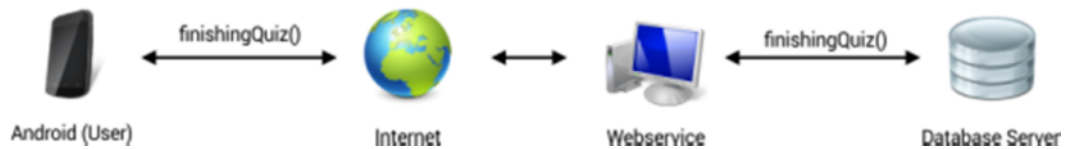
#### 5.2.1.8 Antarmuka Quiz Result



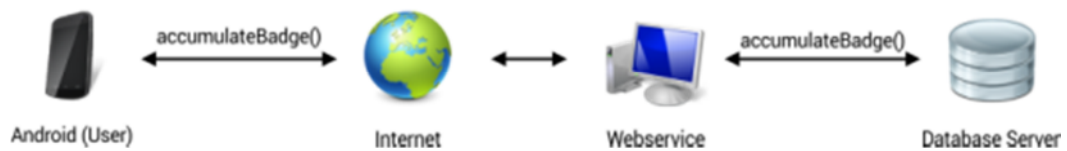
Gambar 5.40 Antarmuka Quiz

Antarmuka gambar 5.40 digunakan untuk menampilkan hasil perolehan skor setelah kuis selesai. Skor yang didapat merupakan hasil dari akumulasi tiap poin jawaban benar yang didapatkan pada antarmuka gambar sebelumnya. Ketika antarmuka gambar 5.40 ini muncul maka aplikasi akan menyimpan perolehan skor ke dalam basis data. Aplikasi akan memanggil fungsi `finising_quiz()` untuk menyimpan perolehan skor berdasarkan id *user* dan id tempat wisata. Selain itu aplikasi juga akan mengecek apakah skor yang diperoleh *user* memenuhi syarat untuk mendapatkan sebuah *badge* dengan fungsi `accumulateBadge()`, jika perolehan skor

user memenuhi syarat untuk mendapatkan *badge* maka aplikasi akan menampilkan peringatan bahwa user mendapatkan *badge* dan menyimpan *badge* id dan user id ke dalam basis data. Proses ini dapat dilihat pada gambar 5.41 dan gambar 5.42.



Gambar 5.41 Proses Menyimpan Skor User



Gambar 5.42 Proses Akumulasi Badge

Pada gambar 5.41 terdapat fungsi `finishingQuiz()` yang digunakan aplikasi untuk menyimpan perolehan skor berdasarkan id user dan id tempat wisata ke dalam table 'history'. Fungsi ini `finishingQuiz()` akan memanggil fungsi `finishing_quiz()` pada controller `tandatanya.php` di *web service* yang fungsi tersebut dapat dilihat pada gambar 5.43

```

public function finishing_quiz(){
    $userid = $this->input->get('userid');
    $place_skor = $this->input->get('placeskor');
    $placeid = $this->input->get('placeid');

    $this->tandatanya_model->delete_check_point_by_user_id($userid);

    $history_data = array(
        "PLACE_ID" => $placeid,
        "USER_ID" => $userid,
        "POINT" => $place_skor
    );

    $this->tandatanya_model->insert_history($history_data);
}
  
```

```

        $skor = $this->tandatanya_model-
>get_total_skor_by_user_id($userid);
        foreach ($skor->result_array() as $row) {
            $theskor = $row['TOTAL_POINT'];
        }

        $total_skor = $theskor + $place_skor;

        $update_user_skor_data = array(
            "TOTAL_POINT" => $total_skor
        );

        $this->tandatanya_model->update_skor_user($userid,
        $update_user_skor_data);
    }

```

**Gambar 5.43 Fungsi finishing\_quiz() pada tandatanya.php**

Fungsi pada gambar 5.43 mula-mula akan memanggil data parameter id user, skor user, dan id tempat wisata, kemudian fungsi ini akan menghapus data pada table 'checkpoint' berdasarkan id user menggunakan fungsi seperti pada gambar 5.44, kemudian fungsi akan memanggil fungsi insert\_history() pada model tandatanya\_model.php seperti pada gambar 5.45, kemudian fungsi akan menambahkan skor yang didapatkan dari aplikasi dengan skor yang sudah didapatkan user sebelumnya dan men-update skor data pada basis data menggunakan fungsi seperti pada gambar 5.46.

```

public function delete_check_point_by_user_id($userid){
    $this->db->where('USER_ID', $userid);
    $this->db->delete('checkpoint');
}

```

**Gambar 5.44 Fungsi delete\_check\_point\_by\_user\_id(\$userid) pada tandatanya\_model.php**

```

public function insert_history($history_data){
    $this->db->insert('history',$history_data);
}

```

Gambar 5.45 Fungsi `insert_history($history_data)` pada `tandatanya_model.php`

```
public function update_skor_user($userid, $value){
    $this->db->where('USER_ID', $userid);
    $this->db->update('users', $value);
}
```

Gambar 5.46 Fungsi `update_skor_user($userid, $value)` pada `tandatanya_model.php`

Kemudian pada gambar 5.42 terdapat fungsi `accumulateBadge()` pada aplikasi yang akan memanggil fungsi `accumulateBadge()` pada controller `tandatanya.php` di *web service* yang dapat dilihat pada gambar 5.47.

```
public function accumulateBadges(){
    $userid = $this->input->get('userid');
    $msg = "";
    $badge_data['user'] = $this->tandatanya_model->check_new_user($userid);

    $badge_data['badges'] = $this->tandatanya_model->get_all_badges_not_in_badge_user_by_user_id($userid);
    if(count($badge_data['badges']) > 0 ){
        for ($i=0; $i < count($badge_data['badges']); $i++) {
            if($badge_data['badges'][$i]['POINT_MIN'] != "0" &&
            $badge_data['badges'][$i]['CAT_SELECT'] == "0" &&
            $badge_data['badges'][$i]['CHECK_IN_COUNT'] == "0"){
                if((int)$badge_data['user'][0]['TOTAL_POINT'] >=
                (int)$badge_data['badges'][$i]['POINT_MIN']){
                    $this->insert_badge_to_user($badge_data['badges'][$i]['BADGE_ID'],$userid);
                    echo json_encode($badge_data['badges'][$i]);
                }
            }
        }

        if($badge_data['badges'][$i]['POINT_MIN'] == "0" &&
        $badge_data['badges'][$i]['CAT_SELECT'] != "0" &&
        $badge_data['badges'][$i]['CHECK_IN_COUNT'] != "0"){
            $cat_count['count'] = $this->tandatanya_model->get_category_id_by_userid($userid,$badge_data['badges'][$i]['CAT_SELECT']);
            if(count($cat_count['count']) >=
            (int)$badge_data['badges'][$i]['CHECK_IN_COUNT']){
                $this->insert_badge_to_user($badge_data['badges'][$i]['BADGE_ID'],$userid);
                echo json_encode($badge_data['badges'][$i]);
            }
        }
    }
}
```

```

    }
}

    if($badge_data['badges'][$i]['POINT_MIN'] != "0" &&
$badge_data['badges'][$i]['CAT_SELECT'] != "0" &&
$badge_data['badges'][$i]['CHECK_IN_COUNT'] == "0"){
    $cat_count['points'] = $this->tandatanya_model-
>count_total_point_category_by_user_id($userid,$badge_data['badges'][$i]['CAT_SELECT']);
    if((int)$cat_count['points'] >=
(int)$badge_data['badges'][$i]['POINT_MIN']){
        $this-
>insert_badge_to_user($badge_data['badges'][$i]['BADGE_ID'],$userid);
        echo json_encode($badge_data['badges'][$i]);
    }
}

    if($badge_data['badges'][$i]['POINT_MIN'] == "0" &&
$badge_data['badges'][$i]['CAT_SELECT'] == "0" &&
$badge_data['badges'][$i]['CHECK_IN_COUNT'] != "0"){
    $history_count['count'] = $this-
>tandatanya_model->get_all_history_by_userid($userid);
    if(count($history_count['count']) >=
(int)$badge_data['badges'][$i]['CHECK_IN_COUNT']){
        $this-
>insert_badge_to_user($badge_data['badges'][$i]['BADGE_ID'],$userid);
        echo json_encode($badge_data['badges'][$i]);
    } } } }
}

```

**Gambar 5.47 Fungsi accumulateBadge pada tandatanya.php**

Pada gambar 5.47 mula-mula fungsi ini mengambil data parameter 'userid' yang dilempar dari aplikasi menggunakan fungsi \$this->input->get('userid') yang kemudian akan memanggil fungsi check\_new\_user(\$userid) untuk mengambil info data user pada model tandatanya\_model.php seperti pada gambar 5.48.

```

public function check_new_user($userid){
    $this->db->select('*');
    $this->db->from('users');
    $this->db->where('USER_ID', $userid);
    $query = $this->db->get();
    return $query->result_array();
}

```

**Gambar 5.48 Fungsi check\_new\_user(\$userid) pada tandatanya\_model.php**

Kemudian fungsi akan memanggil fungsi `get_all_badges_not_in_badge_user_by_user_id($userid)` pada model `tandatanya_model.php` seperti pada gambar 5.49 yang digunakan untuk mengambil semua data *badge* yang tidak ada pada table 'badguser' yang ditampung pada `$badge_data['badges']`.

```
public function get_all_badges_not_in_badge_user_by_user_id($userid){
    $sql = "SELECT * FROM badges WHERE BADGE_ID NOT IN ( SELECT
    BADGE_ID FROM badguser WHERE USER_ID = ". $userid . " )";
    $query = $this->db->query($sql);
    return $query->result_array();
}
```

**Gambar 5.49 Fungsi `check_new_user($userid)` pada `tandatanya_model.php`**

Kemudian fungsi akan mengecek data *badge* yang tidak ada pada table 'badguser' kosong atau tidak kosong menggunakan fungsi `count($badge_data['badges']) > 0`, jika data tidak kosong maka akan dilakukan perulangan sesuai panjang array `$badge_data['badges']`, di dalam perulangan ini, fungsi akan melakukan empat buah pengecekan, yang pertama fungsi akan data 'MIN\_POINT' apakah tidak sama dengan 0 dan data 'CAT\_SELECT' apakah sama dengan 0 dan data 'CHECK\_IN\_COUNT' sama dengan 0, jika memenuhi syarat tersebut maka fungsi akan menambahkan data *badge* pada indeks 'I' kedalam table 'badguser' sesuai 'userid', begitu juga ketika pada pengecekan data 'MIN\_POINT' apakah sama dengan 0 dan data 'CAT\_SELECT' apakah tidak sama dengan 0 dan data 'CHECK\_IN\_COUNT' tidak sama dengan 0, dan juga pada pengecekan data 'MIN\_POINT' apakah tidak sama dengan 0 dan data 'CAT\_SELECT' apakah



tidak sama dengan 0 dan data 'CHECK\_IN\_COUNT' sama dengan 0, dan juga pada pengecekan data 'MIN\_POINT' apakah sama dengan 0 dan data 'CAT\_SELECT' apakah sama dengan 0 dan data 'CHECK\_IN\_COUNT' tidak sama dengan 0. Fungsi

`insert_badge_to_user($badge_data['badges'][$i]['BADGE_ID'],$userid)` akan memanggil fungsi seperti gambar 5.50.

```
public function insert_badge_to_user($badgeid, $userid){
    $badge_data = array(
        "BADGE_ID" => $badgeid,
        "USER_ID" => $userid
    );
    $this->tandatanya_model->insert_badge_to_user($badge_data);
}
```

**Gambar 5.50 Fungsi `check_new_user($userid)` pada `tandatanya_model.php`**

Fungsi pada gambar 5.50 digunakan untuk mengisikan data 'badge' dan 'user' ke dalam table 'badgeuser' yang menandakan bahwa user memenuhi syarat untuk mendapatkan sebuah badge. Fungsi `insert_badge_to_user($badge_data)` akan memanggil fungsi `insert_badge_to_user($data)` pada model `tandatanya_model.php` seperti pada gambar 5.51.

```
public function insert_badge_to_user($data){
    $this->db->insert('badgeuser', $data);
}
```

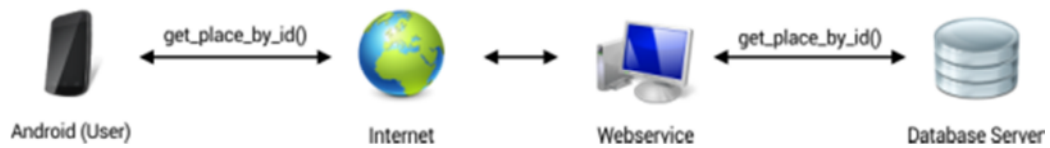
**Gambar 5.51 Fungsi `insert_badge_to_user($data)` pada `tandatanya_model.php`**

#### 5.2.1.9 Antarmuka Places



**Gambar 5.52 Antarmuka Tempat Wisata**

Antarmuka gambar 5.52 digunakan untuk menampilkan profil tempat wisata. Antarmuka ini muncul ketika user menampilkan sejarah user dan menekan salah satu list item sejarah user. Data yang ditampilkan terdiri dari foto profil, nama, alamat, dan deskripsi tempat wisata. Aplikasi mendapatkan data profil tempat wisata berdasarkan id tempat wisata dengan cara memanggil fungsi `get_place_by_id()` yang dimana fungsi akan memberikan nilai balikan berupa json dan kemudian menampilkannya. Proses ini dapat dilihat pada gambar 5.53.



**Gambar 5.53 Proses Mendapatkan Data Tempat Wisata**

Pada gambar 5.53 terdapat fungsi `get_place_by_id()` pada aplikasi yang digunakan untuk mengambil data info tempat wisata berdasarkan id tempat wisata, fungsi ini memanggil fungsi `get_place_info_by_id()` pada controller `tandatanya.php` di *web service* yang dapat dilihat pada gambar 5.54.

```

public function get_place_info_by_id(){
    $placeid = $this->input->get('placeid');
    $place_result = $this->tandatanya_model-
>get_place_by_id_and_one_photo($placeid);
    foreach ($place_result->result_array() as $row) {
        $place_data['id'] = $row['PLACE_ID'];
        $place_data['cat_place_id'] = $row['CAT_PLACE_ID'];
        $place_data['place_name'] = $row['PLACE_NAME'];
        $place_data['place_address'] = $row['PLACE_ADDRESS'];
        $place_data['place_description'] =
$row['PLACE_DESCRIPTION'];
        $place_data['place_lat'] = $row['PLACE_LAT'];
        $place_data['place_lon'] = $row['PLACE_LON'];
        $place_data['photo_url'] = $row['PHOTO_URL'];
    }
    echo json_encode($place_data);
}
  
```

**Gambar 5.54 Fungsi `get_place_info_by_id()` pada `tandatanya.php`**

Fungsi pada gambar 5.54 mula-mula mengambil data parameter id tempat wisata menggunakan fungsi `$this->input->get('placeid')` yang kemudian data parameter tersebut digunakan untuk mengambil data tempat wisata dengan memanggil fungsi `get_place_by_id_and_one_photo($placeid)` pada model `tandatanya_model.php` di *web service* yang fungsinya dapat dilihat pada gambar 5.55.

```

public function get_place_by_id_and_one_photo($placeid){
    $sql = "SELECT * FROM places JOIN photos ON places.PLACE_ID =
photos.PLACE_ID WHERE places.PLACE_ID = ".$placeid." GROUP BY
places.PLACE_ID";
    $query = $this->db->query($sql);
    return $query;
}

```

**Gambar 5.55 Fungsi `get_place_by_id_and_one_photo($placeid)` pada `tandatanya.php`**

Fungsi pada gambar 5.55 akan mengambil semua data tempat wisata dan satu buah foto profil tempat wisata yang ditampung pada `$place_result` yang kemudian dilakukan konversi ke dalam variable `$place_data` untuk dikirimkan ke aplikasi dalam bentuk json. Output json fungsi ini terdapat pada gambar 5.56.

```

{"id":"1","cat_place_id":"1","place_name":"Candi
Prambanan","place_address":"Jalan Raya Jogja - Solo Km 16, Prambanan,
Sleman, Yogyakarta 55571, Indonesia","place_description":"Candi
Prambanan adalah mahakarya kebudayaan Hindu dari abad ke-10.
Bangunannya yang langsing dan menjulang setinggi 47 meter membuat
kecantikan arsitekturnya tak tertandingi.,"place_lat":"-
7.75205","place_lon":"110.491966","photo_url":"http://localhost/ta
ndatanya/img_place/c8068ac420827421bfe27bd43f91715d.jpg"}

```

**Gambar 5.56 JSON Output fungsi `get_place_info_by_id()`**

Ketika json data info tempat wisata sudah didapatkan oleh aplikasi, maka aplikasi akan mengolah data json tersebut untuk ditampilkan menggunakan fungsi seperti pada gambar 5.57.

```

try {
    JSONObject joRoot = new JSONObject(arg0);
    placeResultById = new Place();
    placeResultById.setPlace_id(Integer.valueOf(joRoot.getString("
id"))));
    placeResultById.setPlace_name(joRoot.getString("place_name"));
    placeResultById.setPlace_address(joRoot.getString("place_adre
ss"));
    placeResultById.setPlace_description(joRoot.getString("place_d
escription"));
}

```

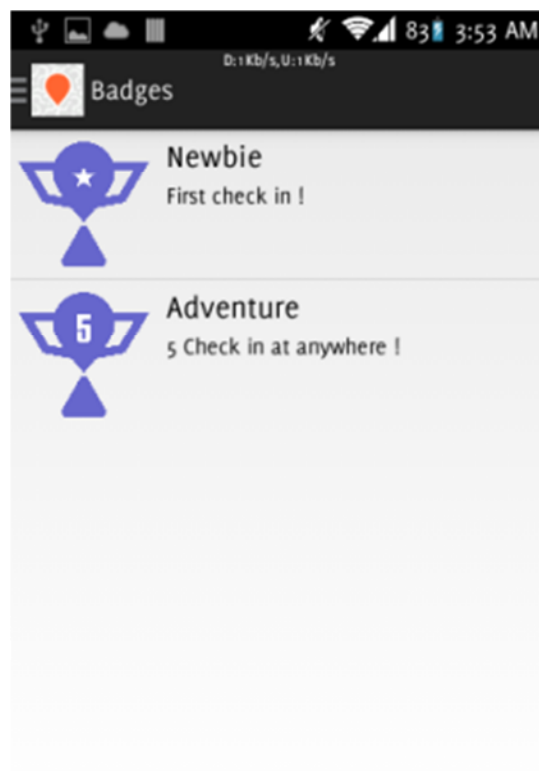
```

        placeResultById.setPlace_lat(Double.valueOf(joRoot.getString("
place_lat")));
        placeResultById.setPlace_lon(Double.valueOf(joRoot.getString("
place_lon")));
        placeResultById.setPhoto_url((joRoot.getString("photo_url").re
place("\\/", "").replace("http://localhost/",
"http://192.168.234.1/"));
        placeAct.onPlaceProfileDataReceived(placeResultById);
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

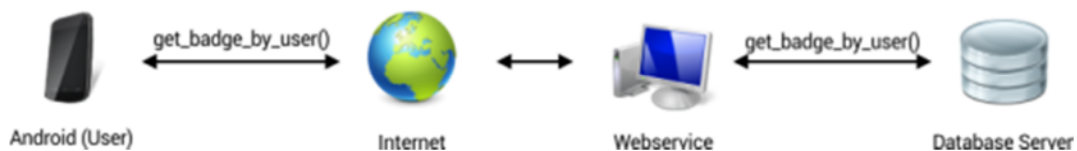
Gambar 5.57 Fungsi onSuccess() pada fungsi get\_place\_info\_by\_id()

#### 5.2.1.10 Antarmuka Badge Gallery



Gambar 5.58 Antarmuka Tempat Badge Gallery

Antarmuka gambar 5.58 digunakan untuk menampilkan *badge* galeri. *Badge* yang telah diperoleh *user* dari bermain kuis akan tampil di sini. *Badge* galeri dibentuk berupa *list* yang berisi *badge* ikon, nama, dan deskripsi. Ketika antarmuka ini diakses maka aplikasi akan memanggil fungsi `get_badge_by_user()` guna mendapatkan semua data *badge* yang dimiliki oleh *user* berdasarkan *user id*. Fungsi `get_badge_by_user()` memiliki data balikan berupa data *json*. Ketika data balikan sudah didapat maka aplikasi membentuk *badge list*. Proses ini dapat dilihat pada gambar 5.59.



**Gambar 5.59 Proses Mendapatkan Data *Badge***

Pada gambar 5.59 terdapat fungsi `get_badge_by_user()` pada aplikasi yang akan memanggil fungsi `get_badge_by_userid()` pada `controller tandatanya.php` di *web service* seperti pada gambar 5.60.

```

public function get_badge_by_userid(){
    $userid = $this->input->get('userid');
    $data['badgeuser'] = $this->tandatanya_model-
>get_badge_by_userid($userid);
    echo json_encode($data);
}
  
```

**Gambar 5.60 Fungsi `get_badge_by_userid()` pada `tandatanya.php`**

Pada gambar 5.60 fungsi mula-mula akan mengambil data *user id* yang dilempar dari aplikasi menggunakan fungsi `$this->input->get('userid')` yang kemudian digunakan untuk mengambil seluruh data '*badgeuser*' dari

basis data menggunakan fungsi `get_badge_by_userid($userid)` pada model `tandatanya_model.php` seperti pada gambar 5.61 dan menampungnya pada `$data['badgeuser']` dan melemparkannya sebagai data balikan ke aplikasi dengan tipe json. JSON output fungsi ini dapat dilihat pada gambar 5.62.

```
public function get_badge_by_userid($userid){
    $sql = "SELECT * FROM badgeuser JOIN badges ON
    badgeuser.BADGE_ID = badges.BADGE_ID WHERE USER_ID = ".$userid;
    $query = $this->db->query($sql);
    return $query->result_array();
}
```

Gambar 5.61 Fungsi `get_badge_by_userid()` pada `tandatanya_model.php`

```
{"badgeuser":[{"BADGE_ID":"13","USER_ID":"1504697984","BADGE_NAME":"Newbie","DESCRIPTION":"First check in!", "ICON":"http://tandatanya.hol.es/76766295247ccf63c06b06062123f686.png","POINT_MIN":"0","CAT_SELECT":"0","CHECK_IN_COUNT":"1"}, {"BADGE_ID":"14","USER_ID":"1504697984","BADGE_NAME":"Adventure","DESCRIPTION":"5 Check in at anywhere!", "ICON":"http://tandatanya.hol.es/ef7c4ef610a4145641556aab99450978.png","POINT_MIN":"0","CAT_SELECT":"0","CHECK_IN_COUNT":"5"}]}
```

Gambar 5.62 JSON Output fungsi `get_badge_by_userid()` pada `tandatanya.php`

Ketika data *badge* dalam bentuk json sudah didapatkan dari *web service* maka aplikasi akan mengolah data json tersebut menggunakan fungsi pada gambar 5.63 untuk ditampilkan ke dalam list *badge*.

```
try {
    JSONObject joRoot = new JSONObject(arg0);
    JSONArray jaBadges = joRoot.getJSONArray("badgeuser");
    for (int i = 0; i < jaBadges.length(); i++) {
        Badge badge = new Badge();
        JSONObject joBadge = jaBadges
            .getJSONObject(i);
        badge.setBadge_id(Integer.valueOf(joBadge.getString("BADGE_ID"
    )));
        badge.setBadge_name(joBadge.getString("BADGE_NAME"));
    }
```

```

        badge.setBadge_desc(joBadge.getString("DESCRIPTION"));
        badge.setBadge_icon((joBadge.getString("ICON").replace("\\\\/",
        "").replace("http://tandatanya.hol.es/",
        "http://192.168.234.1/tandatanya/img_badges/"));

        arlBadge.add(badge);
    }
    badgeFragment.onBadgeuserDataReceived(arlBadge);
} catch (JSONException e) {
// TODO Auto-generated catch block
    e.printStackTrace();
}

```

Gambar 5.63 Fungsi OnSucces() pada get\_badge\_by\_userid()

## 5.2.2 Antarmuka TANDATANYA-Web

### 5.2.2.1 Antarmuka Login



Gambar 5.64 Antarmuka Login

Antarmuka ini digunakan untuk memasukkan *username* dan *password* agar Admin dapat masuk ke dalam system. Ketika halaman ini diakses, system akan memanggil controller tandmin.php dan memanggil fungsi index(). Proses ini dapat dilihat pada gambar 5.65.





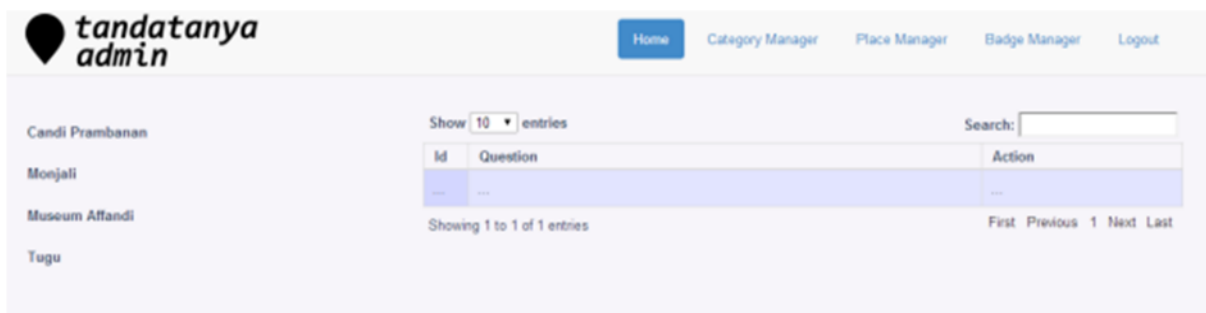
Gambar 5.65 Proses Menampilkan Halaman Log In

Pada saat login, system memanggil controller tandmin.php, kemudian memanggil fungsi doLogin() dan melakukan pengecekan data login. Jika sesuai, maka controller akan memanggil fungsi home() untuk berpindah ke halaman utama. Proses ini dapat dilihat pada gambar 5.66.



Gambar 5.66 Proses Log In

### 5.2.2.2 Antarmuka Home



Gambar 5.67 Antarmuka Home

Antarmuka 5.67 merupakan halaman utama setelah Admin melakukan login. Antarmuka ini sekaligus menjadi antarmuka untuk pengelolaan data pertanyaan, dimana ketika list tempat wisata yang berada di sisi sebelah kiri ditekan maka controller akan memanggil fungsi get\_question\_by\_id() yang mengembalikan data pertanyaan

dalam bentuk json, sehingga table pertanyaan di sebelah kanan akan menampilkan pertanyaan yang terdapat pada tempat wisata sesuai pilihan Admin berupa id dan pertanyaan. Proses ini dapat di lihat pada gambar 5.68.



**Gambar 5.68 Proses Mengambil Pertanyaan Berdasarkan Tempat Wisata**

Pada gambar 5.68 terdapat fungsi `get_question_by_id()` pada controller `tandmin.php` yang dipanggil ketika list pertanyaan dipilih, fungsi ini digunakan untuk mengambil data pertanyaan berdasarkan id tempat wisata, fungsi ini dapat dilihat pada gambar 5.69.

```

public function get_question_by_id(){
    $data = array();
    $data = $this->tandatanya_model->get_question_by_id($this->input->post('idIcon'));
    echo json_encode($data);
}
  
```

**Gambar 5.69 Fungsi `get_question_by_id()` pada `tandmin.php`**

Fungsi pada gambar 5.69 akan mengambil semua data pertanyaan berdasarkan id tempat wisata yang kemudian menampungnya dalam variable `$data` dan melemparkannya sebagai data balikan bertipe json seperti pada gambar 5.70.

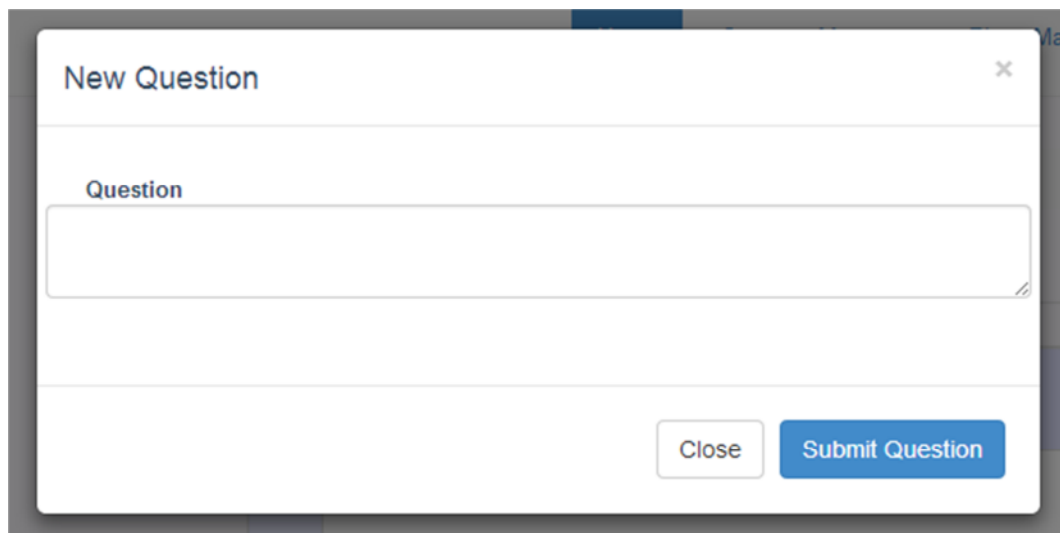
```

[Object { QUESTION_ID="0", PLACE_ID="1", QUESTION="Candi Prambanan merupakan relief sebuah kisah?"}, Object { QUESTION_ID="1", PLACE_ID="1", QUESTION="Terletak di manakah candi prambanan?"},
  
```

```
Object { QUESTION_ID="19", PLACE_ID="1", QUESTION="Pada abad ke
berapa Candi Prambanan dibangun?"}, Object { QUESTION_ID="20",
PLACE_ID="1", QUESTION="Siapakah nama laki-laki...cerita Candi
Prambanan?"}, Object { QUESTION_ID="21", PLACE_ID="1",
QUESTION="Siapakah nama seorang p... kisah Candi Prambanan?"}, Object
{ QUESTION_ID="22", PLACE_ID="1", QUESTION="Candi Prambanan memiliki
3 candi utama, yaitu?"}, Object { QUESTION_ID="23", PLACE_ID="1",
QUESTION="Berapa ruangan yang terdapat di dalam Candi Siwa?"}]
```

Gambar 5.70 JSON Output fungsi `get_question_by_id()` pada `tandmin.php`

### 5.2.2.3 Antarmuka Tambah Pertanyaan



Gambar 5.71 Antarmuka Tambah Pertanyaan

Pada antarmuka gambar 5.71, terdapat form data pertanyaan, tombol 'Close' dan 'Submit Question'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* tambah data pertanyaan, sedangkan tombol 'Submit Question' digunakan untuk menyimpan data pertanyaan baru sesuai isi form ke dalam basis data. Ketika tombol 'Submit Question' ditekan maka system akan memanggil fungsi `insert_new_question()` pada controller `tandmin.php` yang dimana fungsi ini akan mengambil data parameter yang dilemparkan yaitu

'question' dan 'place\_id' menggunakan fungsi `$this->input->post()` dan kemudian fungsi akan memanggil fungsi `insert_new_question_by_place_id($question_data)` pada model `tandatanya_model.php` untuk menambahkan data pertanyaan baru. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.72.



Gambar 5.72 Proses Tambah Data Pertanyaan

#### 5.2.2.4 Antarmuka Edit Data Pertanyaan

Question Candi Prambanan merupakan relief sebuah kisah?

Close Save changes

Gambar 5.73 Proses Tambah Data Pertanyaan

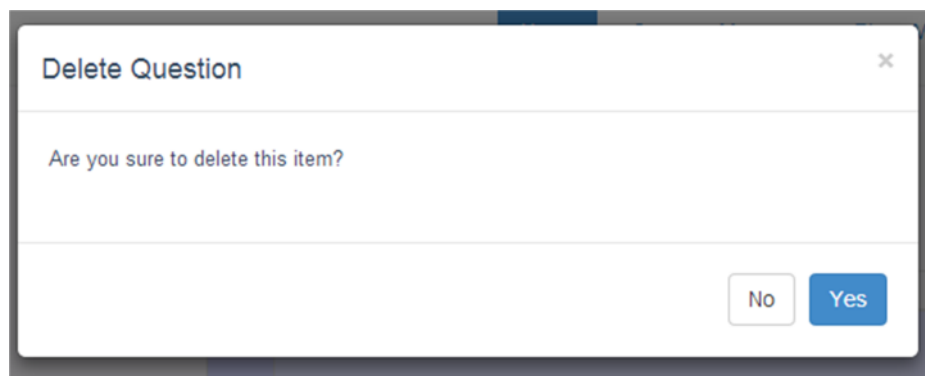
Pada antarmuka gambar 5.73, terdapat form edit data pertanyaan, tombol 'Close' dan 'Save changes'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* edit data pertanyaan, sedangkan tombol 'Save changes' digunakan untuk menyimpan data perubahan pertanyaan sesuai isi form ke dalam basis data berdasarkan id pertanyaan. Ketika tombol 'Save changes' ditekan maka system akan memanggil fungsi `edit_question()` yang mula-mula fungsi akan mengambil data parameter yang dilempar berupa data pertanyaan baru dan id pertanyaan, kemudian fungsi akan memanggil fungsi

`edit_question($edit_question_data,$edit_question_id)` pada model `tandatanya_model.php` untuk meng-*update* data pertanyaan baru ke dalam basis data. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.74



Gambar 5.74 Proses Edit Data Pertanyaan

#### 5.2.2.5 Antarmuka Hapus Pertanyaan



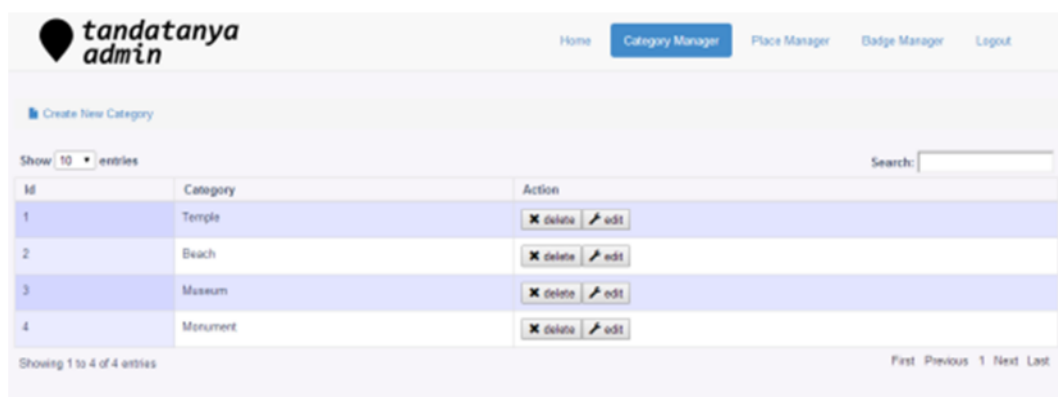
**Gambar 5.75 Antarmuka Hapus Pertanyaan**

Antarmuka gambar 5.75 merupakan antarmuka halaman yang digunakan admin untuk menghapus data pertanyaan sesuai id yang didapatkan ketika tombol 'X Delete' pada kolom 'Action' ditekan. Pada antarmuka gambar 5.75 terdapat tombol 'No' dan 'Yes' dimana tombol 'No' digunakan untuk membatalkan penghapusan data pertanyaan sedangkan fungsi tombol 'Yes' digunakan untuk melakukan penghapusan data pertanyaan. Sehingga ketika tombol 'Yes' ditekan maka system akan memanggil fungsi `delete_question()` pada controller `tandmin.php`, maka data pertanyaan yang dikehendaki akan terhapus dari basis data. Proses ini dapat dilihat pada gambar 5.78.



**Gambar 5.78 Proses Hapus Pertanyaan**

### 5.2.2.6 Antarmuka Category Manager



**Gambar 5.79 Antarmuka Category Manager**

Antarmuka gambar 5.79 merupakan halaman pengelolaan data kategori tempat wisata. Antarmuka gambar 5.79 memiliki table yang terdiri dari kolom 'Id', 'Category', dan 'Action' sebagai table dari data kategori tempat wisata, dan juga tombol 'Create New Category' yang digunakan untuk menampilkan antarmuka penambahan data kategori tempat wisata. Sehingga ketika antarmuka gambar 5.79 ini diakses, maka system akan memanggil fungsi `category_manager()` pada controller `tandmin.php` untuk mengambil semua data kategori tempat wisata dari basis data dan membalikkan data pada antarmuka. Proses ini dapat di lihat pada gambar 5.80.



Gambar 5.80 Proses mendapatkan data kategori tempat wisata

#### 5.2.2.7 Antarmuka Tambah Data Kategori Tempat Wisata

Gambar 5.81 Antarmuka Tambah Data Kategori Tempat Wisata

Pada antarmuka gambar 5.81, terdapat form data kategori tempat wisata, tombol 'Close' dan 'Submit Category'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* tambah data kategori tempat wisata, sedangkan tombol 'Submit Category' digunakan untuk menyimpan data kategori tempat wisata baru sesuai isi form ke dalam basis data. Ketika tombol 'Submit Category' ditekan maka system akan memanggil fungsi `insert_new_category()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.82.



Gambar 5.82 Proses Tambah Data Kategori Tempat Wisata

#### 5.2.2.8 Antarmuka Edit Data Kategori Tempat Wisata

Gambar 5.83 Antarmuka Edit Data Kategori Tempat Wisata

Pada antarmuka gambar 5.83, terdapat form edit data kategori tempat wisata, tombol 'Close' dan 'Submit

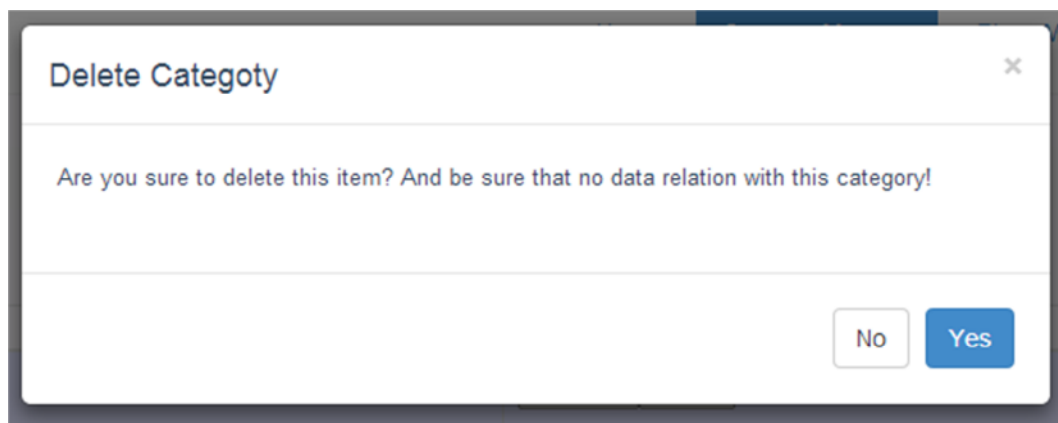


Category'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* edit data kategori tempat wisata, sedangkan tombol 'Submit Category' digunakan untuk menyimpan data perubahan kategori tempat wisata sesuai isi form ke dalam basis data berdasarkan id kategori tempat wisata. Ketika tombol 'Submit Category' ditekan maka system akan memanggil fungsi `edit_category()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.84



Gambar 5.84 Proses Edit Data Kategori Tempat Wisata

#### 5.2.2.9 Antarmuka Hapus Data Kategori Tempat Wisata



Gambar 5.85 Antarmuka Hapus Data Kategori Tempat Wisata

Antarmuka gambar 5.85 merupakan antarmuka halaman yang digunakan admin untuk menghapus data kategori tempat wisata sesuai id yang didapatkan ketika tombol 'X Delete' pada kolom 'Action' ditekan. Pada antarmuka

gambar 5.85 terdapat tombol 'No' dan 'Yes' dimana tombol 'No' digunakan untuk membatalkan penghapusan data kategori tempat wisata sedangkan fungsi tombol 'Yes' digunakan untuk melakukan penghapusan data kategori tempat wisata. Sehingga ketika tombol 'Yes' ditekan maka system akan memanggil fungsi `delete_category()`, maka data pertanyaan yang dikehendaki akan terhapus dari basis data. Proses ini dapat dilihat pada gambar 5.86



Gambar 5.86 Proses Hapus Data Kategori Tempat Wisata

### 5.2.2.10 Antarmuka Place Manager

Id	Name	Category	Address	Description	Latitude	Longitude	Action
1	Candi Prambanan	Temple	Jalan Raya Joga - Solo Km 16, Prambanan, Sleman, Yogyakarta 55771, Indonesia	Candi Prambanan adalah mahakarya kebudayaan Hindu dari abad ke-10. Bangunannya yang langung dan menjulang setinggi 47 meter membuat kecantikan arsitekturnya tak tertandingi.	-7.75205	110.491966	<a href="#">delete</a> <a href="#">edit</a> <a href="#">show photo</a>
2	Merjali	Museum	J. Lingkar Utara, Yogyakarta 55581, Indonesia	Museum Monumen Yogya Kembali, adalah sebuah museum sejarah perjuangan kemerdekaan Republik Indonesia yang ada di kota Yogyakarta dan dikelola oleh Departemen Kebudayaan dan Pariwisata. Museum yang berada di bagian utara kota ini banyak dikunjungi ole	-7.749945	110.389593	<a href="#">delete</a> <a href="#">edit</a> <a href="#">show photo</a>
10	Museum Afandi	Museum	J. Laksa Adisucipto 167, Yogyakarta 55281, Indonesia	Museum afandi adalah seluruh bagian dari kehidupan Afandi sebagai maestro seni kalisa. Di wilayah tepi sungai Gajah Wong itu, Afandi hidup, berkarya, mentransformasikan ilmunya dan beresmayam di rumah abadinya.	-7.783154	110.396154	<a href="#">delete</a> <a href="#">edit</a> <a href="#">show photo</a>

Gambar 5.87 Antarmuka Place Manager

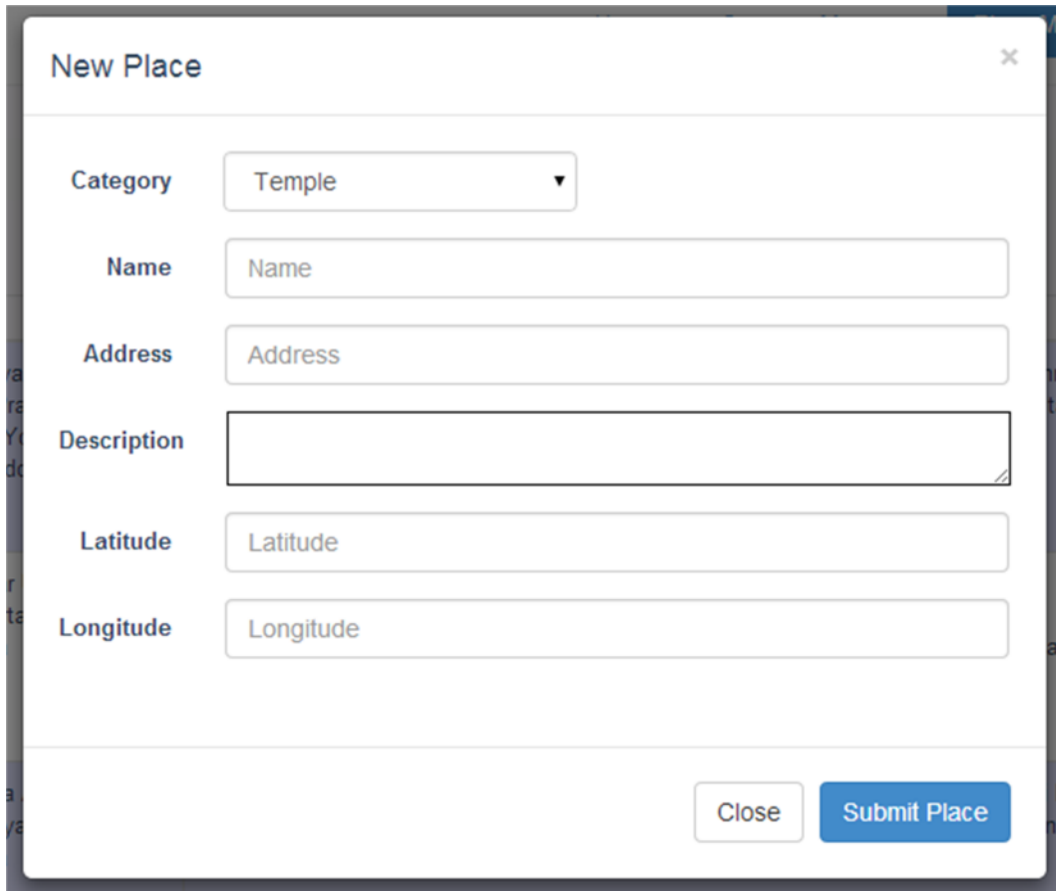
Antarmuka gambar 5.87 merupakan halaman pengelolaan data tempat wisata. Antarmuka gambar 5.87 memiliki table yang terdiri dari kolom 'Id', 'Name', 'Category', 'Address', 'Description', 'Latitude',

'Longitude' dan 'Action' sebagai table dari data tempat wisata, dan juga tombol 'Create New Place' yang digunakan untuk menampilkan antarmuka penambahan data tempat wisata. Sehingga ketika antarmuka gambar 5.87 ini diakses, maka system akan memanggil fungsi `place_manager()` untuk mengambil semua data tempat wisata dari basis data dan membalikkan data pada antarmuka dalam bentuk json. Proses ini dapat di lihat pada gambar 5.88.



**Gambar 5.88 Proses Mendapatkan Data Tempat Wisata**

#### 5.2.2.11 Antarmuka Tambah Data Tempat Wisata



The image shows a web form titled "New Place" with a close button in the top right corner. The form contains the following fields:

- Category:** A dropdown menu with "Temple" selected.
- Name:** A text input field with the placeholder "Name".
- Address:** A text input field with the placeholder "Address".
- Description:** A larger text area with a placeholder and a small icon in the bottom right corner.
- Latitude:** A text input field with the placeholder "Latitude".
- Longitude:** A text input field with the placeholder "Longitude".

At the bottom right of the form, there are two buttons: a "Close" button and a "Submit Place" button.

Gambar 5.89 Antarmuka Tambah Data Tempat Wisata

Pada antarmuka gambar 5.89, terdapat form data tempat wisata, tombol 'Close' dan 'Submit Place'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* tambah data tempat wisata, sedangkan tombol 'Submit Place' digunakan untuk menyimpan data tempat wisata baru sesuai isi form ke dalam basis data. Ketika tombol 'Submit Place' ditekan maka system akan memanggil fungsi `insert_new_place()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.90.



Gambar 5.90 Proses Tambah Data Tempat Wisata

#### 5.2.2.12 Antarmuka Edit Data Tempat Wisata

Gambar 5.91 Antarmuka Edit Data Tempat Wisata

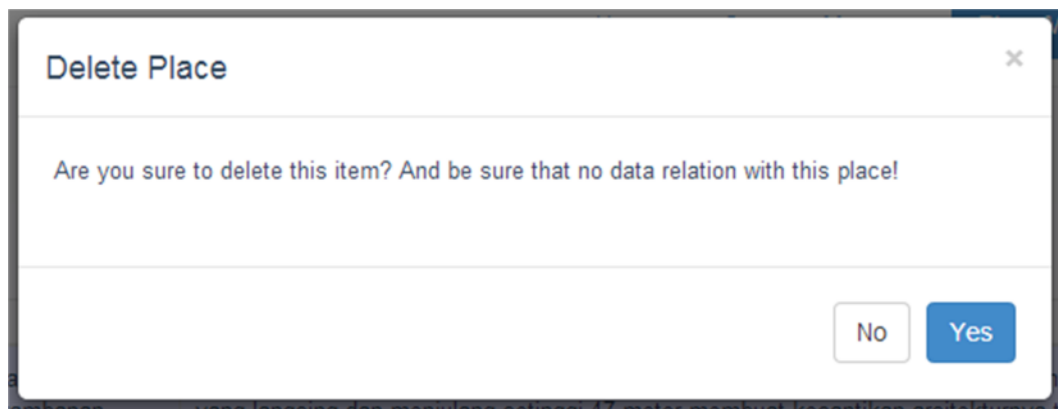
Pada antarmuka gambar 5.91, terdapat form edit data tempat wisata, tombol 'Close' dan 'Submit Place'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* edit data tempat wisata, sedangkan tombol 'Submit Place' digunakan untuk menyimpan data perubahan

tempat wisata sesuai isi form ke dalam basis data berdasarkan id tempat wisata. Ketika tombol 'Submit Place' ditekan maka system akan memanggil fungsi `edit_place()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.92



Gambar 5.92 Proses Edit Data Tempat Wisata

#### 5.2.2.13 Antarmuka Hapus Data Tempat Wisata



Gambar 5.93 Antarmuka Hapus Data Tempat Wisata

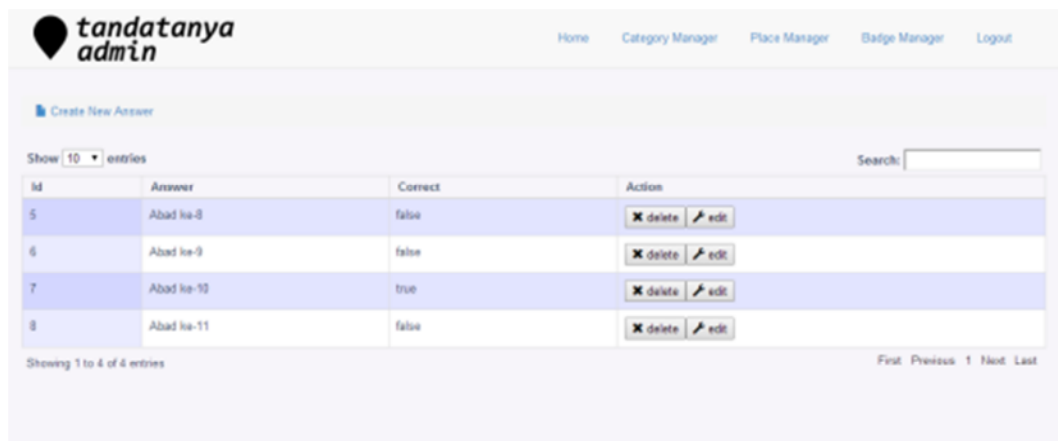
Antarmuka gambar 5.93 merupakan antarmuka halaman yang digunakan admin untuk menghapus data tempat wisata sesuai id yang didapatkan ketika tombol 'X Delete' pada kolom 'Action' ditekan. Pada antarmuka gambar 5.93 terdapat tombol 'No' dan 'Yes' dimana tombol 'No' digunakan untuk membatalkan penghapusan data tempat wisata sedangkan fungsi tombol 'Yes' digunakan untuk melakukan penghapusan data tempat wisata. Sehingga

ketika tombol 'Yes' ditekan maka system akan memanggil fungsi `delete_place()`, maka data pertanyaan yang dikehendaki akan terhapus dari basis data. Proses ini dapat dilihat pada gambar 5.94



Gambar 5.94 Proses Hapus Data Tempat Wisata

#### 5.2.2.14 Antarmuka Answer Manager



Gambar 5.95 Antarmuka Answer Manager

Antarmuka gambar 5.95 merupakan halaman pengelolaan data jawaban. Antarmuka gambar 5.95 memiliki table yang terdiri dari kolom 'Id', 'Answer', 'Correct', dan 'Action' sebagai table dari data jawaban, dan juga tombol 'Create New Answer' yang digunakan untuk menampilkan antarmuka penambahan data jawaban. Sehingga ketika antarmuka gambar 5.95 ini diakses, maka system akan memanggil fungsi `show_answer()` untuk mengambil data jawaban sesuai jawaban yang dipilih dari basis data dan membalikkan

data pada antarmuka. Proses ini dapat di lihat pada gambar 5.96.



Gambar 5.96 Proses Mendapatkan Data Pertanyaan

#### 5.2.2.15 Antarmuka Tambah Data Jawaban

New Answer

Answer

Correct

Close Submit answer

Gambar 5.97 Antarmuka Tambah Data Jawaban

Pada antarmuka gambar 5.97, terdapat form data jawaban, tombol 'Close' dan 'Submit Answer'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* tambah data jawaban, sedangkan tombol 'Submit Answer' digunakan untuk menyimpan data jawaban baru sesuai isi form ke dalam basis data. Ketika tombol 'Submit Answer' ditekan maka system akan memanggil fungsi `insert_new_answer()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.98.





Gambar 5.98 Proses Tambah Data Pertanyaan

#### 5.2.2.16 Antarmuka Edit Data Jawaban

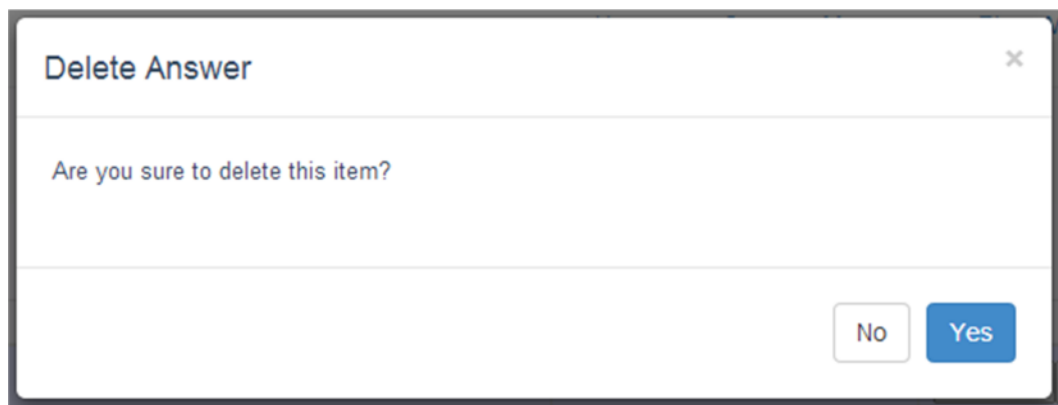
Gambar 5.99 Antarmuka Edit Data Jawaban

Pada antarmuka gambar 5.99, terdapat form edit data jawaban, tombol 'Close' dan 'Save changes'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* edit data jawaban, sedangkan tombol 'Save changes' digunakan untuk menyimpan data perubahan jawaban sesuai isi form ke dalam basis data berdasarkan id jawaban. Ketika tombol 'Save changes' ditekan maka system akan memanggil fungsi `edit_answer()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.100



Gambar 5.100 Proses Edit Data Jawaban

#### 5.2.2.17 Antarmuka Hapus Data Jawaban



Gambar 5.101 Antarmuka Hapus Data Jawaban

Antarmuka gambar 5.101 merupakan antarmuka halaman yang digunakan admin untuk menghapus data jawaban sesuai id yang didapatkan ketika tombol 'X Delete' pada kolom 'Action' ditekan. Pada antarmuka gambar 5.101 terdapat tombol 'No' dan 'Yes' dimana tombol 'No' digunakan untuk membatalkan penghapusan data jawaban sedangkan fungsi tombol 'Yes' digunakan untuk melakukan penghapusan data jawaban. Sehingga ketika tombol 'Yes' ditekan maka system akan memanggil fungsi `delete_answer()`, maka data pertanyaan yang dikehendaki akan terhapus dari basis data. Proses ini dapat dilihat pada gambar 5.102.



Gambar 5.102 Proses Hapus Data Jawaban

### 5.2.2.18 Antarmuka Badge Manager

Id	Name	Description	Icon	Minimal Point	Selected Category	Check In Count	Action
1	Temple Hunter	Congratulation , you have checked into 5 Temples!	0	0	1	5	<a href="#">delete</a> <a href="#">edit</a>
6	Nearbe Adventure	Congratulation for your first check in !	0	0	0	1	<a href="#">delete</a> <a href="#">edit</a>
7	Adventure	Congratulation , you have checked into 5 places !	0	0	0	5	<a href="#">delete</a> <a href="#">edit</a>
8	Explorer	Congratulation , you have checked into 10 places !	0	0	0	10	<a href="#">delete</a> <a href="#">edit</a>
9	Addict	Congratulation , you have checked into 20 places !	0	0	0	20	<a href="#">delete</a> <a href="#">edit</a>
10	Museum Hunter	Congratulation , you have checked into 5 Museums !	0	0	3	5	<a href="#">delete</a> <a href="#">edit</a>
11	Beach Hunter	Congratulation , you have checked into 5 Beaches!	0	0	2	5	<a href="#">delete</a> <a href="#">edit</a>

Gambar 5.103 Antarmuka Badge Manager

Antarmuka gambar 5.103 merupakan halaman pengelolaan data *badge*. Antarmuka gambar 5.103 memiliki table yang terdiri dari kolom 'Id', 'Name', 'Description', 'Icon', 'Minimal Point', 'Selected Category', 'Check In Count', dan 'Action' sebagai table dari data *badge*, dan juga tombol 'Create New Badge' yang digunakan untuk menampilkan antarmuka penambahan data *badge*. Sehingga ketika antarmuka gambar 5.103 ini diakses, maka system akan memanggil fungsi `badge_manager()` untuk mengambil data *badge* sesuai *badge* yang dipilih dari basis data dan membalikkan data pada antarmuka dalam bentuk json. Proses ini dapat di lihat pada gambar 5.104.



Gambar 5.104 Proses Mendapatkan Data Badge

#### 5.2.2.19 Antarmuka Tambah Data Badge

Gambar 5.105 Antarmuka Tambah Data Badge

Pada antarmuka gambar 5.105, terdapat form data badge, tombol 'Close' dan 'Submit Badge'. Tombol

'Close' digunakan untuk membatalkan sekaligus menutup *modal* tambah data *badge*, sedangkan tombol 'Submit Badge' digunakan untuk menyimpan data *badge* baru sesuai isi form ke dalam basis data. Ketika tombol 'Submit Badge' ditekan maka system akan memanggil fungsi `insert_new_badge()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.106.



Gambar 5.106 Proses Tambah Data Badge

#### 5.2.2.20 Antarmuka Edit Data Badge

**Edit Badge** [Close]

Name: Temple Hunter

Description: Congratulation , you have checked into 5 Temples!

Use Minimal Point

Minimal Point: 0

Use Category Manager

Category: Temple

Use Check In Count

Check In Count: 5

**File**

Tidak ada file yang dipilih

[Close] [Submit Badge]

Gambar 5.107 Antarmuka Edit Data Badge

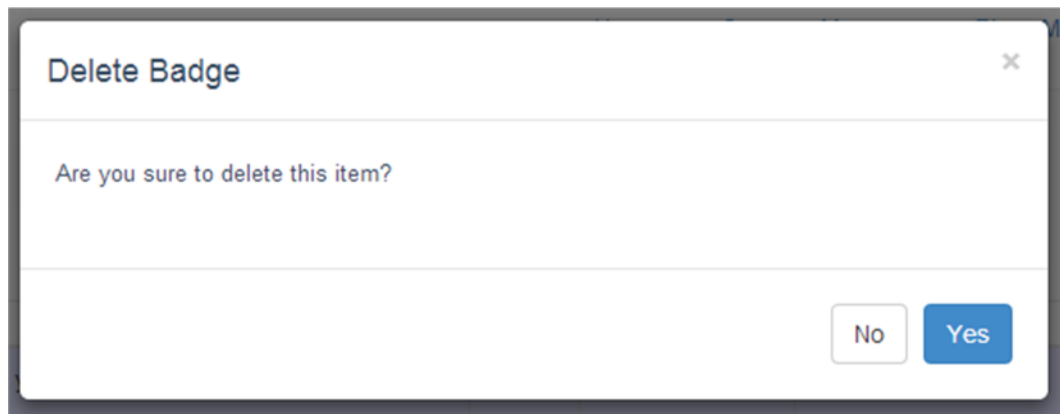
Pada antarmuka gambar 5.107, terdapat form edit data *badge*, tombol 'Close' dan 'Submit Badge'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* edit data *badge*, sedangkan tombol 'Submit Badge' digunakan untuk menyimpan data perubahan *badge* sesuai isi form ke dalam basis data berdasarkan id *badge*. Ketika tombol 'Submit Badge' ditekan maka system akan

memanggil fungsi `edit_badge()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.108



Gambar 5.109 Proses Edit Data Badge

#### 5.2.2.21 Antarmuka Hapus Data Badge



Gambar 5.110 Antarmuka Hapus Data Badge

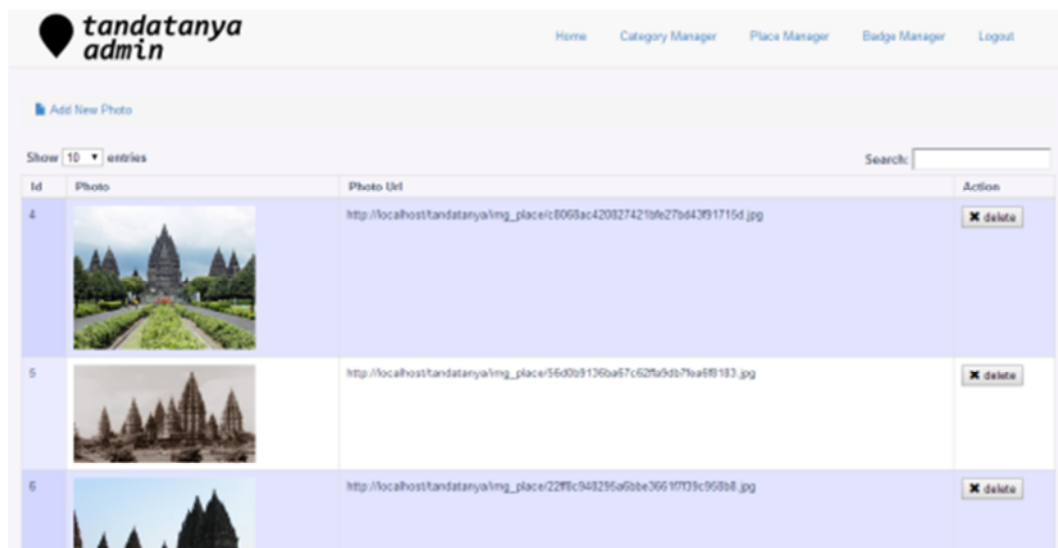
Antarmuka gambar 5.110 merupakan antarmuka halaman yang digunakan admin untuk menghapus data *badge* sesuai id yang didapatkan ketika tombol 'X Delete' pada kolom 'Action' ditekan. Pada antarmuka gambar 5.110 terdapat tombol 'No' dan 'Yes' dimana tombol 'No' digunakan untuk membatalkan penghapusan data *badge* sedangkan fungsi tombol 'Yes' digunakan untuk melakukan penghapusan data *badge*. Sehingga ketika tombol 'Yes' ditekan maka system akan memanggil fungsi `delete_badge()`, maka data *badge* yang dikehendaki akan

terhapus dari basis data. Proses ini dapat dilihat pada gambar 5.111



Gambar 5.111 Proses Hapus Data Badge

#### 5.2.2.22 Antarmuka Photo Manager



Gambar 5.112 Antarmuka Photo Manager

Antarmuka gambar 5.112 merupakan halaman pengelolaan data foto. Antarmuka gambar 5.112 memiliki table yang terdiri dari kolom 'Id', 'Photo', 'Photo Url', dan 'Action' sebagai table dari data foto, dan juga tombol 'Add New Photo' yang digunakan untuk menampilkan antarmuka penambahan data foto. Sehingga ketika antarmuka gambar 5.112 ini diakses, maka system akan memanggil fungsi photo\_manager() untuk mengambil data foto sesuai foto yang dipilih dari basis data dan

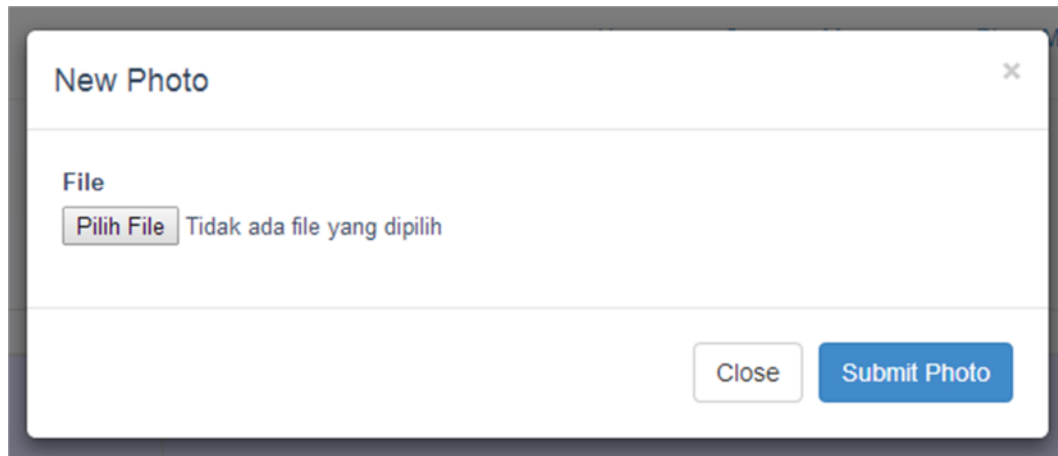


membalikkan data pada antarmuka dalam bentuk json. Proses ini dapat di lihat pada gambar 5.113.



Gambar 5.113 Proses Mendapatkan Data Foto

#### 5.2.2.23 Antarmuka Tambah Data Foto



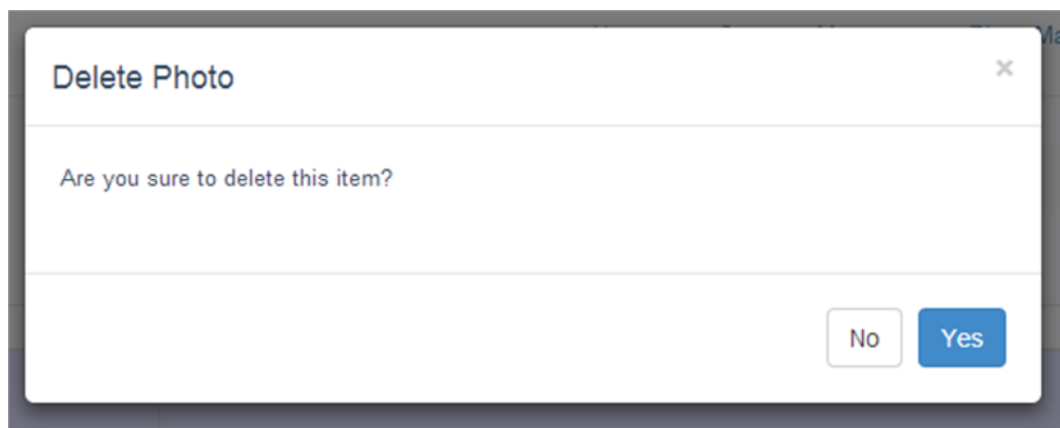
Gambar 5.114 Antarmuka Tambah Data Foto

Pada antarmuka gambar 5.114, terdapat form data foto, tombol 'Close' dan 'Submit Foto'. Tombol 'Close' digunakan untuk membatalkan sekaligus menutup *modal* tambah data foto, sedangkan tombol 'Submit Photo' digunakan untuk menyimpan data foto baru sesuai isi form ke dalam basis data. Ketika tombol 'Submit Photo' ditekan maka system akan memanggil fungsi `upload_file()`. Setelah itu, controller mengembalikan pesan sukses menuju halaman utama. Proses ini dapat dilihat pada gambar 5.115.



Gambar 5.115 Proses Tambah Data Foto

#### 5.2.2.24 Antarmuka Hapus Data Foto



Gambar 5.116 Antarmuka Hapus Data Foto

Antarmuka gambar 5.116 merupakan antarmuka halaman yang digunakan admin untuk menghapus data foto sesuai id yang didapatkan ketika tombol 'X Delete' pada kolom 'Action' ditekan. Pada antarmuka gambar 5.116 terdapat tombol 'No' dan 'Yes' dimana tombol 'No' digunakan untuk membatalkan penghapusan data foto sedangkan fungsi tombol 'Yes' digunakan untuk melakukan penghapusan data foto. Sehingga ketika tombol 'Yes' ditekan maka system akan memanggil fungsi `delete_photo()`, maka data foto yang dikehendaki akan terhapus dari basis data. Proses ini dapat dilihat pada gambar 5.117

### 5.3 Hasil Pengujian Perangkat Lunak

Pada bagian ini akan ditampilkan hasil pengujian setiap fungsi dari sistem yang telah dikembangkan. Bentuk pengujian dan keterangannya dapat dilihat di tabel 5.9.

Tabel 5.8 Tabel Hasil Pengujian Perangkat Lunak

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
TT-001	Pengujian Login	- Mengisi username dan password	- Textbox username = anombhayu.maharsi@yahoo.co.id - Password = 2ndemail	- Masuk ke halaman utama aplikasi TANDATANYA - Tampil halaman utama aplikasi	- Masuk ke halaman utama aplikasi TANDATANYA - Tampil halaman utama aplikasi	- Masuk ke halaman utama aplikasi TANDATANYA - Tampil halaman utama aplikasi	Handal
TT-002	Pengujian Logout	- Menekan menu 'Logout'	-	- Keluar dari aplikasi, tampil halaman login	- Keluar dari aplikasi, tampil halaman login	- Keluar dari aplikasi, tampil halaman login	Handal
TT-003	Pengujian Tampil Profil Pengguna	- Menekan menu 'Home'	-	- Tampil profil user berupa foto, nama, dan total skor	- Tampil profil user berupa foto, nama, dan total skor	- Tampil profil user berupa foto, nama, dan total skor	Handal
TT-004	Pengujian Tampil Sejarah Pengguna	- Menekan menu 'History'	-	- Tampil sejarah user dalam bentuk list berupa foto profil, nama, skor tempat wisata	- Tampil sejarah user dalam bentuk list berupa foto profil, nama, skor tempat wisata	- Tampil sejarah user dalam bentuk list berupa foto profil, nama, skor tempat wisata	Handal
TT-005	Pengujian Tampil	- Menekan menu 'Friends'	-	- Tampil teman Facebook user	- Tampil teman Facebook user	- Tampil teman Facebook user	Handal

	Temannya Pengguna			yang menggunakan aplikasi TANDATANYA dalam bentuk list berupa foto profil, nama, skor.	yang menggunakan aplikasi TANDATANYA dalam bentuk list berupa foto profil, nama, skor.	yang menggunakan aplikasi TANDATANYA dalam bentuk list berupa foto profil, nama, skor.	
TT-006	Pengujian Tampil Tempat Wisata pada Peta	- Menekan tombol 'Play' pada antarmuka 'Home'	-	- Tampil lokasi tempat-tempat wisata pada peta.	- Tampil lokasi tempat-tempat wisata pada peta.	- Tampil lokasi tempat-tempat wisata pada peta.	Handal
TT-007	Pengujian Tampil Profil Tempat Wisata	- Menekan list item pada antarmuka 'History'	-	- Tampil profil tempat wisata terpilih berupa foto profil, nama, alamat, dan deskripsi tempat wisata	- Tampil profil tempat wisata terpilih berupa foto profil, nama, alamat, dan deskripsi tempat wisata	- Tampil profil tempat wisata terpilih berupa foto profil, nama, alamat, dan deskripsi tempat wisata	Handal
TT-008	Pengujian Tampil Tempat Wisata Tujuan Pada Peta	- Menekan tombol 'Play' pada antarmuka 'Home'	-	- Tampil lokasi tempat wisata tujuan pada peta berupa marker.	- Tampil lokasi tempat wisata tujuan pada peta berupa marker.	- Tampil lokasi tempat wisata tujuan pada peta berupa marker.	Handal
TT-009-01	Pengujian Tampil Soal	- Berhasil melakukan <i>check in</i> pada antarmuka 'Play'	-	- Tampil pertanyaan mengenai tempat wisata terpilih pada sebuah <i>textview</i>	- Tampil pertanyaan mengenai tempat wisata terpilih pada sebuah <i>textview</i>	- Tampil pertanyaan mengenai tempat wisata terpilih pada sebuah <i>textview</i>	Handal

TT-009-02	Pengujian Tampil Jawaban	- Berhasil melakukan <i>check in</i> pada antarmuka 'Play'	-	- Tampil jawaban-jawaban dari setiap soal yang muncul pada sebuah <i>textview</i>	- Tampil jawaban-jawaban dari setiap soal yang muncul pada sebuah <i>textview</i>	- Tampil jawaban-jawaban dari setiap soal yang muncul pada sebuah <i>textview</i>	Handal
TT-010	Pengujian Check In	- Menekan tombol 'Check In' pada antarmuka 'Play'	-	- Berhasil melakukan <i>check in</i> dan menuju antarmuka selanjutnya yaitu antarmuka kuis.	- Berhasil melakukan <i>check in</i> dan menuju antarmuka selanjutnya yaitu antarmuka kuis.	- Berhasil melakukan <i>check in</i> dan menuju antarmuka selanjutnya yaitu antarmuka kuis.	Handal
TT-011-01	Pengujian Tampil Skor	- Selesai mengerjakan kuis pada antarmuka kuis	-	- Tampil skor yang didapat sesuai jumlah benar menjawab pertanyaan pada sebuah <i>textview</i>	- Tampil skor yang didapat sesuai jumlah benar menjawab pertanyaan pada sebuah <i>textview</i>	- Tampil skor yang didapat sesuai jumlah benar menjawab pertanyaan pada sebuah <i>textview</i>	Handal
TT-011-02	Pengujian Tampil Badge	- Selesai mengerjakan kuis pada antarmuka kuis - Memenuhi syarat mendapatkan sebuah <i>badge</i>	-	- Tampil <i>badge</i> yang didapatkan berupa nama, ikon, dan deskripsi <i>badge</i>	- Tampil <i>badge</i> yang didapatkan berupa nama, ikon, dan deskripsi <i>badge</i>	- Tampil <i>badge</i> yang didapatkan berupa nama, ikon, dan deskripsi <i>badge</i>	Handal
TT-012	Pengujian	- Memasukkan	- 'admin', 'ta	- Admin masuk	- Admin masuk	- Admin masuk	Handal

	Log In Admin	<i>username</i> dan <i>password</i>	ndatanya'	ke dalam system dan halaman utama	ke dalam system dan halaman utama	ke dalam system dan halaman utama	
TT-013	Pengujian Log Out Admin	- Memilih menu 'Logout'	-	- Admin keluar dari system, halaman <i>login</i> ditampilkan	- Admin keluar dari system, halaman <i>login</i> ditampilkan	- Admin keluar dari system, halaman <i>login</i> ditampilkan	Handal
TT-014-01	Pengujian Tampil Kategori Tempat Wisata	- Memilih menu 'Category Manager'	-	- Tampil halaman kelola data kategori tempat wisata dengan table data kategori tempat wisata	- Tampil halaman kelola data kategori tempat wisata dengan table data kategori tempat wisata	- Tampil halaman kelola data kategori tempat wisata dengan table data kategori tempat wisata	Handal
TT-014-02	Pengujian Tambah Kategori Tempat Wisata	- Memilih menu 'Add New Category' - Mengisi data kategori tempat wisata - Tekan tombol 'Submit'	- 'Monument'	- Berhasil menambahkan data kategori tempat wisata - Muncul notifikasi berhasil menginput data kategori tempat wisata baru	- Berhasil menambahkan data kategori tempat wisata - Muncul notifikasi berhasil menginput data kategori tempat wisata baru	- Berhasil menambahkan data kategori tempat wisata - Muncul notifikasi berhasil menginput data kategori tempat wisata baru	Handal
TT-014-03	Pengujian Edit Kategori Tempat Wisata	- Memilih data yang ingin dirubah melalui table - Klik tombol 'Edit' pada kolom action - Mengisi data	- 'Temple'	- Berhasil merubah data kategori tempat wisata - Muncul notifikasi data kategori tempat wisata	- Berhasil merubah data kategori tempat wisata - Muncul notifikasi data kategori tempat wisata	- Berhasil merubah data kategori tempat wisata - Muncul notifikasi data kategori tempat wisata	Handal

		<p>kategori tempat wisata</p> <ul style="list-style-type: none"> <li>- Tekan tombol 'Submit'</li> </ul>		<p>berhasil diubah</p>	<p>berhasil diubah</p>	<p>berhasil diubah</p>	
TT-014-04	<p>Pengujian Hapus Kategori Tempat Wisata</p>	<ul style="list-style-type: none"> <li>- Memilih data yang akan dihapus</li> <li>- Tekan tombol 'Delete' pada kolom action</li> </ul>	-	<ul style="list-style-type: none"> <li>- Berhasil menghapus data kategori tempat wisata</li> <li>- Muncul notifikasi data kategori tempat wisata berhasil dihapus</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil menghapus data kategori tempat wisata</li> <li>- Muncul notifikasi data kategori tempat wisata berhasil dihapus</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil menghapus data kategori tempat wisata</li> <li>- Muncul notifikasi data kategori tempat wisata berhasil dihapus</li> </ul>	Handal
TT-015-01	<p>Pengujian Tampil Tempat Wisata</p>	<ul style="list-style-type: none"> <li>- Memilih menu 'Place Manager'</li> </ul>	-	<ul style="list-style-type: none"> <li>- Tampil halaman kelola data tempat wisata dengan table data tempat wisata</li> </ul>	<ul style="list-style-type: none"> <li>- Tampil halaman kelola data tempat wisata dengan table data tempat wisata</li> </ul>	<ul style="list-style-type: none"> <li>- Tampil halaman kelola data tempat wisata dengan table data tempat wisata</li> </ul>	Handal
TT-015-02	<p>Pengujian Tambah Tempat Wisata</p>	<ul style="list-style-type: none"> <li>- Memilih menu 'Add New Place'</li> <li>- Mengisi data tempat wisata</li> <li>- Tekan tombol 'Submit'</li> </ul>	<ul style="list-style-type: none"> <li>- 'Temple', 'Prambanan', 'Jalan Raya Jogja - Solo Km 16, Prambanan, Sleman, Yogyakarta 55571, Indonesia', 'Candi Prambanan adalah</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil menambahkan data tempat wisata</li> <li>- Muncul notifikasi berhasil menginput data tempat wisata baru</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil menambahkan data tempat wisata</li> <li>- Muncul notifikasi berhasil menginput data tempat wisata baru</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil menambahkan data tempat wisata</li> <li>- Muncul notifikasi berhasil menginput data tempat wisata baru</li> </ul>	Handal

			mahakarya kebudayaan Hindu dari abad ke-10.,' - 7.75205', '110.491966'				
TT-015-03	Pengujian Edit Tempat Wisata	<ul style="list-style-type: none"> <li>- Memilih data yang ingin dirubah melalui table</li> <li>- Klik tombol 'Edit' pada kolom action</li> <li>- Mengisi data tempat wisata</li> <li>- Tekan tombol 'Submit'</li> </ul>	<ul style="list-style-type: none"> <li>- 'Temple', 'Mendhut', 'Jalan Raya Jogja - Solo Km 16, Prambanan, Sleman, Yogyakarta 55571, Indonesia', 'Candi Mendhut adalah mahakarya kebudayaan Hindu dari abad ke-10.,' - 7.75205', '110.491966'</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil merubah data tempat wisata</li> <li>- Muncul notifikasi data tempat wisata berhasil diubah</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil merubah data tempat wisata</li> <li>- Muncul notifikasi data tempat wisata berhasil diubah</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil merubah data tempat wisata</li> <li>- Muncul notifikasi data tempat wisata berhasil diubah</li> </ul>	Handal
TT-015-04	Pengujian Hapus Tempat Wisata	<ul style="list-style-type: none"> <li>- Memilih data yang akan dihapus</li> <li>- Tekan tombol 'Delete' pada kolom action</li> </ul>	-	<ul style="list-style-type: none"> <li>- Berhasil menghapus data tempat wisata</li> <li>- Muncul notifikasi data tempat</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil menghapus data tempat wisata</li> <li>- Muncul notifikasi data tempat</li> </ul>	<ul style="list-style-type: none"> <li>- Berhasil menghapus data tempat wisata</li> <li>- Muncul notifikasi data tempat</li> </ul>	Handal



				wisata berhasil dihapus	wisata berhasil dihapus	wisata berhasil dihapus	
TT-016-01	Pengujian Tampil Pertanyaan	- Memilih list tempat wisata pada antarmuka 'Home'	-	- Tampil halaman kelola data pertanyaan dengan table data pertanyaan	- Tampil halaman kelola data pertanyaan dengan table data pertanyaan	- Tampil halaman kelola data pertanyaan dengan table data pertanyaan	Handal
TT-016-02	Pengujian Tambah Pertanyaan	- Memilih menu 'Add New Question' - Mengisi data pertanyaan - Tekan tombol 'Submit'	- 'Pada abad berapa Candi Prambanan dibangun?'	- Berhasil menambahkan data pertanyaan - Muncul notifikasi berhasil menginput data pertanyaan baru	- Berhasil menambahkan data pertanyaan - Muncul notifikasi berhasil menginput data pertanyaan baru	- Berhasil menambahkan data pertanyaan - Muncul notifikasi berhasil menginput data pertanyaan baru	Handal
TT-016-03	Pengujian Edit Pertanyaan	- Memilih data yang ingin dirubah melalui table - Klik tombol 'Edit' pada kolom action - Mengisi data pertanyaan - Tekan tombol 'Submit'	- 'Pada tahun berapa Candi Prambanan dibangun?'	- Berhasil merubah data tempat pertanyaan - Muncul notifikasi data pertanyaan berhasil diubah	- Berhasil merubah data tempat pertanyaan - Muncul notifikasi data pertanyaan berhasil diubah	- Berhasil merubah data tempat pertanyaan - Muncul notifikasi data pertanyaan berhasil diubah	Handal
TT-016-04	Pengujian Hapus	- Memilih data yang akan	-	- Berhasil menghapus	- Berhasil menghapus	- Berhasil menghapus	Handal

	Pertanyaan	dihapus - Tekan tombol 'Delete' pada kolom action		data pertanyaan - Muncul notifikasi data pertanyaan berhasil dihapus	data pertanyaan - Muncul notifikasi data pertanyaan berhasil dihapus	data pertanyaan - Muncul notifikasi data pertanyaan berhasil dihapus	
TT-017-01	Pengujian Tampil Jawaban	- Menekan tombol 'Show Answer' pada kolom action table pertanyaan antarmuka 'Home'	-	- Tampil halaman kelola data jawaban dengan table data jawaban	- Tampil halaman kelola data jawaban dengan table data jawaban	- Tampil halaman kelola data jawaban dengan table data jawaban	Handal
TT-017-02	Pengujian Tambah Jawaban	- Memilih menu 'Add New Answer'; - Mengisi data jawaban - Tekan tombol 'Submit'	- 'abad ke 10'	- Berhasil menambahkan data jawaban - Muncul notifikasi berhasil menginput data jawaban baru	- Berhasil menambahkan data jawaban - Muncul notifikasi berhasil menginput data jawaban baru	- Berhasil menambahkan data jawaban - Muncul notifikasi berhasil menginput data jawaban baru	Handal
TT-017-03	Pengujian Edit Jawaban	- Memilih data yang ingin dirubah melalui table - Klik tombol 'Edit' pada kolom action - Mengisi data jawaban	- 'abad ke 11'	- Berhasil merubah data jawaban - Muncul notifikasi data jawaban berhasil diubah	- Berhasil merubah data jawaban - Muncul notifikasi data jawaban berhasil diubah	- Berhasil merubah data jawaban - Muncul notifikasi data jawaban berhasil diubah	Handal

		- Tekan tombol 'Submit'					
TT-017-04	Pengujian Hapus Jawaban	- Memilih data yang akan dihapus - Tekan tombol 'Delete' pada kolom action	-	- Berhasil menghapus data jawaban - Muncul notifikasi data jawaban berhasil dihapus	- Berhasil menghapus data jawaban - Muncul notifikasi data jawaban berhasil dihapus	- Berhasil menghapus data jawaban - Muncul notifikasi data jawaban berhasil dihapus	Handal
TT-018-01	Pengujian Tampil Badge	- Memilih menu 'Badge Manager'	-	- Tampil halaman kelola data <i>badge</i> dengan table data <i>badge</i>	- Tampil halaman kelola data <i>badge</i> dengan table data <i>badge</i>	- Tampil halaman kelola data <i>badge</i> dengan table data <i>badge</i>	Handal
TT-018-02	Pengujian Tambah Badge	- Memilih menu 'Add New Badge' - Mengisi data <i>badge</i> - Tekan tombol 'Submit'	- 'Adventure', '10 check in anywhere!'	- Berhasil menambahkan data <i>badge</i> - Muncul notifikasi berhasil menginput data <i>badge</i> baru	- Berhasil menambahkan data <i>badge</i> - Muncul notifikasi berhasil menginput data <i>badge</i> baru	- Berhasil menambahkan data <i>badge</i> - Muncul notifikasi berhasil menginput data <i>badge</i> baru	Handal
TT-018-03	Pengujian Edit Badge	- Memilih data yang ingin dirubah melalui table - Klik tombol 'Edit' pada kolom action - Mengisi data <i>badge</i>	- 'Adventure', '5 check in anywhere!'	- Berhasil merubah data <i>badge</i> - Muncul notifikasi data <i>badge</i> berhasil diubah	- Berhasil merubah data <i>badge</i> - Muncul notifikasi data <i>badge</i> berhasil diubah	- Berhasil merubah data <i>badge</i> - Muncul notifikasi data <i>badge</i> berhasil diubah	Handal

		- Tekan tombol 'Submit'					
TT-018-04	Pengujian Hapus Badge	- Memilih data yang akan dihapus - Tekan tombol 'Delete' pada kolom action	-	- Berhasil menghapus data <i>badge</i> - Muncul notifikasi data <i>badge</i> berhasil dihapus	- Berhasil menghapus data <i>badge</i> - Muncul notifikasi data <i>badge</i> berhasil dihapus	- Berhasil menghapus data <i>badge</i> - Muncul notifikasi data <i>badge</i> berhasil dihapus	Handal
TT-019-01	Pengujian Tampil Foto	- Menekan tombol 'Show photo' pada kolom action table tempat wisata antarmuka 'Place Manager'	-	- Tampil halaman kelola data foto dengan table data foto	- Tampil halaman kelola data foto dengan table data foto	- Tampil halaman kelola data foto dengan table data foto	Handal
TT-019-02	Pengujian Tambah Foto	- Memilih menu 'Add New Photo' - Mengisi data foto - Tekan tombol 'Submit'	- 'Memilih gambar'	- Berhasil menambahkan data foto - Muncul notifikasi berhasil menginput data foto baru	- Berhasil menambahkan data foto - Muncul notifikasi berhasil menginput data foto baru	- Berhasil menambahkan data foto - Muncul notifikasi berhasil menginput data foto baru	Handal
TT-019-03	Pengujian Hapus Foto	- Memilih data yang akan dihapus - Tekan tombol 'Delete' pada kolom action	-	- Berhasil menghapus data foto - Muncul notifikasi data foto	- Berhasil menghapus data foto - Muncul notifikasi data foto	- Berhasil menghapus data foto - Muncul notifikasi data foto	Handal

				berhasil dihapus			
--	--	--	--	---------------------	--	--	--

#### 5.4 Hasil Pengujian Terhadap Pengguna

Pengujian sistem ini dilakukan terhadap 31 responden dengan spesifikasi :

Tabel 5.9 Tabel Spesifikasi Responden

Kategori	Data
Range Usia	20-30 tahun
Pekerjaan	Beragam, sebagian besar mahasiswa

Ada dua macam pengujian yang diberikan kepada responden yaitu pengujian antarmuka dan fasilitas situs web administrasi dan pengujian antarmuka dan fasilitas aplikasi Android. Lembar pengujian responden dapat dilihat pada bagian lampiran.

#### 5.5 Pengujian Antarmuka dan Fasilitas Situs Web Administrasi

Tabel 5.10 Tabel Pengujian Web Administrasi

NO	PERTANYAAN	PILIHAN JAWABAN			
		SS	S	KS	TS
1	Antarmuka web rapi dan enak dilihat	9	24	0	0
2	Pengelolaan data kategori, tempat wisata, pertanyaan, jawaban, badge, dan foto mudah dilakukan	13	19	1	0
3	Secara umum Anda merasa nyaman menggunakan sistem web ini	11	21	1	0

Pada pengujian ini, responden bertindak sebagai Administrator dan diminta menguji sistem menangani

pengelolaan data kategori, tempat wisata, pertanyaan, jawaban, badge, dan foto. Pengujian ini digunakan untuk membuktikan bahwa proses pengelolaan data kategori, tempat wisata, pertanyaan, jawaban, badge, dan foto dapat dilakukan dengan mudah dan terorganisir dengan baik.

Rekapitulasi data yang diperoleh dari setiap pernyataan di tabel pengujian ini adalah :

a. Pernyataan 1

9 responden Sangat Setuju dan 24 responden Setuju.

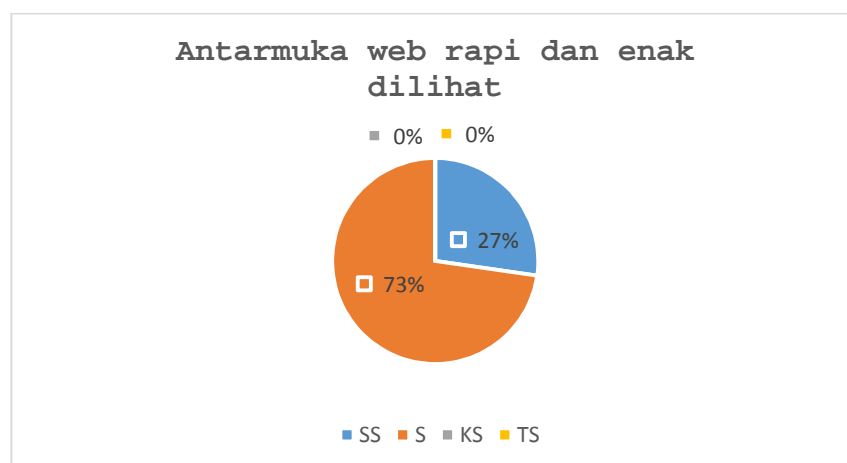
b. Pernyataan 2

13 responden Sangat Setuju, 19 responden Setuju, dan 1 responden Kurang Setuju.

c. Pernyataan 3

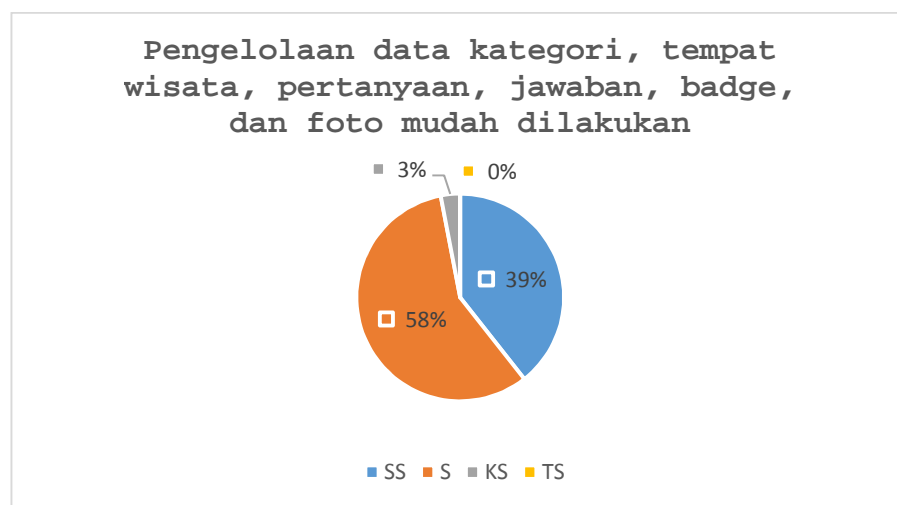
11 responden Sangat Setuju, 21 responden Setuju, dan 1 responden Kurang Setuju.

Dari rekapitulasi data di atas diperoleh hasil persentase jawaban pada grafik pie sebagai berikut:



**Gambar 5.118 Persentase Hasil Rekapitulasi Pernyataan 1 Web Administrasi**

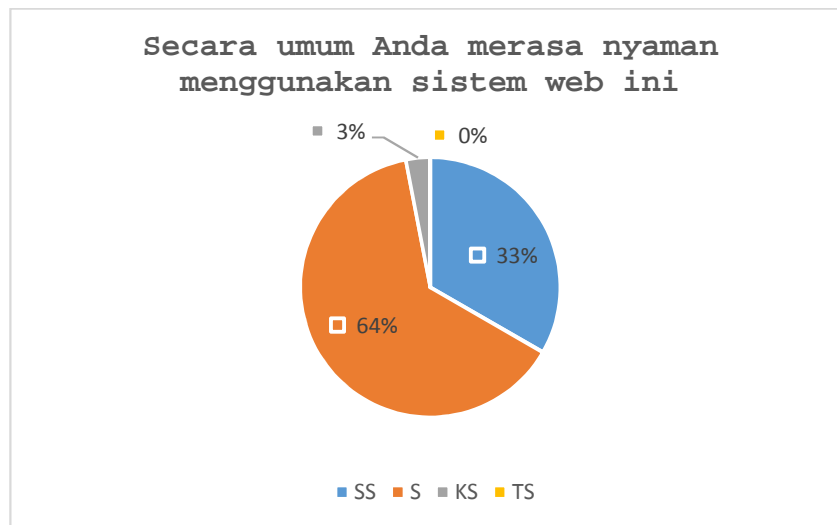
Pernyataan 1 menyebutkan bahwa antarmuka web rapi dan enak dilihat. Gambar 5.118 menunjukkan bahwa sebanyak 27% responden menyatakan sangat setuju dan 73% menyatakan setuju pada pernyataan 1. Dari hasil pengujian tersebut dapat disimpulkan bahwa antarmuka web administrasi rapi dan enak dilihat.



**Gambar 5.119 Persentase Hasil Rekapitulasi Pernyataan 2 Web Administrasi**

Pernyataan 2 menyebutkan bahwa pengelolaan data kategori, tempat wisata, pertanyaan, jawaban, badge, dan foto mudah dilakukan. Gambar 5.119 menunjukkan bahwa Sebanyak 39% responden menyatakan sangat setuju, 58% menyatakan setuju, dan hanya 3% dari responden menyatakan kurang setuju pada pernyataan 2. Dari hasil pengujian tersebut dapat disimpulkan bahwa web administrasi memberikan kemudahan dalam pengelolaan data kategori, tempat wisata, pertanyaan, jawaban, badge, dan foto.





**Gambar 5.120 Persentase Hasil Rekapitulasi Pernyataan 3 Web Administrasi**

Pernyataan 3 menyebutkan bahwa secara umum responden merasa nyaman menggunakan sistem web. Gambar 5.120 menunjukkan bahwa sebanyak 33% responden menyatakan sangat setuju, 64% menyatakan setuju, dan hanya 3% dari responden menyatakan kurang setuju pada pernyataan 3. secara umum responden nyaman dalam menggunakan web administrasi. Hasil pengujian tersebut menunjukkan bahwa secara umum pengguna nyaman menggunakan sistem web yang dibangun. Dari hasil rekapitulasi ketiga pernyataan tersebut dapat dipastikan bahwa aplikasi web administrasi memiliki kemudahan untuk mengelola data kategori, tempat wisata, pertanyaan, jawaban, badge, dan foto.

## 5.6 Pengujian Antarmuka dan Fasilitas Aplikasi Android

Tabel 5.11 Tabel Pengujian Antarmuka dan Fasilitas Aplikasi Android

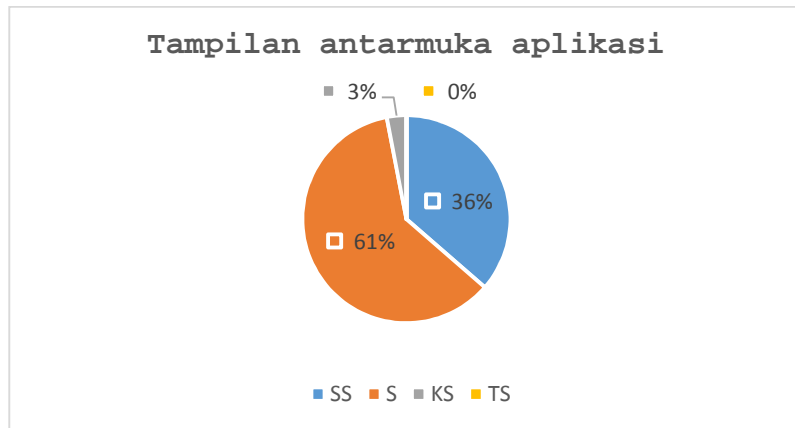
NO	PERTANYAAN	PILIHAN JAWABAN			
		SS	S	KS	TS
1	Tampilan antarmuka aplikasi	12	20	1	0
2	Adanya aplikasi Android ini mempermudah anda mengetahui lokasi tempat wisata di Yogyakarta	13	19	1	0
3	Adanya aplikasi Android ini mempermudah anda untuk mendapat informasi dari masing-masing tempat wisata di Yogyakarta	10	20	3	0
4	Adanya fitur kuis memberi suasana baru saat berkunjung ke tempat wisata	15	17	1	0
5	Adanya fitur perolehan skor & badge memberi suasana baru saat berkunjung ke tempat wisata	17	13	3	0
6	Komposisi warna antarmuka aplikasi	9	23	1	0
7	Secara umum anda merasa nyaman menggunakan sistem Android ini	15	17	1	0

Pada pengujian ini, responden bertindak sebagai pengguna dan diminta menguji aplikasi Android dengan untuk bermain dan mencari informasi tempat wisata di Yogyakarta.

Rekapitulasi data yang diperoleh dari setiap pernyataan di tabel pengujian ini adalah :

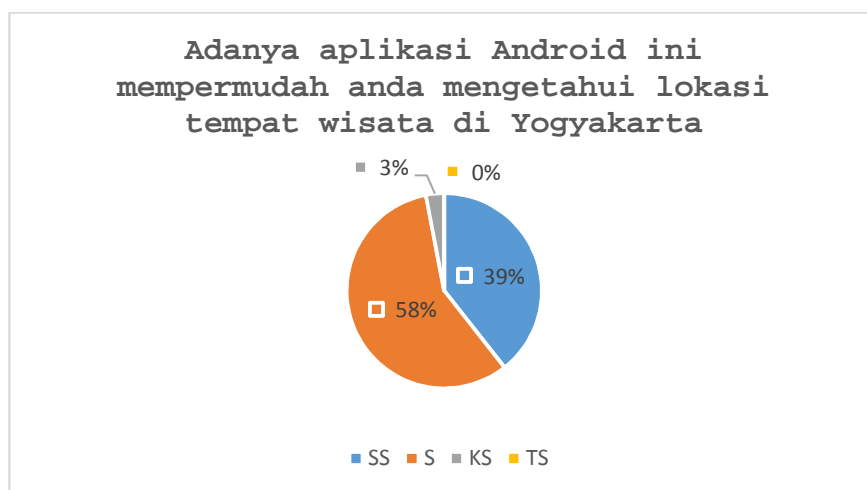
- a. Pernyataan 1  
12 responden Sangat Setuju, 20 responden Setuju, dan 1 responden Kurang Setuju.
- b. Pernyataan 2  
13 responden Sangat Setuju, 19 responden Setuju, dan 1 responden Kurang Setuju.
- c. Pernyataan 3  
10 responden Sangat Setuju, 20 responden Setuju, dan 3 responden Kurang Setuju.
- d. Pernyataan 4  
15 responden Sangat Setuju, 17 responden Setuju, dan 1 responden Kurang Setuju.
- e. Pertanyaan 5  
17 responden Sangat Setuju, 13 responden Setuju, dan 3 responden Kurang Setuju.
- f. Pertanyaan 6  
9 responden Sangat Setuju, 23 responden Setuju, dan 1 responden Kurang Setuju.
- g. Pertanyaan 7  
15 responden Sangat Setuju, 17 responden Setuju, dan 1 responden Kurang Setuju.

Dari rekapitulasi data di atas diperoleh hasil persentase jawaban pada grafik pie sebagai berikut:



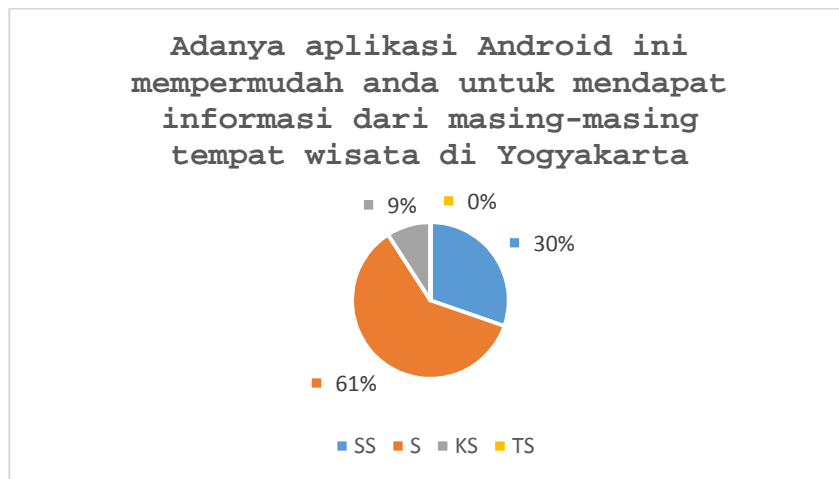
**Gambar 5.121 Persentase Hasil Rekapitulasi Pernyataan 1 Aplikasi Android**

Pernyataan 1 menyebutkan bahwa antarmuka aplikasi Android rapi dan enak dilihat. Gambar 5.121 menunjukkan bahwa Sebanyak 36% responden menyatakan sangat setuju, 61% responden menyatakan setuju, dan hanya 3% responden menyatakan kurang setuju pada pernyataan 1. Dari hasil pengujian pada pernyataan 1 dapat disimpulkan bahwa antarmuka aplikasi Android rapi dan enak dilihat.



**Gambar 5.122 Persentase Hasil Rekapitulasi Pernyataan 2 Aplikasi Android**

Pernyataan 2 menyebutkan bahwa adanya aplikasi Android ini dapat mempermudah pengguna mengetahui lokasi tempat wisata di Yogyakarta. Gambar 5.112 menunjukkan bahwa sebanyak 39% responden menyatakan sangat setuju, 58% menyatakan setuju, dan hanya 3% dari responden menyatakan kurang setuju pada pernyataan 2. Hasil pengujian terhadap pernyataan 2 tersebut menunjukkan bahwa aplikasi Android dapat mempermudah pengguna mengetahui lokasi tempat wisata di Yogyakarta.



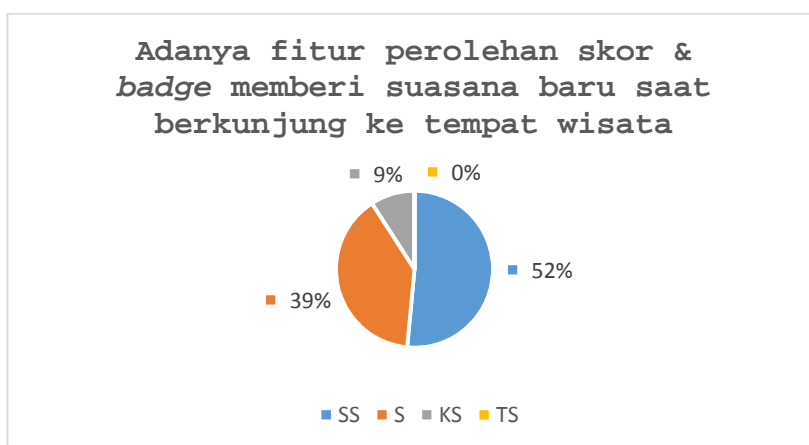
**Gambar 5.123 Persentase Hasil Rekapitulasi Pernyataan 3 Aplikasi Android**

Pernyataan 3 menyebutkan bahwa adanya sistem Android ini mempermudah responden untuk mendapatkan informasi dari masing-masing tempat wisata di Yogyakarta. Gambar 5.123 menunjukkan bahwa sebanyak 30% responden menyatakan sangat setuju, 61% menyatakan setuju, dan 9% dari responden menyatakan kurang setuju pada pernyataan 3. Hasil pengujian pada pernyataan 3 menunjukkan bahwa aplikasi Android mempermudah pengguna untuk mendapatkan informasi dari masing-masing tempat wisata di Yogyakarta.



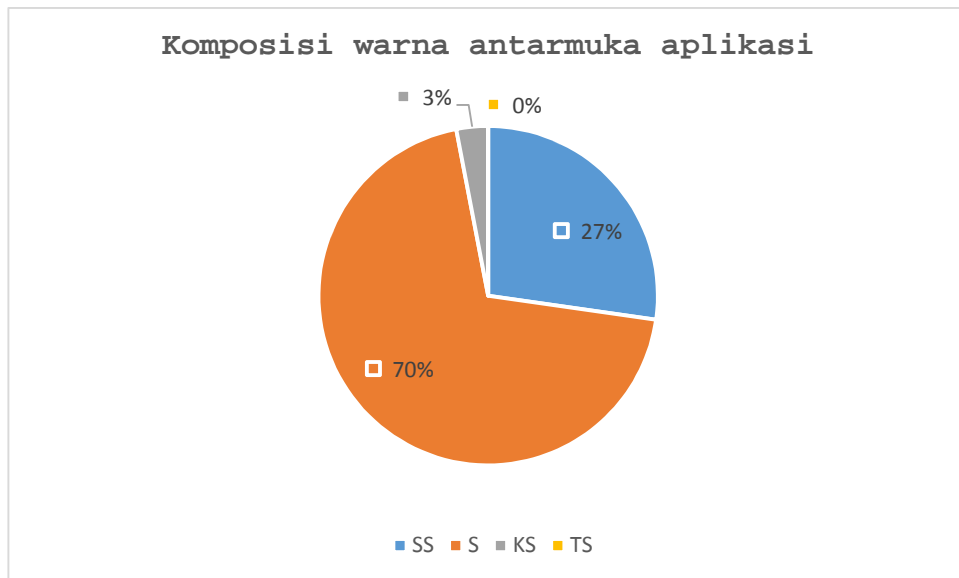
**Gambar 5.124 Persentase Hasil Rekapitulasi Pernyataan 4 Aplikasi Android**

Pernyataan 4 menyebutkan bahwa adanya fitur kuis memberi suasana baru saat berkunjung ke tempat wisata. Gambar 5.124 menunjukkan bahwa sebanyak 45% responden menyatakan sangat setuju, 52% menyatakan setuju, dan 3% dari responden menyatakan kurang setuju pada pernyataan 4. Hasil pengujian pada pernyataan 4 menunjukkan bahwa adanya fitur kuis memberi suasana baru saat berkunjung ke tempat wisata.



**Gambar 5.125 Persentase Hasil Rekapitulasi Pernyataan 5 Aplikasi Android**

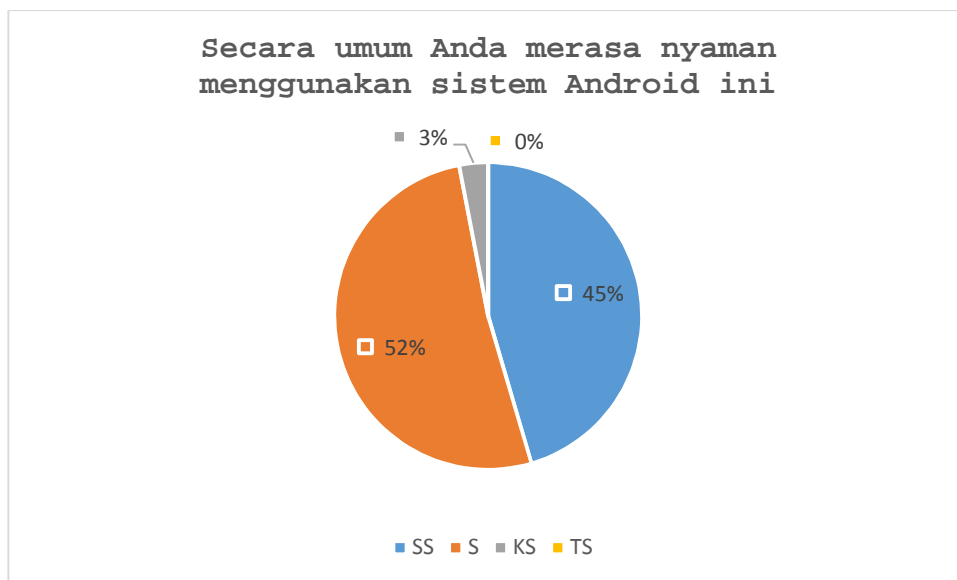
Pernyataan 5 menyebutkan bahwa adanya fitur perolehan skor dan *badge* memberi suasana baru saat berkunjung ke tempat wisata. Gambar 5.125 menunjukkan bahwa sebanyak 52% responden menyatakan sangat setuju, 39% menyatakan setuju, dan 9% dari responden menyatakan kurang setuju pada pernyataan 5. Hasil pengujian terhadap pernyataan 5 menunjukkan bahwa adanya fitur perolehan skor dan *badge* memberi suasana baru saat berkunjung ke tempat wisata.



**Gambar 5.126 Persentase Hasil Rekapitulasi Pernyataan 6 Aplikasi Android**

Pernyataan 6 menyebutkan bahwa komposisi warna antarmuka aplikasi sudah cocok. Gambar 5.126 menunjukkan bahwa sebanyak 27% responden menyatakan sangat setuju, 70% responden menyatakan setuju, dan 3%

dari responden menyatakan kurang setuju pada pernyataan 6. Hasil pengujian terhadap pernyataan 6 menunjukkan bahwa komposisi warna antarmuka aplikasi sudah cocok.



**Gambar 5.127 Persentase Hasil Rekapitulasi Pernyataan 7 Aplikasi Android**

Pernyataan 7 menyebutkan bahwa secara umum responden merasa nyaman menggunakan aplikasi Android yang dibangun. Gambar 5.127 menunjukkan bahwa Sebanyak 45% responden menyatakan sangat setuju, 52% responden menyatakan setuju, dan 3% dari responden menyatakan kurang setuju pada pernyataan 7. Hasil pengujian terhadap pernyataan 7 menunjukkan bahwa aplikasi Android yang dibangun nyaman digunakan oleh pengguna. Dari hasil rekapitulasi ketujuh pernyataan tersebut dapat disimpulkan bahwa aplikasi Android pada perangkat *mobile* mampu mempermudah pengguna untuk mencari , mendapatkan informasi tempat wisata di Yogyakarta dengan suasana baru.



## 5.7 Kritik dan Saran Pengguna

Setelah kedua pengujian dilakukan, responden diminta untuk menulis kritik dan saran mengenai sistem yang telah dikembangkan. Berikut ini adalah kritik dan saran yang diberikan oleh responden :

Tabel 5.12 Tabel Kritik dan Saran Terhadap Sistem

Kritik	Saran
Antarmuka aplikasi Android masih banyak ruang kosong yang tidak digunakan.	Antarmuka aplikasi Android lebih dikembangkan dalam disain dan variasi konten.
Fitur kuis kadang membosankan	Memberi level pada fitur kuis, dan tidak hanya pertanyaan <i>plain text</i>
Antarmuka situs web kurang menarik.	Memperindah antarmuka aplikasi web.
Tampilan data pada table tidak rapi dan tidak menarik untuk dilihat	Memperindah tampilan table dengan memberikan warna dan ikon-ikon.

### 5.7.1 Analisis Kelebihan dan Kekurangan Sistem

Dari implementasi dan pengujian, baik oleh penulis maupun responden, dapat dianalisis kelebihan dan kekurangan dari aplikasi TANDATANYA yang mencakup :

#### 5.7.1.1 Kelebihan

1. Aplikasi web administrasi memiliki kemudahan dalam memproses data kategori, tempat wisata, pertanyaan, jawaban, badge, dan foto .

2. Aplikasi Android pada perangkat *mobile* dapat mempermudah pengguna mendapatkan informasi tempat wisata di Yogyakarta.
3. Terdapat fitur kuis dan fitur perolehan skor dan *badge* sehingga menambah suasana baru pada pengguna saat berwisata di kota Yogyakarta.

#### **5.7.2 Kekurangan**

1. Terlalu banyak data yang harus di akses melalui internet.
2. Pembagian proses akses data dari internet tidak teratur.

Demikian pembahasan mengenai implementasi dan pengujian perangkat lunak. Pada bab selanjutnya, yaitu bab penutup, akan diberikan kesimpulan dan saran yang didapatkan selama pembuatan Tugas Akhir.

## **BAB VI**

### **PENUTUP**

Pada bab penutup ini akan diberikan kesimpulan dan saran yang didapatkan selama pembuatan Tugas Akhir.

#### **6.1 Kesimpulan**

Berdasarkan dari pembahasan-pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini :

1. TANDATANYA-Android berhasil dibangun untuk menampilkan posisi lokasi-lokasi pariwisata pada perangkat *smartphone* dengan sistem operasi Android.
2. TANDATANYA-Android berhasil dibangun untuk dapat menentukan lokasi pariwisata tujuan secara acak kepada pengguna pada perangkat *smarphone* dengan sistem operasi Android.
3. TANDATANYA-Android berhasil dibangun untuk dapat menampilkan beberapa pertanyaan seputar tempat pariwisata yang terpilih pada perangkat *smarphone* dengan sistem operasi Android.
4. TANDATANYA-Android berhasil dibangun untuk dapat menampilkan perolehan angka yang didapatkan dari menjawab pertanyaan-pertanyaan yang diberikan pada perangkat *smartphone* dengan sistem operasi Android

#### **6.2 Saran**

Dari proses analisis, perancangan, implementasi hingga pengujian sistem pada pembuatan Tugas Akhir,

didapatkan beberapa saran untuk pengembangan lebih lanjut perangkat lunak TANDATANYA, yaitu :

1. Menambahkan fungsionalitas "*Help*" untuk membantu pengguna menggunakan aplikasi.
2. Menambahkan fitur level pada kuis.

## DAFTAR PUSTAKA

- Ableson, W. Frank, Sen, Robi, King, Chris, 2011, Android in Action Second Edition, Manning Publications Co., United States of America.
- Bharkuus Louise, Dey Anind. 2003. Location-Based Services for Mobile Telephony: a study of users' privacy concerns. Proceedings of the INTERACT 2003, 9, TH, IFIP TC13 International Conference on Human-Computer Interaction.
- Cheng Zhang Shi. 2009. Development and Research of Application Based on Google Android. Computer Knowledge and Technology 2009-28.
- Diewald Stefan, Moller Andreas, Roalter Luis, Kranz Matthias. 2012. Drive Assist - A V2X - Based Driver Assistance System for Android. Luella University of Technology, Department of Computer Science, Electrical and Space Engineering, Sweden.
- Elian, Alqod. 2012. Layanan Informasi Kereta Api Menggunakan GPS, Google Maps, dan Android. Institut Teknologi Sepuluh Nopember Surabaya
- Goldberg, DW., 2008, A Geocoding Best Practices Guide, University of Southern California, Los Angeles.
- Griffiths, A. (2010). *CodeIgniter 1.7 Professional Development*. Birmingham: Packt Publishing Ltd.
- Harriehausen Bettina. 2012. Quiz Lounge Game-based Learnin on Mobile Devices. Computer Science Department, Univ. of Applied Sciences, Darmstadt, Germany.
- Hashimi, Sayed, Komatineni, Satya, MacLean, Dave, 2009, Pro Android 2, Apress, United States of America.

- Hengming Fang, Jia Chen, Bin Xu. 2013. The Interaction Mechanism Based on JSON for Android Database Application. Information Technology Journal, ISSN: 1812-5638.
- Ingale Jitendra. 2013. Android Based WS Security and MVC Based UI Representation of Data. International Journal of Computer Science, Engineering and Information Technology, Vol 3, No. 1, February 2013.
- Jiang Fan. 2010. How To Display The Data From Database by List View on Android. Intelligent Systems and Applications (ISA), 2010 2nd International Workshop.
- Kristanto, Harianto. 2004 Konsep dan Perancangan Database. Penerbit Andi Yogyakarta
- Omprakash Khanna SamratVivekanand, Patel Pritesh. 2012. Application of Google API and KML to Draw Path from Source to Destination on Android Phone. International Journal of Advances in Engineering & Technology, E-ISSN: 0976-3945.
- Kushwaha A, Kushwaha V. 2012. Location Based Services in Android. International Journal of Advances in Engineering & Technology, ISSN: 2231-1963.
- Openhandsetalliance, 2012, OHA FAQ, [http://www.openhandsetalliance.com/oha\\_faq.html](http://www.openhandsetalliance.com/oha_faq.html), diakses tanggal 12 Januari 2012.
- Reddy Siddartha Sreenivasa. 2011. Trip Tracker Application on Android. Master of Science in Computer Science, San Diego State University.
- Singhal Manav, Shukla, Anupam. 2012. Implementation of Location based Services in Android using GPS and Web Services. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012.
- Steiniger, Stefan, Moritz Neun, Edwardes, Alistair, 2006, Foundations of Location Based Services, University of Zurich, Zurich.

- Suryatiningsih, 2010. Aplikasi Administrasi Aktivitas Kemahasiswaan Politeknik Telkom. Politeknik Telkom Bandung.
- Whipple. 2009. A public safety application of GPS-enabled smartphones and the android operating system. *Systems, Man and Cybernetics*, 2009. SMC 2009.
- Wijesuria M.U.E, Mendis S.U, Bandara B.E.S, Mahawattge K.P, Walgampaya N, Silva De D. 2013. Interactive Mobile Based Tour Guide. SAITM Research Symposium on Engginering Advancements 2013, Malabe, Sri Lanka.
- Xianhua Shu, Zhenjun Du, Rong Chen, Research on Mobile Location Service Design Based on Android, *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on* , vol., no., pp.1,4, 24-26 Sept. 2009.
- Yu Chien-Chih, Chang ,Hsiao-ping. 2009. Personalized Location-Based Recommendation Services for Tour Planning in Mobile Tourism Applications. *Lecture Notes in Computer Science Volume 5692*, 2009, pp 38-49.
- Yuan, Yuan. 2012. Design and Implementation of Positioning and Navigation System Based on Android. School of Computer Science and Engineering Beihang University, Beijing, China.
- Zhijian Long. 2011. Research and Implementation of Navigation Application which Based on the Google Map API on the Android Platform. Academy of Computer Science, National University of Defense Technology Changsha 410073.
- Zigkolis Christos, Kompatsiaris Yiannis, Vakali Athena. 2009. Infromation Analysis in Mobile Social Networks for Added-Value Services.

**LAMPIRAN**



**SKPL**

**SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK**


**PEMBANGUNAN APLIKASI MOBILE BERBASIS  
LOKASI UNTUK PERMAINAN PENGENALAN OBYEK  
WISATA DI YOGYAKARTA**

Dipersiapkan oleh:

**PLACYDUS ANOM BHAYU MAHARSI / 5869**

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

**Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<b>SKPL-TT</b>		1/45
	Fakultas Teknologi Industri			

## DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	PABM							
Diperik sa oleh	YSP,DEV							
Disetuj ui oleh	YSP,DEV							

## DAFTAR HALAMAN PERUBAHAN

Halaman	Revisi	Halaman	Revisi

## DAFTAR ISI

DAFTAR PERUBAHAN .....	2
DAFTAR HALAMAN PERUBAHAN .....	3
DAFTAR ISI.....	4
DAFTAR GAMBAR .....	6
1 Pendahuluan.....	7
1.1 Tujuan .....	7
1.2 Lingkup Masalah .....	7
1.3 Definisi, Akronim dan Singkatan .....	8
1.4 Referensi .....	8
1.5 Deskripsi umum .....	8
2 Deskripsi Kebutuhan.....	9
2.1 Perspektif produk .....	9
2.2 Fungsi Produk .....	11
2.3 Karakteristik User .....	17
2.4 Batasan-batasan .....	18
2.5 Asumsi dan Ketergantungan .....	18
3. Kebutuhan khusus .....	18
3.1. Kebutuhan antarmuka eksternal .....	18
3.1.1. Antarmuka pemakai.....	19
3.1.2. Antarmuka perangkat keras.....	19
3.1.3. Antarmuka perangkat lunak.....	19
3.1.4. Antarmuka Komunikasi.....	20
3.2. Kebutuhan fungsionalitas Perangkat Lunak .....	21
3.2.1. Use Case Diagram.....	21
4. Spesifikasi Rinci Kebutuhan.....	22
4.1. Spesifikasi Kebutuhan Fungsionalitas .....	22
4.1.1 Use Case Spesification : Log In.....	22
4.1.2 Use Case Spesification : Log out.....	23
4.1.3 Use Case Spesification : Tampil Profil Pengguna.....	24
4.1.4 Use Case Spesification : Tampil Sejarah Pengguna.....	25
4.1.5 Use Case Spesification : Tampil Teman Pengguna.....	26
4.1.6 Use Case Spesification : Tampil Tempat Wisata pada Peta	

4.1.7	Use Case Spesification : Tampil Profil Tempat Wisata.	28
4.1.8	Use Case Spesification : Tampil Tempat Wisata Tujuan pada Peta .....	29
4.1.9	Use Case Spesification : Answer Quiz.....	31
4.1.10	Use Case Spesification : Check In.....	32
4.1.11	Use Case Spesification : Tampil Skor.....	33
4.1.12	Use Case Spesification : Log In Admin.....	34
4.1.13	Use Case Spesification : Log Out Admin.....	35
4.1.14	Use Case Spesification : Mengelola Data Kategori Tempat Wisata	36
4.1.15	Use Case Spesification : Mengelola Data Tempat Wisata	38
4.1.16	Use Case Spesification : Mengelola Data Pertanyaan...	39
4.1.17	Use Case Spesification : Mengelola Data Jawaban.....	41
4.1.18	Use Case Spesification : Mengelola Data Badge.....	42
4.1.19	Use Case Spesification : Mengelola Data Foto.....	44
5.	Entity Relationship Diagram (ERD) .....	46

## DAFTAR GAMBAR

Gambar 1. Arsitektur Perangkat lunak TT.....	10
Gambar 2. Use Case Diagram (User).....	21
Gambar 3. Use Case Diagram (Administrator).....	22
Gambar 4. Entity Relationship Diagram.....	46

## 1 Pendahuluan

### 1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak TANDATANYA untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan user) performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-TT ini juga mendefinisikan batasan perancangan perangkat lunak.

### 1.2 Lingkup Masalah

Perangkat Lunak TANDATANYA dikembangkan dengan tujuan untuk :

1. Mampu membangun aplikasi TANDATANYA yang dapat menampilkan posisi lokasi-lokasi pariwisata pada perangkat *smartphone* dengan sistem operasi Android.
2. Mampu membangun aplikasi TANDATANYA yang dapat menentukan lokasi pariwisata tujuan secara acak kepada pengguna pada perangkat *smartphone* dengan sistem operasi Android.
3. Mampu membangun aplikasi TANDATANYA yang dapat menampilkan beberapa pertanyaan seputar tempat pariwisata yang terpilih pada perangkat *smartphone* dengan sistem operasi Android.

Program Studi Teknik Informatika	SKPL – TT	7/46
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		

4. Mampu membangun aplikasi TANDATANYA yang dapat menampilkan perolehan angka yang didapatkan dari menjawab pertanyaan-pertanyaan yang diberikan pada perangkat *smartphone* dengan sistem operasi Android.

### 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-TT-XXX	Kode yang merepresentasikan kebutuhan pada TANDATANYA dimana XXX merupakan nomor fungsi produk.
TANDATANYA	Database yang digunakan oleh aplikasi TANDATANYA
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

### 1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Maharsi, Placydus Anom Bhayu. Spesifikasi Kebutuhan Perangkat Lunak SPP, Universitas Atma Jaya Yogyakarta 2012.

### 1.5 Deskripsi umum

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL,

Program Studi Teknik Informatika	SKPL – TT	8/46
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		



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 TANDATANYA yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik user, batasan dalam user perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak TANDATANYA tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak TANDATANYA yang akan dikembangkan.

## **2 Deskripsi Kebutuhan**

### **2.1 Perspektif produk**

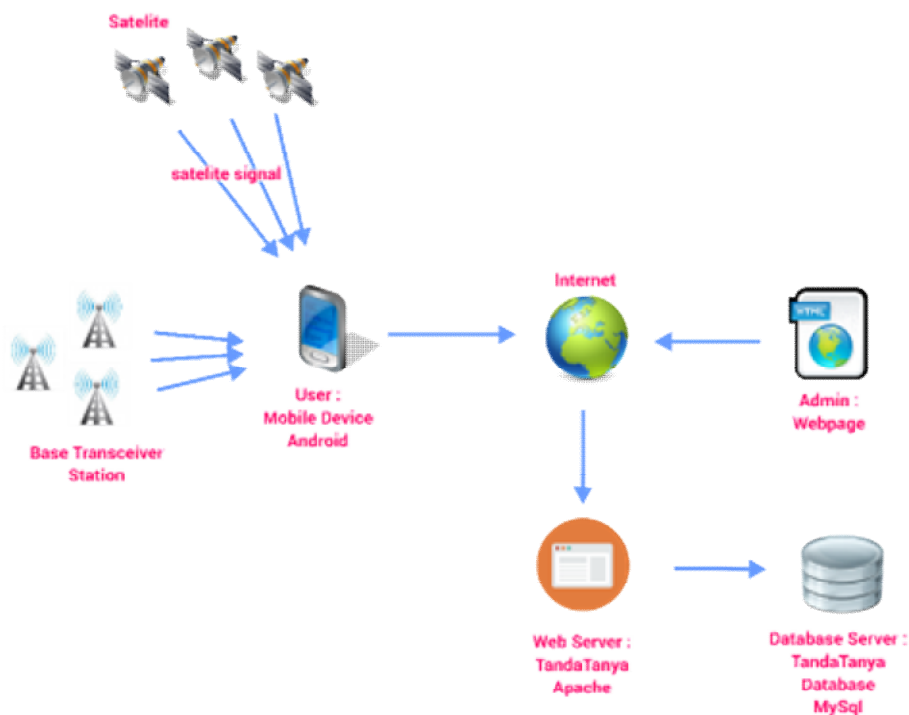
TANDATANYA merupakan perangkat lunak yang dikembangkan untuk membantu penyebaran informasi tempat-tempat pariwisata yang ada di kota Yogyakarta dengan memanfaatkan Layanan Berbasis Lokasi. Selain itu sistem ini diharapkan dapat menjadi sarana untuk menambah wawasan mengenai sejarah lokal kota Yogyakarta yang interaktif sehingga dapat menarik User untuk mencari tahu tentang sejarah lokal.

Perangkat lunak ini dikembangkan untuk ditanamkan pada sisi *client* di lingkungan sistem operasi Android OS. Perangkat lunak ini dikembangkan menggunakan bahasa pemrograman Java dan menggunakan lingkungan pemrograman Eclipse Juno 4.20.

Program Studi Teknik Informatika	SKPL – TT	9/46
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		

Pada sisi server ditanamkan sistem *webservice* pada sistem operasi Windows 7 / Windows 8 dengan *web server* Apache dan dibuat menggunakan bahasa pemrograman PHP dengan menggunakan framework CodeIgniter2.0.3. Sedangkan untuk lingkungan pemrogramannya menggunakan Sublime Text 2.

Pengguna akan berinteraksi dengan sistem melalui antarmuka *Graphical User Interface*(GUI). Pada sistem ini, seperti terlihat pada gambar 4.1, arsitektur teknologi yang digunakan perangkat lunak berupa *client-server*, di mana semua data disimpan di *server*. *Client* dapat mengakses data yang ada di *server* tersebut secara *online* dengan membuka aplikasi TANDATANYA yang merujuk pada sistem yang telah ditanamkan pada *web server*.



**Gambar 1. Arsitektur Perangkat lunak TT**

## 2.2 Fungsi Produk

Fungsi produk perangkat lunak TANDATANYA adalah sebagai berikut :

### 1. Fungsi Log In (**SKPL-TT-001**)

Fungsi ini digunakan oleh pengguna yang telah terdaftar oleh Facebook untuk masuk ke dalam aplikasi.

### 2. Fungsi Log Out (**SKPL-TT-002**)

Fungsi ini digunakan oleh pengguna yang telah terdaftar oleh Facebook untuk keluar dari aplikasi.

### 3. Fungsi Tampil Profil Pengguna (**SKPL-TT-003**)

Fungsi ini digunakan untuk menampilkan profil pengguna yang berupa *profile picture*, nama, dan total skor pengguna.

### 4. Fungsi Tampil Sejarah Pengguna (**SKPL-TT-004**)

Fungsi ini digunakan pengguna untuk menampilkan sejarah dari pengguna.

### 5. Fungsi Tampil Teman Pengguna (**SKPL-TT-005**)

Fungsi ini digunakan oleh pengguna untuk menampilkan teman (Facebook) pengguna yang juga menggunakan aplikasi TANDATANYA.

### 6. Fungsi Tampil Tempat Wisata pada Peta (**SKPL-TT-006**)

Fungsi ini digunakan oleh pengguna untuk menampilkan tempat - tempat wisata yang ada di dalam database ke sebuah peta berupa marker.

7. Fungsi Tampil Profil Tempat Wisata (**SKPL-TT-007**)

Fungsi ini digunakan untuk menampilkan profil tempat wisata yang dipilih berupa foto, nama, alamat, dan deskripsi tempat wisata.

8. Fungsi Tampil Tempat Wisata Tujuan pada Peta (**SKPL-TT-008**)

Fungsi ini digunakan oleh pengguna untuk menampilkan lokasi tempat tujuan wisata secara acak ke dalam sebuah peta dengan sebuah marker.

9. Fungsi Answer Quiz (**SKPL-TT-009**)

Fungsi ini digunakan oleh pengguna untuk menjawab pertanyaan seputar tempat wisata yang dipilih. Fungsi ini mencakup :

a. Fungsi Tampil Soal (**SKPL-TT-009-01**)

Fungsi ini digunakan untuk menampilkan soal satu persatu berdasarkan lokasi yang dipilih.

b. Fungsi Tampil Jawaban (**SKPL-TT-009-02**)

Fungsi ini digunakan untuk menampilkan jawaban setiap soal yang sedang tertampil.

10. Fungsi Check In (**SKPL-TT-010**)

Fungsi ini digunakan oleh pengguna untuk melakukan check in di lokasi tempat wisata.

11. Fungsi Tampil Skor (**SKPL-TT-011**)

Fungsi ini digunakan oleh pengguna untuk menampilkan peroleh skor ketika sudah selesai mengerjakan quiz. Fungsi ini mencakup:

a. Fungsi Tampil Skor (**SKPL-TT-011-01**)

Program Studi Teknik Informatika	SKPL – TT	12/46
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		

Fungsi yang digunakan untuk menampilkan skor yang diperoleh pengguna.

b. Fungsi Tampil Badge (**SKPL-TT-011-02**)

Fungsi ini digunakan untuk menampilkan Badge yang diperoleh pengguna.

12. Fungsi Log In Admin (**SKPL-TT-012**)

Fungsi ini digunakan oleh Administrator untuk masuk ke dalam web khusus Administrator.

13. Fungsi Log Out Admin (**SKPL-TT-013**)

Fungsi ini digunakan oleh Administrator untuk keluar dari web khusus Administrator.

14. Fungsi Pengelolaan Data Kategori Tempat Wisata (**SKPL-TT-014**)

Fungsi ini digunakan oleh Administrator untuk mengelola data Kategori Tempat Wisata. Fungsi ini mencakup :

a. Fungsi Tampil Kategori Tempat Wisata (**SKPL-TT-014-01**)

Fungsi ini digunakan untuk menampilkan semua data kategori tempat wisata yang ada di dalam database TANDATANYA.

b. Fungsi Tambah Kategori Tempat Wisata (**SKPL-TT-014-02**)

Fungsi ini digunakan untuk menambahkan data kategori tempat wisata ke dalam database TANDATANYA.

c. Fungsi Edit Kategori Tempat Wisata (**SKPL-TT-014-03**)

Fungsi ini digunakan untuk merubah data kategori tempat wisata yang ada di dalam database TANDATANYA.

d. Fungsi Hapus Kategori Tempat Wisata (**SKPL-TT-014-04**)

Fungsi ini digunakan untuk menghapus data kategori tempat wisata yang ada di dalam database TANDATANYA.

15. Fungsi Pengelolaan Data Tempat Wisata (**SKPL-TT-015**)

Fungsi ini digunakan oleh Administrator untuk melakukan pengelolaan data tempat wisata yang ada di dalam database TANDATANYA. Fungsi ini mencakup :

a. Fungsi Tampil Tempat Wisata (**SKPL-TT-015-01**)

Fungsi ini digunakan untuk menampilkan semua data tempat wisata yang ada di dalam database TANDATANYA.

b. Fungsi Tambah Tempat Wisata (**SKPL-TT-015-02**)

Fungsi ini digunakan untuk menambahkan tempat wisata baru ke dalam database TANDATANYA.

c. Fungsi Edit Tempat Wisata (**SKPL-TT-015-03**)

Fungsi ini digunakan untuk merubah data tempat wisata yang ada di dalam database TANDATANYA.

d. Fungsi Hapus Tempat Wisata (**SKPL-TT-015-04**)

Fungsi ini digunakan untuk menghapus data tempat wisata yang ada di dalam database TANDATANYA.

16. Fungsi Pengelolaan Data Pertanyaan (**SKPL-TT-016**)

Fungsi ini digunakan oleh Administrator untuk mengelola data pertanyaan yang ada di dalam database TANDATANYA. Fungsi ini mencakup :

a. Fungsi Tampil Pertanyaan (**SKPL-TT-016-01**)

Fungsi ini digunakan untuk menampilkan semua data pertanyaan yang ada dalam database TANDATANYA berdasarkan tempat wisata.

b. Fungsi Tambah Pertanyaan (**SKPL-TT-016-02**)

Fungsi ini digunakan untuk menambahkan pertanyaan baru ke dalam database TANDATANYA berdasarkan tempat wisata.

c. Fungsi Edit Pertanyaan (**SKPL-TT-016-03**)

Fungsi ini digunakan untuk merubah data pertanyaan yang ada di dalam database TANDATANYA berdasarkan tempat wisata.

d. Fungsi Hapus Pertanyaan (**SKPL-TT-016-04**)

Fungsi ini digunakan untuk menghapus data pertanyaan yang ada di dalam database TANDATANYA berdasarkan tempat wisata.

17. Fungsi Pengelolaan Data Jawaban (**SKPL-TT-017**)

Fungsi ini digunakan oleh Administrator untuk mengelola data jawaban yang ada di dalam database TANDATANYA berdasarkan pertanyaan. Fungsi ini mencakup :

Program Studi Teknik Informatika	SKPL – TT	15/46
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. Fungsi Tampil Jawaban (**SKPL-TT-017-01**)

Fungsi ini digunakan untuk menampilkan semua data jawaban yang ada di dalam database TANDATANYA berdasarkan pertanyaan.

b. Fungsi Tambah Jawaban (**SKPL-TT-017-02**)

Fungsi ini digunakan untuk menambahkan data jawaban baru ke dalam database TANDATANYA berdasarkan pertanyaan.

c. Fungsi Edit Jawaban (**SKPL-TT-017-03**)

Fungsi ini digunakan untuk merubah data jawaban yang ada di dalam database TANDATANYA berdasarkan pertanyaan.

d. Fungsi Hapus Jawaban (**SKPL-TT-017-04**)

Fungsi ini digunakan untuk menghapus data jawaban yang ada di dalam database TANDATANYA berdasarkan pertanyaan.

18. Fungsi Pengelolaan Data Badge (**SKPL-TT-018**)

Fungsi ini digunakan administrator untuk mengelola data badge yang ada di dalam database TANDATANYA. Fungsi ini mencakup :

a. Fungsi Tampil Badge (**SKPL-TT-018-01**)

Fungsi ini digunakan untuk menampilkan semua data badge yang ada di dalam database TANDATANYA.

b. Fungsi Tambah Badge (**SKPL-TT-018-02**)

Fungsi ini digunakan untuk menambahkan badge baru ke dalam database TANDATANYA.



c. Fungsi Edit Badge (**SKPL-TT-018-03**)

Fungsi ini digunakan untuk merubah data badge yang ada di dalam database TANDATANYA.

d. Fungsi Hapus Badge (**SKPL-TT-018-04**)

Fungsi ini digunakan untuk menghapus data badge yang ada di dalam database TANDATANYA.

19. Fungsi Pengelolaan Data Foto (**SKPL-TT-019**)

Fungsi ini digunakan administrator untuk mengelola data foto yang ada di dalam database TANDATANYA. Fungsi ini mencakup :

a. Fungsi Tampil Foto (**SKPL-TT-019-01**)

Fungsi ini digunakan untuk menampilkan semua data foto yang ada di dalam database TANDATANYA.

b. Fungsi Tambah Foto (**SKPL-TT-019-02**)

Fungsi ini digunakan untuk menambahkan foto baru ke dalam database TANDATANYA.

c. Fungsi Hapus Foto (**SKPL-TT-019-03**)

Fungsi ini digunakan untuk menghapus data foto yang ada di dalam database TANDATANYA.

### 2.3 Karakteristik User

Karakteristik dari user perangkat lunak TANDATANYA adalah sebagai berikut :

1. Memahami pengoperasian telepon genggam dengan sistem operasi Android.

Program Studi Teknik Informatika	SKPL – TT	17/46
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. Mengerti tentang internet dan web.

#### **2.4 Batasan-batasan**

Batasan-batasan dalam pengembangan perangkat lunak TANDATANYA tersebut adalah :

1. Kebijaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak TANDATANYA.

2. Keterbatasan perangkat keras

Dapat diketahui kemudian sesudah sistem ini berjalan (sesuai dengan kebutuhan).

#### **2.5 Asumsi dan Ketergantungan**

Sistem ini dapat dijalankan pada perangkat mobile (Handphone) yang menggunakan sistem operasi Android dengan versi minimal 4.0.1 (Ice Cream Sandwich) yang memiliki koneksi internet (untuk aplikasi mobile) dan perangkat desktop dengan sistem operasi Windows/Linux/MacOS menggunakan *web browser*.

### **3. Kebutuhan khusus**

#### **3.1. Kebutuhan antarmuka eksternal**

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

### 3.1.1. Antarmuka pemakai

User berinteraksi dengan antarmuka yang ditampilkan dalam bentuk layout aplikasi mobile, sementara Administrator berinteraksi dengan antarmuka web.

### 3.1.2. Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak TANDATANYA adalah:

#### 1. Perangkat keras.

Spesifikasi minimum handphone yang digunakan:

- OS Android API 15+
- Mendukung modul GPS
- Mendukung modul OpenGL ES 2.0

Spesifikasi minimum komputer yang dibutuhkan:

- Processor 1.4 GHz
- RAM 1 GB

### 3.1.3. Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak TANDATANYA adalah sebagai berikut :

1. Nama : Android  
Sumber : Google

Sebagai sistem operasi untuk *mobile platform* oleh pengguna.

2. Nama : Internet Explorer, Mozilla Firefox, Google Chrome, Opera, Safari, dll.

Sebagai antarmuka pengelolaan data oleh admin.

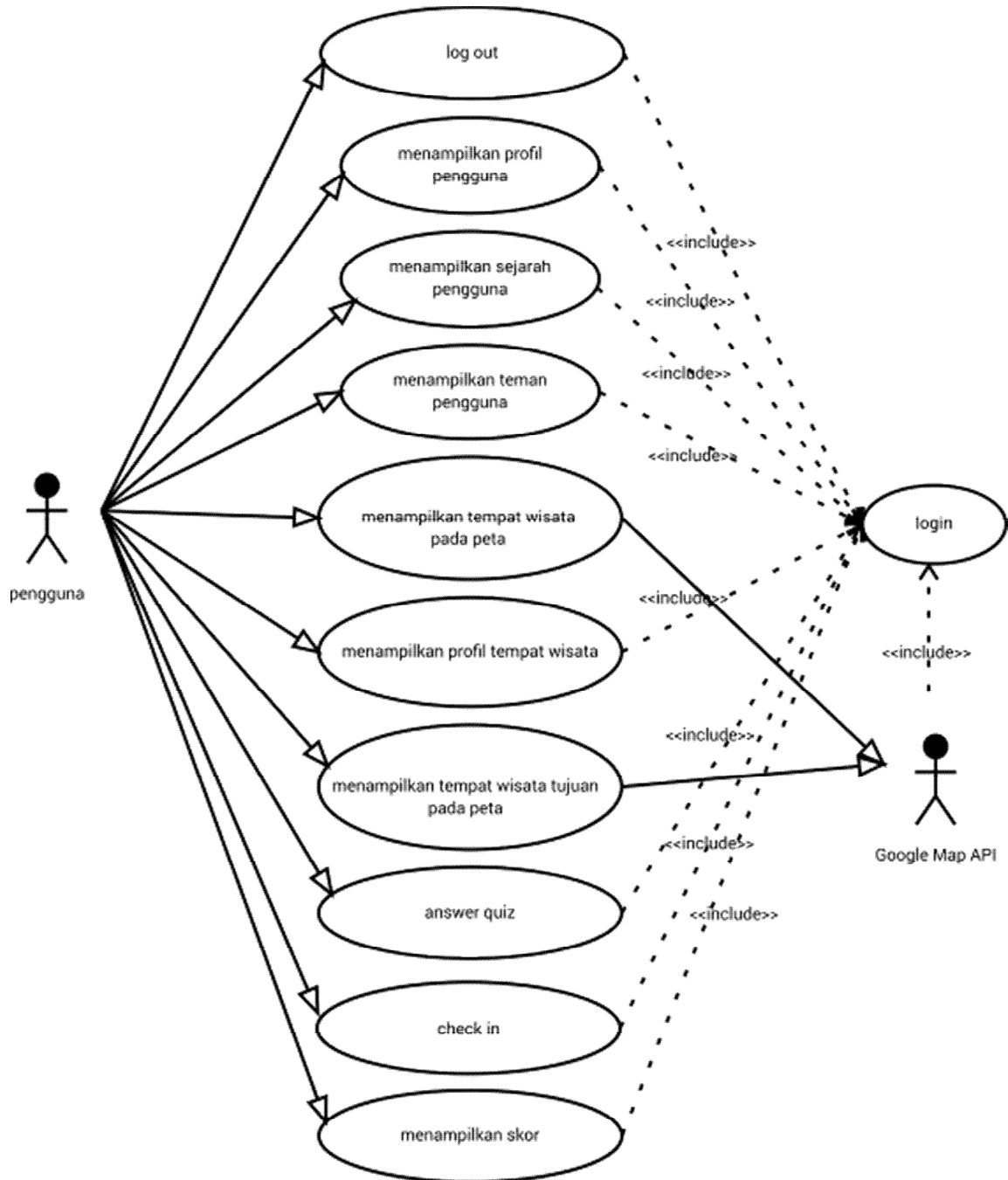
Program Studi Teknik Informatika	SKPL – TT	19/46
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

#### **3.1.4. Antarmuka Komunikasi**

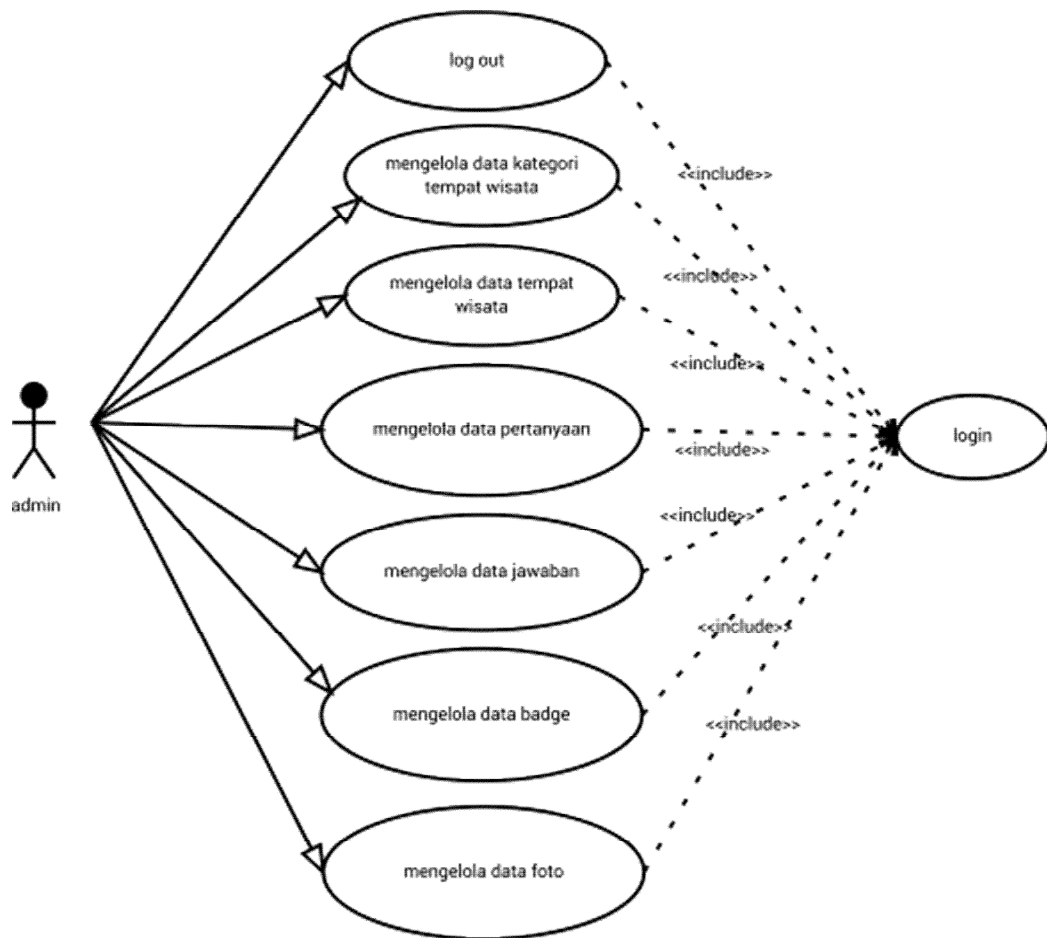
Antarmuka komunikasi perangkat lunak TandaTanya terhubung secara client-server dalam lingkup jaringan Internet.

### 3.2. Kebutuhan fungsionalitas Perangkat Lunak

#### 3.2.1. Use Case Diagram



Gambar 2. Use Case Diagram (User)



Gambar 3. Use Case Diagram (Administrator)

## 4. Spesifikasi Rinci Kebutuhan

### 4.1. Spesifikasi Kebutuhan Fungsionalitas

#### 4.1.1 Use Case Specification : Log In

##### 1. Brief Description

Use Case ini digunakan oleh aktor untuk masuk ke dalam aplikasi menggunakan akun Facebook.

##### 2. Primary Actor

User

##### 3. Supporting Actor

None

Program Studi Teknik Informatika	SKPL – TT	22/46
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		

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

1. Use Case ini dimulai ketika aktor memilih untuk masuk ke dalam aplikasi.
2. Sistem menampilkan antar muka log in.
3. Aktor memasukkan data log in.
4. Sistem melakukan validasi data.
5. Sistem menyimpan data aktor pada aplikasi.
6. Use Case selesai.

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

*None*

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

E-1. Form yang diisikan Aktor tidak lengkap

1. Sistem menampilkan pesan kesalahan bahwa form tidak valid.
2. Kembali ke Basic Flow langkah ke-3.

E-2. Username/Password tidak sesuai

1. Sistem menampilkan pesan kesalahan bahwa username/password tidak sesuai.
2. Kembali ke Basic Flow langkah ke-3.

#### **7. PreConditions**

*None*

#### **8. PostConditions**

Aktor berhasil masuk ke dalam aplikasi

#### **4.1.2 Use Case Spesification : Log out**

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

Use Case ini digunakan oleh aktor untuk keluar dari aplikasi.

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

User

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

*None*

Program Studi Teknik Informatika	SKPL – TT	23/46
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		

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

1. Use Case ini dimulai ketika aktor memilih untuk keluar aplikasi.
2. Aktor mengindikasikan ingin keluar dari aplikasi.
3. Sistem menghapus data aktor dari aplikasi.
4. Use Case selesai

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

*None*

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

*None*

#### **7. PreConditions**

Aktor sudah login

#### **8. PostConditions**

1. Aktor keluar dari aplikasi.
2. Sistem menampilkan antarmuka login.

### **4.1.3 Use Case Spesification : Tampil Profil Pengguna**

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

Use Case ini digunakan oleh aktor untuk menampilkan profil pengguna.

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

User

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk menampilkan profil pengguna.
2. Aplikasi mengambil data aktor dari database.
3. Aplikasi menampilkan profil pengguna yaitu foto profil, nama, *badges*, dan total skor pengguna.
4. Use Case ini selesai.

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

Program Studi Teknik Informatika	SKPL – TT	24/46
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		



None

## 6. Error Flow

None

## 7. PreConditions

1. Aplikasi berjalan dengan baik dan lancar.
2. Aktor memiliki koneksi internet sehingga dapat terhubung dengan sistem.
3. Sistem pada server berjalan dengan baik dan lancar.
4. Aktor memiliki koneksi GPS.

## 8. PostConditions

Profil pengguna telah tertampil.

### 4.1.4 Use Case Spesification : Tampil Sejarah Pengguna

#### 1. Brief Description

Merupakan fungsi yang digunakan oleh pengguna untuk menampilkan sejarah pengguna dalam bentuk sebuah list yang berisi foto profil tempat wisata, nama, skor, dan tanggal main.

#### 2. Primary Actor

User

#### 3. Supporting Actor

None

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk menampilkan sejarah pengguna.
2. Aplikasi mengambil data sejarah aktor dari database.
3. Aplikasi menampilkan sejarah pengguna berupa list tempat wisata yang pernah dikunjungi yang terdiri dari foto profil tempat wisata, nama, skor, dan

Program Studi Teknik Informatika	SKPL – TT	25/46
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		

tanggal pengguna bermain pada tempat wisata tersebut.

4. Use Case ini selesai.

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

*None*

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

*None*

#### **7. PreConditions**

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

#### **8. PostConditions**

Sejarah pengguna telah tertampil.

### **4.1.5 Use Case Spesification : Tampil Teman Pengguna**

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

Merupakan fungsi yang digunakan oleh pengguna untuk menampilkan teman pengguna dari Facebook yang juga menggunakan aplikasi TandaTanya.

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

1. User

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

*None*

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

1. Use Case ini dimulai ketika aktor memilih untuk menampilkan teman pengguna.
2. Aplikasi mengambil data teman pengguna dari database Facebook dan database TANDATANYA.

Program Studi Teknik Informatika	SKPL – TT	26/46
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		

3. Aplikasi menampilkan teman pengguna berupa list teman pengguna yang terdiri dari foto profil, nama, dan total skor.

4. Use Case selesai.

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

*None*

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

*None*

#### **7. PreConditions**

1. Aplikasi berjalan dengan baik dan lancar.

2. Aktor memiliki koneksi internet sehingga dapat terhubung dengan sistem.

3. Aplikasi pada server berjalan dengan baik dan lancar.

#### **8. PostConditions**

Data teman pengguna telah tertampil.

### **4.1.6 Use Case Spesification : Tampil Tempat Wisata pada Peta**

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

Merupakan fungsi yang digunakan oleh pengguna untuk menampilkan lokasi tempat-tempat wisata di Yogyakarta berupa marker pada peta.

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

User

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

Google Maps API

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

1. Use case ini dimulai ketika aktor memilih untuk menampilkan lokasi tempat-tempat wisata di Yogyakarta berupa marker pada peta.

Program Studi Teknik Informatika	SKPL – TT	27/46
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. Aplikasi mengambil semua data tempat wisata dari *database*.
3. Aplikasi meminta hak akses untuk menampilkan peta dan marker ke layar *mobile device* dari aktor *Google Maps API*.
4. Aplikasi menampilkan lokasi tempat-tempat wisata di Yogyakarta berupa marker pada peta.
5. Use case selesai.

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

*None*

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

*None*

#### **7. PreConditions**

1. Aplikasi berjalan dengan baik.
2. Aktor memiliki koneksi internet sehingga dapat terhubung dengan system.
3. Sistem pada server berjalan dengan baik dan lancar.

#### **8. PostConditions**

Lokasi tempat-tempat wisata di Yogyakarta telah tertampil.

### **4.1.7 Use Case Spesification : Tampil Profil Tempat Wisata**

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

Use Case ini digunakan oleh aktor untuk menampilkan profil tempat wisata yang dipilih pengguna berupa galeri foto dan deskripsi singkat dengan syarat pengguna telah menyelesaikan kuis di tempat wisata tersebut.

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

User

Program Studi Teknik Informatika	SKPL – TT	28/46
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		

### 3. Supporting Actor

None

### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk menampilkan profil tempat wisata yang dipilih pengguna.
2. Aplikasi mengambil data tempat wisata sesuai yang dipilih aktor dari *database*.
3. Aplikasi menampilkan profil tempat wisata ke layar *mobile device*.
4. Use case selesai.

### 5. Alternative Flow

None

### 6. Error Flow

None

### 7. PreConditions

4. Aplikasi berjalan dengan baik.
5. Aktor memiliki koneksi internet sehingga dapat terhubung dengan system.
6. Sistem pada server berjalan dengan baik dan lancar.

### 8. PostConditions

Profil tempat wisata berhasil tertampil.

#### 4.1.8 Use Case Spesification : Tampil Tempat Wisata Tujuan pada Peta

##### 1. Brief Description

Use Case ini digunakan oleh aktor untuk menampilkan lokasi tujuan pengguna secara acak berdasarkan lokasi tempat-tempat wisata di Yogyakarta pada peta.

##### 2. Primary Actor

Program Studi Teknik Informatika	SKPL – TT	29/46
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		

User

### 3. Supporting Actor

Google Maps API

### 4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk menampilkan lokasi tujuan pengguna secara acak berdasarkan lokasi tempat-tempat wisata di Yogyakarta yang belum pernah dikunjungi.
2. Aplikasi mengambil data tempat wisata dari *database*.
3. Aplikasi memilih salah satu lokasi dari semua lokasi yang didapat.
4. Aplikasi menyimpan data lokasi yang terpilih dan user yang sedang aktif sebagai *checkpoint*.
5. Aplikasi meminta hak akses untuk menampilkan peta dan marker ke layar *mobile device* dari aktor *Google Maps API*
6. Aplikasi menampilkan lokasi tujuan aktor secara acak berdasarkan lokasi tempat-tempat wisata di Yogyakarta pada peta berupa sebuah marker.
7. Use case selesai.

### 5. Alternative Flow

*None.*

### 6. Error Flow

*None.*

### 7. PreConditions

1. Aplikasi berjalan dengan baik dan lancar.
2. Aktor memiliki koneksi internet sehingga terhubung dengan sistem.
3. Sistem pada server berjalan dengan baik dan lancar.

Program Studi Teknik Informatika	SKPL – TT	30/46
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		

4. Aktor memiliki koneksi GPS.

#### **8. PostConditions**

Lokasi tujuan aktor telah tertampil berupa sebuah marker pada peta.

#### **4.1.9 Use Case Specification : Answer Quiz**

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

Use Case ini digunakan oleh aktor untuk menampilkan kuis dan mengerjakan kuis tersebut.

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

1. User.

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

None.

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

1. Use Case ini dimulai ketika aktor memilih untuk menampilkan kuis.
2. Aplikasi mengambil data pertanyaan dan jawaban dari database TANDATANYA sesuai tempat wisata.
3. Aplikasi menampilkan kuis satu persatu berdasarkan tempat wisata yang sedang dikunjungi pengguna.
4. Aktor mengerjakan kuis yang ditampilkan aplikasi.
5. Aktor berhasil menyelesaikan kuis.
6. Use Case ini selesai.

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

*None.*

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

*None.*

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

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

Program Studi Teknik Informatika	SKPL – TT	31/46
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		

3. Sistem pada server berjalan dengan baik dan lancar.

## 8. PostConditions

Aktor telah berhasil mengerjakan kuis.

### 4.1.10 Use Case Spesification : Check In

#### 1. Brief Description

Use Case ini digunakan oleh aktor untuk *check in* pada sebuah lokasi wisata di Yogyakarta.

#### 2. Primary Actor

1. User.

#### 3. Supporting Actor

None.

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk *check in* pada sebuah lokasi wisata di Yogyakarta.

2. Aplikasi memeriksa akumulasi jarak lokasi pengguna dengan lokasi tempat wisata tujuan.

E-1 Jarak antara lokasi pengguna dan lokasi tempat wisata tujuan tidak dalam kondisi benar untuk melakukan *check in*.

3. Aplikasi menampilkan pesan berhasil *check in* ke layar *mobile device* pengguna.

4. Use Case ini selesai.

#### 5. Alternative Flow

None.

#### 6. Error Flow

E-1 Jarak antara lokasi pengguna dan lokasi tempat wisata tujuan tidak dalam kondisi benar untuk melakukan *check in*.

1. Aplikasi menampilkan peringatan gagal melakukan *check in* dikarenakan jarak antara lokasi



pengguna dan lokasi tempat wisata tidak dalam kondisi benar.

2. Kembali ke Basic Flow ke-1

#### **7. PreConditions**

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

#### **8. PostConditions**

Aktor telah berhasil melakukan *check in*.

#### **4.1.11 Use Case Spesification : Tampil Skor**

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

Use Case ini digunakan oleh aktor untuk menampilkan skor pengguna setelah pengguna menyelesaikan kuis.

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

User.

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

None.

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

1. Use Case ini dimulai ketika aktor telah menyelesaikan kuis.
2. Aplikasi mengakumulasi skor dari setiap jawaban yang telah dipilih oleh aktor.
3. Aplikasi menampilkan skor yang diperoleh aktor dan menyimpannya ke dalam database TANDATANYA.
4. Use Case ini selesai.

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

A-1 Aktor memenuhi syarat perolahan *badge* dalam akumulasi skor.

Program Studi Teknik Informatika	SKPL – TT	33/46
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. Aplikasi menampilkan *badge* yang diperoleh user ke layar.

2. Berlanjut ke Basic Flow langkah 4.

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

*None.*

#### **7. PreConditions**

1. Aplikasi berjalan dengan baik dan lancar.

2. Aktor memiliki koneksi internet sehingga dapat terhubung dengan sistem.

3. Sistem pada server berjalan dengan baik dan lancar.

#### **8. PostConditions**

Aplikasi berhasil menampilkan skor yang diperoleh aktor.

### **4.1.12 Use Case Spesification : Log In Admin**

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

Use Case ini digunakan oleh aktor untuk masuk ke dalam system berbasis web.

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

Administrator.

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

*None.*

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

1. Use Case ini dimulai ketika aktor memilih untuk login di system web.

2. Sistem menampilkan antarmuka login.

3. Aktor mengisi data login.

4. Sistem melakukan validasi data login yang diisikan oleh aktor.

5. Sistem memeriksa data login di dalam basis data.

6. Aktor masuk ke dalam system.

Program Studi Teknik Informatika	SKPL – TT	34/46
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		

7. Use Case selesai.

**5. Alternative Flow**

*None.*

**6. Error Flow**

E-1 Data yang diinputkan aktor tidak valid

1. Sistem menampilkan pesan kesalahan bahwa data tidak valid.
2. Kembali ke Basic Flow langkah ke 2.

**7. PreConditions**

1. Aktor memiliki koneksi internet sehingga dapat terhubung dengan sistem.

**8. PostConditions**

Aktor masuk ke dalam sistem.

**4.1.13 Use Case Spesification : Log Out Admin**

**1. Brief Description**

Use Case ini digunakan oleh aktor untuk keluar dari sistem.

**2. Primary Actor**

Administrator

**3. Supporting Actor**

*None*

**4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk keluar sistem.
2. Aktor mengindikasikan ingin keluar dari sistem.
3. Sistem mengeluarkan aktor dari sistem.
4. Use Case selesai.

**5. Alternative Flow**

*None*

**6. Error Flow**

*None*

Program Studi Teknik Informatika	SKPL – TT	35/46
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		

## 7. PreConditions

Aktor sudah login

## 8. PostConditions

Aktor keluar dari sistem

### 4.1.14 Use Case Specification : Mengelola Data Kategori Tempat Wisata

#### 1. Brief Description

Use Case ini memungkinkan aktor untuk mengelola data kategori tempat wisata, termasuk di dalamnya menampilkan, menambah, merubah dan menghapus data kategori tempat wisata.

#### 2. Primary Actor

Administrator

#### 3. Supporting Actor

None

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data kategori tempat wisata.
2. Sistem menampilkan semua data kategori tempat wisata.
3. Use Case selesai.

#### 5. Alternative Flow

A-1 Aktor memilih untuk menambah data kategori tempat wisata

1. Sistem menampilkan form data kategori tempat wisata.
2. Aktor mengisi form data kategori tempat wisata.
3. Sistem melakukan validasi data kategori tempat wisata yang diisikan actor.
4. Sistem menyimpan data kategori tempat wisata baru ke dalam database TANDATANYA.

5. Berlanjut ke Basic Flow langkah ke-3.

A-2 Aktor memilih untuk merubah data kategori tempat wisata.

1. Aktor memilih data kategori tempat wisata yang akan dirubah.
2. Sistem menampilkan form data kategori tempat wisata beserta data yang dipilih oleh aktor.
3. Aktor merubah data kategori tempat wisata yang dipilih.
4. Sistem melakukan validasi data yang diisikan oleh actor.
5. Sistem menyimpan data yang diisikan oleh actor ke dalam database TANDATANYA.

A-3 Aktor memilih untuk menghapus data kategori tempat wisata.

1. Aktor memilih data kategori tempat wisata yang akan dihapus.
2. Sistem menghapus data kategori tempat wisata dari database TEMPATWISATA.

## **6. Error Flow**

E-1 Data yang diinputkan oleh actor tidak valid

1. Sistem menampilkan pesan kesalahan bahwa data tidak valid.
2. Kembali ke Basic Flow langkah ke-3.

## **7. PreConditions**

Aktor sudah login

## **8. PostConditions**

Aktor selesai mengelola data kategori tempat wisata.

#### 4.1.15 Use Case Spesification : Mengelola Data Tempat Wisata

##### 1. Brief Description

Use Case ini memungkinkan aktor untuk mengelola data tempat wisata, termasuk di dalamnya menampilkan, menambah, merubah dan menghapus data tempat wisata.

##### 2. Primary Actor

Administrator

##### 3. Supporting Actor

None

##### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data tempat wisata.
2. Sistem menampilkan semua data tempat wisata.
3. Use Case selesai.

##### 5. Alternative Flow

A-1 Aktor memilih untuk menambah data tempat wisata

1. Sistem menampilkan form data tempat wisata.
2. Aktor mengisi form data tempat wisata.
3. Sistem melakukan validasi data tempat wisata yang diisikan actor.
4. Sistem menyimpan data tempat wisata baru ke dalam database TANDATANYA.
5. Berlanjut ke Basic Flow langkah ke-3.

A-2 Aktor memilih untuk merubah data tempat wisata.

1. Aktor memilih data tempat wisata yang akan dirubah.
2. Sistem menampilkan form data tempat wisata beserta data yang dipilih oleh aktor.

Program Studi Teknik Informatika	SKPL – TT	38/46
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		

3. Aktor merubah data tempat wisata yang dipilih.
4. Sistem melakukan validasi data yang diisikan oleh actor.
5. Sistem menyimpan data yang diisikan oleh actor ke dalam database TANDATANYA.

A-3 Aktor memilih untuk menghapus data kategori tempat wisata.

1. Aktor memilih data tempat wisata yang akan dihapus.
2. Sistem menghapus data tempat wisata dari database TANDATANYA.

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

E-1 Data yang diinputkan oleh actor tidak valid

1. Sistem menampilkan pesan kesalahan bahwa data tidak valid.
2. Kembali ke Basic Flow langkah ke-3.

#### **7. PreConditions**

Aktor sudah login

#### **8. PostConditions**

Aktor selesai mengelola data tempat wisata.

### **4.1.16 Use Case Spesification : Mengelola Data Pertanyaan**

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

Use Case ini memungkinkan aktor untuk mengelola data pertanyaan, termasuk di dalamnya menampilkan, menambah, merubah dan menghapus data pertanyaan.

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

Administrator

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

None

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

Program Studi Teknik Informatika	SKPL – TT	39/46
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		

4. Use Case ini dimulai ketika aktor memilih untuk mengelola data pertanyaan.
5. Sistem menampilkan semua data pertanyaan.
6. Use Case selesai.

## 5. Alternative Flow

A-1 Aktor memilih untuk menambah data pertanyaan

1. Sistem menampilkan form data pertanyaan.
2. Aktor mengisi form data pertanyaan.
3. Sistem melakukan validasi data pertanyaan yang diisikan actor.
4. Sistem menyimpan data pertanyaan baru ke dalam database TANDATANYA.
5. Berlanjut ke Basic Flow langkah ke-3.

A-2 Aktor memilih untuk merubah data pertanyaan.

1. Aktor memilih data pertanyaan yang akan dirubah.
2. Sistem menampilkan form data pertanyaan beserta data yang dipilih oleh aktor.
3. Aktor merubah data pertanyaan yang dipilih.
4. Sistem melakukan validasi data yang diisikan oleh actor.
5. Sistem menyimpan data yang diisikan oleh actor ke dalam database TANDATANYA.

A-3 Aktor memilih untuk menghapus data kategori tempat wisata.

1. Aktor memilih data pertanyaan yang akan dihapus.
2. Sistem menghapus data pertanyaan dari database TANDATANYA.

## 6. Error Flow

E-1 Data yang diinputkan oleh actor tidak valid

Program Studi Teknik Informatika	SKPL – TT	40/46
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 menampilkan pesan kesalahan bahwa data tidak valid.
2. Kembali ke Basic Flow langkah ke-3.

**7. PreConditions**

Aktor sudah login

**8. PostConditions**

Aktor selesai mengelola data pertanyaan.

**4.1.17 Use Case Spesification : Mengelola Data Jawaban**

**1. Brief Description**

Use Case ini memungkinkan aktor untuk mengelola data jawaban, termasuk di dalamnya menampilkan, menambah, merubah dan menghapus data jawaban.

**2. Primary Actor**

Administrator

**3. Supporting Actor**

None

**4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data jawaban.
2. Sistem menampilkan semua data jawaban.
3. Use Case selesai.

**5. Alternative Flow**

A-1 Aktor memilih untuk menambah data jawaban

1. Sistem menampilkan form data jawaban.
2. Aktor mengisi form data jawaban.
3. Sistem melakukan validasi data jawaban yang diisikan actor.
4. Sistem menyimpan data jawaban baru ke dalam database TANDATANYA.
5. Berlanjut ke Basic Flow langkah ke-3.

A-2 Aktor memilih untuk merubah data jawaban.

Program Studi Teknik Informatika	SKPL – TT	41/46
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. Aktor memilih data jawaban yang akan dirubah.
2. Sistem menampilkan form data jawaban beserta data yang dipilih oleh aktor.
3. Aktor merubah data jawaban yang dipilih.
4. Sistem melakukan validasi data yang diisikan oleh actor.
5. Sistem menyimpan data yang diisikan oleh actor ke dalam database TANDATANYA.

A-3 Aktor memilih untuk menghapus data kategori tempat wisata.

1. Aktor memilih data jawaban yang akan dihapus.
2. Sistem menghapus data jawaban dari database TANDATANYA.

## **6. Error Flow**

E-1 Data yang diinputkan oleh actor tidak valid

1. Sistem menampilkan pesan kesalahan bahwa data tidak valid.
2. Kembali ke Basic Flow langkah ke-3.

## **7. PreConditions**

Aktor sudah login

## **8. PostConditions**

Aktor selesai mengelola data jawaban.

### **4.1.18 Use Case Spesification : Mengelola Data Badge**

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

Use Case ini memungkinkan aktor untuk mengelola data badge, termasuk di dalamnya menampilkan, menambah, merubah dan menghapus data badge.

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

Administrator

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

None

Program Studi Teknik Informatika	SKPL – TT	42/46
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		

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

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data badge.
2. Sistem menampilkan semua data badge.
3. Use Case selesai.

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

A-1 Aktor memilih untuk menambah data badge

1. Sistem menampilkan form data badge.
2. Aktor mengisi form data badge.
3. Sistem melakukan validasi data badge yang diisikan actor.
4. Sistem menyimpan data badge baru ke dalam database TANDATANYA.
5. Berlanjut ke Basic Flow langkah ke-3.

A-2 Aktor memilih untuk merubah data badge.

1. Aktor memilih data badge yang akan dirubah.
2. Sistem menampilkan form data badge beserta data yang dipilih oleh aktor.
3. Aktor merubah data badge yang dipilih.
4. Sistem melakukan validasi data yang diisikan oleh actor.
5. Sistem menyimpan data yang diisikan oleh actor ke dalam database TANDATANYA.

A-3 Aktor memilih untuk menghapus data kategori tempat wisata.

1. Aktor memilih data badge yang akan dihapus.
2. Sistem menghapus data badge dari database TANDATANYA.

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

E-1 Data yang diinputkan oleh actor tidak valid

Program Studi Teknik Informatika	SKPL – TT	43/46
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 menampilkan pesan kesalahan bahwa data tidak valid.
2. Kembali ke Basic Flow langkah ke-3.

**7. PreConditions**

Aktor sudah login

**8. PostConditions**

Aktor selesai mengelola data badge.

**4.1.19 Use Case Spesification : Mengelola Data Foto**

**1. Brief Description**

Use Case ini memungkinkan aktor untuk mengelola data foto, termasuk di dalamnya menampilkan, menambah dan menghapus data foto.

**2. Primary Actor**

Administrator

**3. Supporting Actor**

None

**4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data foto.
2. Sistem menampilkan semua data foto.
3. Use Case selesai.

**5. Alternative Flow**

A-1 Aktor memilih untuk menambah data foto

1. Sistem menampilkan form data foto.
2. Aktor memilih file foto yang ingin ditambahkan.
3. Sistem menyimpan data foto baru ke dalam database TANDATANYA.
4. Berlanjut ke Basic Flow langkah ke-3.

A-2 Aktor memilih untuk menghapus data kategori tempat wisata.

1. Aktor memilih data foto yang akan dihapus.

Program Studi Teknik Informatika	SKPL – TT	44/46
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. Sistem menghapus data foto dari database

TANDATANYA.

**6. Error Flow**

*None.*

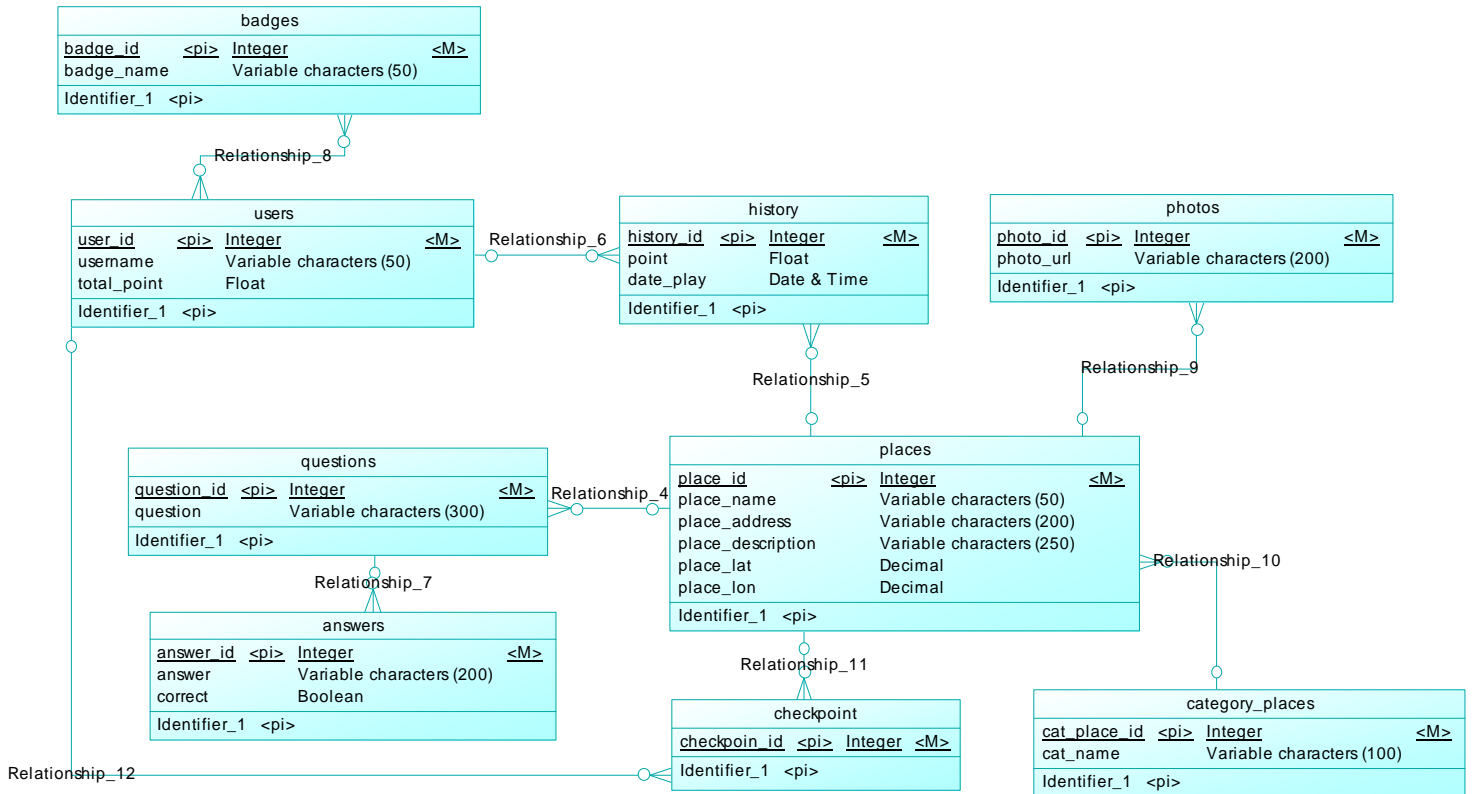
**7. PreConditions**

Aktor sudah login

**8. PostConditions**

Aktor selesai mengelola data foto.

## 5. Entity Relationship Diagram (ERD)



Gambar 4. Entity Relationship Diagram

**DPPL**

**DESKRIPSI PERANCANGAN PERANGKAT LUNAK**


**PEMBANGUNAN APLIKASI MOBILE BERBASIS  
LOKASI UNTUK PERMAINAN PENGENALAN OBYEK  
WISATA DI YOGYAKARTA**

**Dipersiapkan oleh:**

**Placydus Anom Bhayu Maharsi / 090705869**

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

**Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<b><i>DPPL-TandtaTanya</i></b>		1/67
		Revisi		

## Daftar Perubahan

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

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	PABM							
Diperik sa oleh	YSP, DEV							
Disetuj ui oleh	YSP, DEV							



## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

## Daftar Isi

Daftar Perubahan.....	2
Daftar Halaman Perubahan.....	3
Daftar Isi.....	4
Daftar Gambar.....	9
1 Pendahuluan .....	12
1.1 Tujuan .....	12
1.2 Ruang Lingkup .....	12
1.3 Definisi dan Akronim.....	12
1.4 Referensi .....	13
2 Perancangan Sistem .....	14
2.1 Perancangan Arsitektur Aplikasi TandaTanya untuk User (Android).....	14
2.2 Perancangan Arsitektur Aplikasi TandaTanya untuk Admin (Web)	15
2.3 Perancangan Rinci .....	16
2.3.1 Sequence Diagram : Aplikasi TandaTanya untuk User (Android)	16
2.3.1.1 Login.....	16
2.3.1.2 Logout.....	16
2.3.1.3 Menampilkan Profil Pengguna.....	17
2.3.1.4 Menampilkan Sejarah Pengguna.....	17
2.3.1.5 Menampilkan Teman Pengguna.....	18
2.3.1.6 Menampilkan Tempat Wisata pada Peta.....	18
2.3.1.7 Menampilkan Profil Tempat Wisata.....	19
2.3.1.8 Menampilkan Tempat Wisata Tujuan pada Peta.....	19
2.3.1.9 Answer Quiz.....	20
2.3.1.10 Check In.....	21
2.3.1.11 Menampilkan Skor.....	21
2.3.2 Sequence Diagram : Aplikasi TandaTanya untuk Administrator (Web) .....	22
2.3.2.1 Login Admin.....	22
2.3.2.2 Logout Admin.....	22

2.3.2.3	Pengelolaan Data Kategori Tempat Wisata.....	23
2.3.2.3.1	Tampil Kategori Tempat Wisata.....	23
2.3.2.3.2	Tambah Kategori Tempat Wisata.....	23
2.3.2.3.3	Edit Kategori Tempat Wisata.....	24
2.3.2.3.4	Hapus Kategori Tempat Wisata.....	24
2.3.2.4	Pengelolaan Data Tempat Wisata.....	25
2.3.2.4.1	Tampil Tempat Wisata.....	25
2.3.2.4.2	Tambah Tempat Wisata.....	25
2.3.2.4.3	Edit Tempat Wisata.....	26
2.3.2.4.4	Hapus Tempat Wisata.....	26
2.3.2.5	Pengelolaan Data Pertanyaan.....	27
2.3.2.5.1	Tampil Pertanyaan.....	27
2.3.2.5.2	Tambah Pertanyaan.....	27
2.3.2.5.3	Edit Pertanyaan.....	28
2.3.2.5.4	Hapus Pertanyaan.....	28
2.3.2.6	Pengelolaan Data Jawaban.....	29
2.3.2.6.1	Tampil Jawaban.....	29
2.3.2.6.2	Tambah Jawaban.....	29
2.3.2.6.3	Edit Jawaban.....	30
2.3.2.6.4	Hapus Jawaban.....	30
2.3.2.7	Pengelolaan Data Badge.....	31
2.3.2.7.1	Tampil Badge.....	31
2.3.2.7.2	Tambah Badge.....	31
2.3.2.7.3	Edit Badge.....	32
2.3.2.7.4	Hapus Badge.....	32
2.3.2.8	Pengelolaan Data Foto.....	33
2.3.2.8.1	Tampil Foto.....	33
2.3.2.8.2	Tambah Foto.....	33
2.3.2.8.3	Hapus Foto.....	34

2.3.3	Class Diagram Aplikasi TandaTanya untuk User (Android) ...	35
2.3.4	Class Diagram Aplikasi TandaTanya untuk Administrator (Web)	36
2.3.5	Class Diagram Specific Descriptions TandaTanya (Android) .	37
2.3.5.1	Specific Design Class LoginActivity.....	37
2.3.5.2	Spesific Design Class MainActivity.....	37
2.3.5.3	Specific Design Class ProfileFragment.....	37
2.3.5.4	Spesific Design Class FriendsFragment.....	38
2.3.5.5	Specific Design Class HistoryFragment.....	38
2.3.5.6	Specific Design Class AboutUsFragment.....	38
2.3.5.7	Specific Design Class MapActivity.....	39
2.3.5.8	Specific Design Class QuizActivity.....	39
2.3.5.9	Specific Design Class QuizResultActivity.....	40
2.3.5.10	Specific Design Class PlaceActivity.....	40
2.3.5.11	Specific Design Class UserController.....	40
2.3.5.12	Specific Design Class PlaceController.....	41
2.3.5.13	Specific Design Class QuizController.....	41
2.3.5.14	Specific Design Class BadgeController.....	41
2.3.5.15	Specific Design Class CheckPointController.....	42
2.3.5.16	Specific Design Class HistoryController.....	42
2.3.5.17	Specific Design Class UserModel.....	42
2.3.5.18	Specific Design Class PlaceModel.....	42
2.3.5.19	Specific Design Class QuizModel.....	43
2.3.5.20	Specific Design Class BadgeModel.....	43
2.3.5.21	Specific Design Class CheckPointModel.....	43
2.3.5.22	Specific Design Class HistoryModel.....	44
2.3.6	Class Diagram Specific Description TandaTanya (Web) .....	44
2.3.6.1	Specific Design Class Login.....	44
2.3.6.2	Specific Design Class Home.....	44
2.3.6.3	Specific Design Class category_manager.....	44

2.3.6.4	Specific Design Class place_manager.....	44
2.3.6.5	Specific Design Class badge_manager.....	45
2.3.6.6	Specific Design Class answer_view.....	45
2.3.6.7	Specific Design Class tandmin.....	45
2.3.6.8	Specific Design Class tandatanya_model.....	47
3	Perancangan Data .....	52
3.1	Dekomposisi Data .....	52
3.1.1	Deskripsi Entitas Users .....	52
3.1.2	Deskripsi Entitas Category_places .....	52
3.1.3	Deskripsi Entitas Places .....	52
3.1.4	Deskripsi Entitas Questions .....	52
3.1.5	Deskripsi Entitas Answers .....	52
3.1.6	Deskripsi Entitas Badges .....	53
3.1.7	Deskripsi Entitas Badgeuser .....	53
3.1.8	Deskripsi Entitas Checkpoint .....	53
3.1.9	Deskripsi Entitas History .....	53
3.1.10	Deskripsi Entitas Photos .....	53
3.2	<i>Physical Data Model</i> .....	55
4	Perancangan Antarmuka .....	56
4.1	Sketsa UI Aplikasi TandaTanya untuk User (Android) .....	56
4.1.1	Login.....	56
4.1.2	Home .....	57
4.1.3	Menu Navigasi .....	58
4.1.4	Menampilkan Lokasi Tempat Wisata .....	59
4.1.5	Kuis .....	60
4.1.6	Menampilkan Skor Kuis .....	61
4.1.7	Menampilkan Profil Tempat Wisata .....	62
4.1.8	Menampilkan Foto Galeri .....	63
4.1.9	Menampilkan Sejarah Pengguna .....	64
4.1.10	Menampilkan Teman .....	65
4.1.11	Menampilkan Tentang Aplikasi TandaTanya .....	66
4.2	Sketsa UI Aplikasi TandaTanya untuk Administrator (Web) .....	67
4.2.1	Login.....	67
4.2.2	Antarmuka Kelola Data Pertanyaan .....	67
4.2.3	Antarmuka Tambah Data Pertanyaan .....	69
4.2.4	Antarmuka Edit Data Pertanyaan .....	70

4.2.5	Antarmuka Hapus Data Pertanyaan .....	70
4.2.6	Antarmuka Kelola Data Jawaban .....	71
4.2.7	Antarmuka Tambah Data Jawaban .....	72
4.2.8	Antarmuka Edit Data Jawaban .....	73
4.2.9	Antarmuka Hapus Data Jawaban .....	73
4.2.10	Antarmuka Kelola Data Kategori Tempat Wisata .....	74
4.2.11	Antarmuka Tambah Data Kategori Tempat Wisata .....	75
4.2.12	Antarmuka Edit Data Kategori Tempat Wisata .....	76
4.2.13	Antarmuka Hapus Data Kategori Tempat Wisata .....	76
4.2.14	Antarmuka Kelola Data Badge .....	77
4.2.15	Antarmuka Tambah Data Badge .....	78
4.2.16	Antarmuka Edit Data Badge .....	79
4.2.17	Antarmuka Hapus Data Badge .....	80
4.2.18	Antarmuka Kelola Data Tempat Wisata .....	80
4.2.19	Antarmuka Tambah Data Tempat wisata .....	81
4.2.20	Antarmuka Edit Data Tempat wisata .....	82
4.2.21	Antarmuka Hapus Data Tempat wisata .....	83
4.2.22	Antarmuka Kelola Data Foto .....	84
4.2.23	Antarmuka Tambah Data Foto .....	85
4.2.24	Antarmuka Hapus Data Foto .....	85

## Daftar Gambar

Gambar 1. Perancangan Arsitektur Aplikasi TandaTanya untuk User (Android).....	14
Gambar 2. Perancangan Arsitektur Aplikasi TandaTanya untuk Administrator (Web).....	15
Gambar 3. Sequence Diagram : Login.....	16
Gambar 4. Sequence Diagram : Logout.....	16
Gambar 5. Sequence Diagram : Menampilkan Profil Pengguna.....	17
Gambar 6. Sequence Diagram : Menampilkan Sejarah Pengguna.....	17
Gambar 7. Sequence Diagram : Menampilkan Teman Pengguna.....	18
Gambar 8. Sequence Diagram : Menampilkan Tempat Wisata pada Peta	18
Gambar 9. Sequence Diagram : Menampilkan Profil Tempat Wisata...	19
Gambar 10. Sequence Diagram : Menampilkan Tempat Wisata Tujuan pada Peta.....	19
Gambar 10. Sequence Diagram : Answer Quiz.....	20
Gambar 11. Sequence Diagram : Check In.....	21
Gambar 12. Sequence Diagram : Menampilkan Skor.....	21
Gambar 13. Sequence Diagram : Login Admin.....	22
Gambar 14. Sequence Diagram : Logout Admin.....	22
Gambar 15. Sequence Diagram : Tampil Kategori Tempat Wisata.....	23
Gambar 16. Sequence Diagram : Tambah Kategori Tempat Wisata.....	23
Gambar 17. Sequence Diagram : Edit Kategori Tempat Wisata.....	24
Gambar 18. Sequence Diagram : Hapus Kategori Tempat Wisata.....	24
Gambar 19. Sequence Diagram : Tampil Tempat Wisata.....	25
Gambar 20. Sequence Diagram : Tambah Tempat Wisata.....	25
Gambar 21. Sequence Diagram : Edit Tempat Wisata.....	26
Gambar 22. Sequence Diagram : Hapus Tempat Wisata.....	26
Gambar 23. Sequence Diagram : Tampil Pertanyaan.....	27
Gambar 24. Sequence Diagram : Tambah Pertanyaan.....	27
Gambar 25. Sequence Diagram : Edit Pertanyaan.....	28
Gambar 26. Sequence Diagram : Hapus Pertanyaan.....	28
Gambar 27. Sequence Diagram : Tampil Jawaban.....	29
Gambar 28. Sequence Diagram : Tambah Jawaban.....	29
Gambar 29. Sequence Diagram : Edit Jawaban.....	30
Gambar 30. Sequence Diagram : Hapus Jawaban.....	30

Gambar 31. Sequence Diagram : Tampil Badge.....	31
Gambar 32. Sequence Diagram : Tambah Badge.....	31
Gambar 33. Sequence Diagram : Edit Badge.....	32
Gambar 34. Sequence Diagram : Hapus Badge.....	32
Gambar 35. Sequence Diagram : Tampil Foto.....	33
Gambar 36. Sequence Diagram : Tambah Foto.....	34
Gambar 37. Sequence Diagram : Hapus Foto.....	34
Gambar 46. Class Diagram TandaTanya (Android).....	35
Gambar 47. Class Diagram TandaTanya (Web).....	36
Gambar 48. Physical Data Model.....	55
Gambar 49. Antarmuka Login.....	56
Gambar 50. Antarmuka Home.....	57
Gambar 51. Antarmuka Navigasi.....	58
Gambar 52. Antarmuka Tampil Lokasi Tempat Wisata.....	59
Gambar 53. Antarmuka Upload Gambar.....	60
Gambar 54. Antarmuka Menampilkan Skor Kuis.....	61
Gambar 55. Antarmuka Ubah Menampilkan Profil Tempat Wisata.....	62
Gambar 56. Antarmuka Menampilkan Foto Galeri.....	63
Gambar 57. Antarmuka Sejarah Pengguna.....	64
Gambar 58. Antarmuka Menampilkan Teman.....	65
Gambar 60. Menampilkan Tentang Aplikasi TandaTanya.....	66
Gambar 64. Antarmuka Login Web.....	67
Gambar 65. Antarmuka Kelola Data Pertanyaan.....	68
Gambar 66. Antarmuka Tambah Data Pertanyaan.....	69
Gambar 68. Antarmuka Edit Data Pertanyaan.....	70
Gambar 68. Antarmuka Hapus Data Pertanyaan.....	70
Gambar 68. Antarmuka Kelola Data Jawaban.....	71
Gambar 68. Antarmuka Tambah Data Jawaban.....	72
Gambar 68. Antarmuka Edit Data Jawaban.....	73
Gambar 68. Antarmuka Hapus Data Jawaban.....	73
Gambar 68. Antarmuka Kelola Data Kategori Tempat Wisata.....	74
Gambar 68. Antarmuka Tambah Data Kategori Tempat Wisata.....	75
Gambar 68. Antarmuka Edit Data Kategori Tempat Wisata.....	76
Gambar 68. Antarmuka Hapus Data Kategori Tempat Wisata.....	76
Gambar 68. Antarmuka Kelola Data Badge.....	77
Gambar 68. Antarmuka Tambah Data Badge.....	78
Gambar 68. Antarmuka Edit Data Badge.....	79



Gambar 68. Antarmuka Hapus Data Badge.....	80
Gambar 68. Antarmuka Kelola Data Tempat wisata.....	80
Gambar 68. Antarmuka Tambah Data Tempat wisata.....	81
Gambar 68. Antarmuka Edit Data Tempat wisata.....	82
Gambar 68. Antarmuka Hapus Data Tempat wisata.....	83
Gambar 68. Antarmuka Kelola Data Foto.....	84
Gambar 68. Antarmuka Tambah Data Foto.....	85
Gambar 68. Antarmuka Hapus Data Foto.....	85

## 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 TANDATANYA dikembangkan dengan tujuan untuk :

1. Mampu membangun aplikasi TANDATANYA yang dapat mencari dan menampilkan posisi lokasi-lokasi pariwisata pada perangkat *smartphone* dengan sistem operasi Android.
2. Mampu membangun aplikasi TANDATANYA yang dapat menentukan lokasi pariwisata tujuan secara acak kepada pengguna pada perangkat *smarphone* dengan sistem operasi Android.
3. Mampu membangun aplikasi TANDATANYA yang dapat menampilkan beberapa pertanyaan seputar tempat pariwisata yang terpilih pada perangkat *smarphone* dengan sistem operasi Android.
4. Mampu membangun aplikasi TANDATANYA yang dapat menampilkan perolehan angka yang didapatkan dari menjawab pertanyaan-pertanyaan yang diberikan pada perangkat *smartphone* dengan sistem operasi Android.

### 1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Program Studi Teknik Informatika	DPPL-TandaTanya	12/ 85
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		

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
TANDATANYA	Perangkat lunak untuk berbagi dan pencarian event.
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.
ANDROID	Sistem operasi untuk perangkat <i>mobile</i> .
CHECK IN	Aktivitas untuk menandai bahwa pengguna berada di lokasi tertentu.

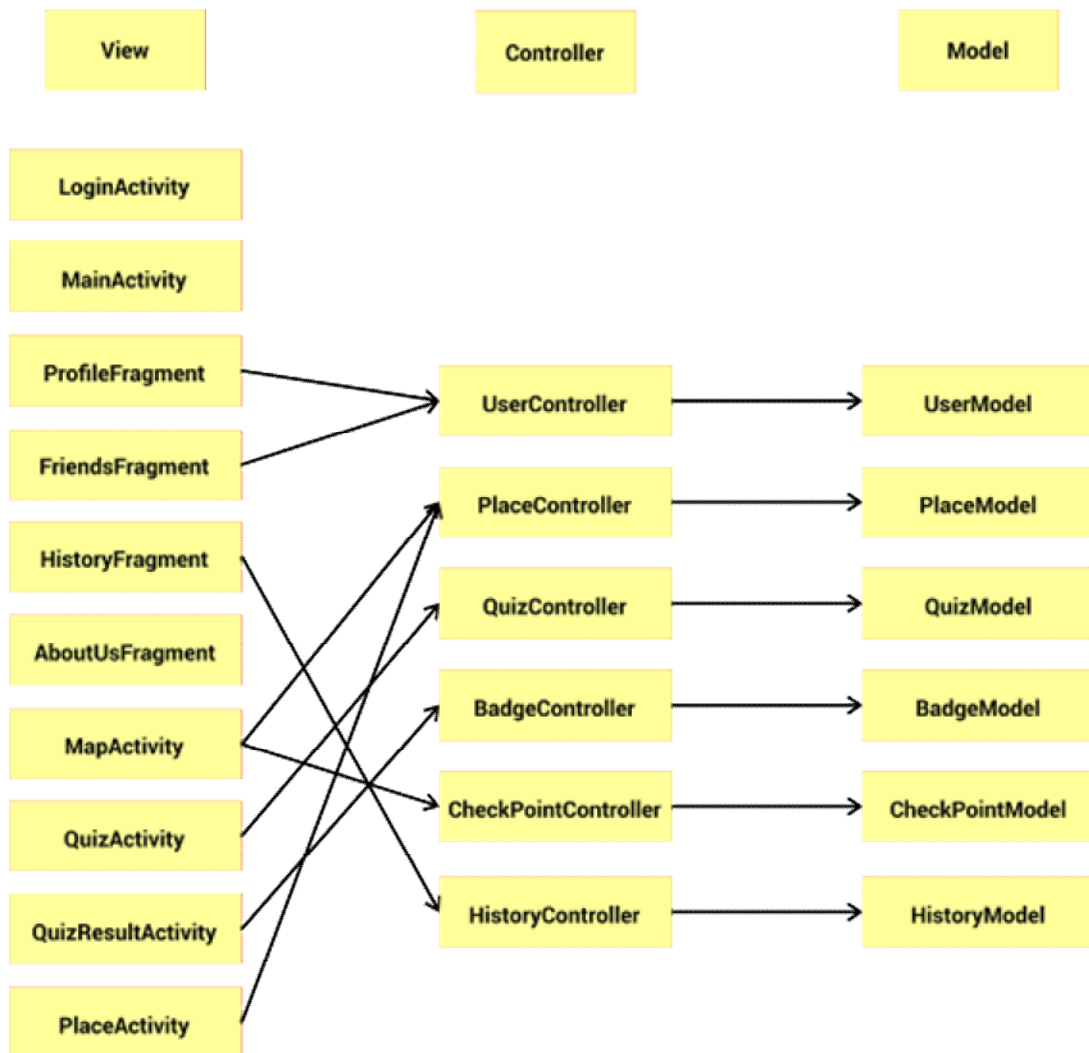
#### 1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Maharsi, Placydus Anom Bhayu. Spesifikasi Kebutuhan Perangkat Lunak SPP, Universitas Atma Jaya Yogyakarta 2012.

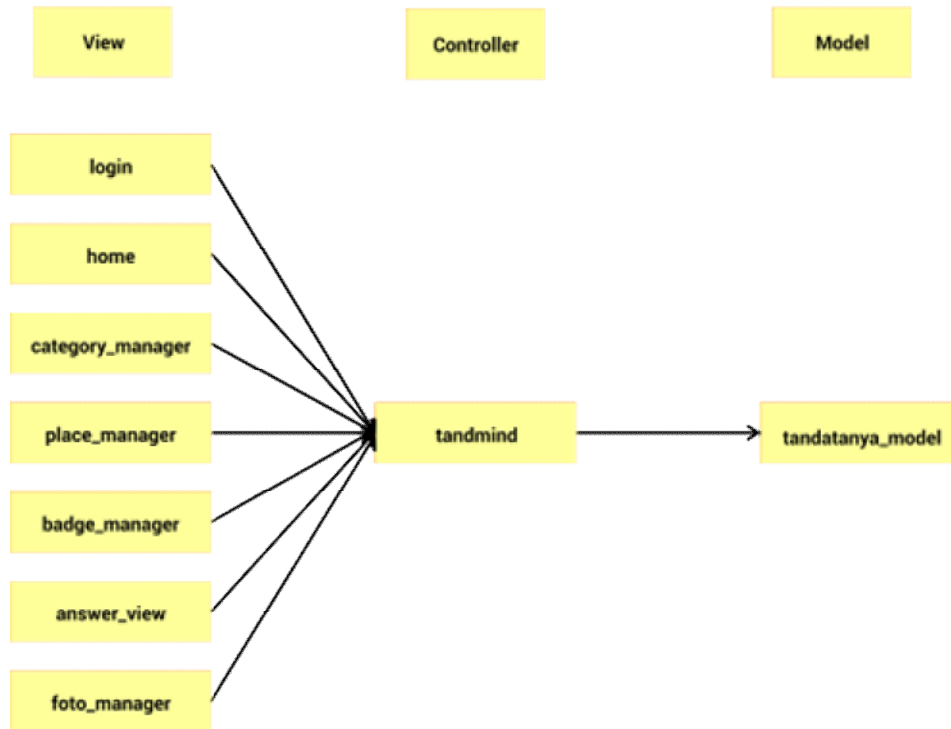
## 2 Perancangan Sistem

### 2.1 Perancangan Arsitektur Aplikasi TandaTanya untuk User (Android)



Gambar 1. Perancangan Arsitektur Aplikasi TandaTanya untuk User (Android)

## 2.2 Perancangan Arsitektur Aplikasi TandaTanya untuk Admin (Web)

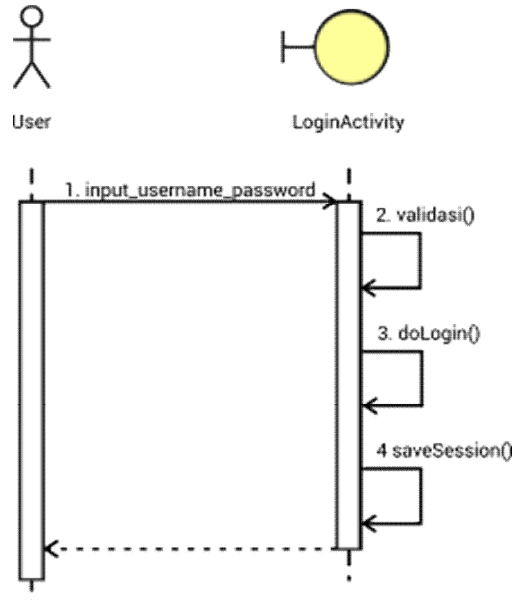


Gambar 2. Perancangan Arsitektur Aplikasi TandaTanya untuk Administrator (Web)

## 2.3 Perancangan Rinci

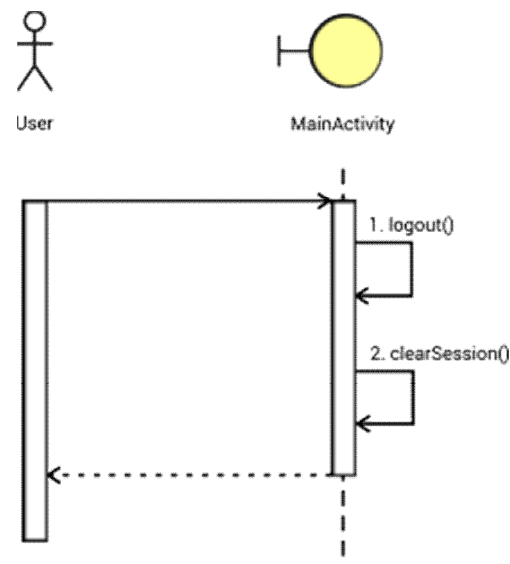
### 2.3.1 Sequence Diagram : Aplikasi TandaTanya untuk User (Android)

#### 2.3.1.1 Login



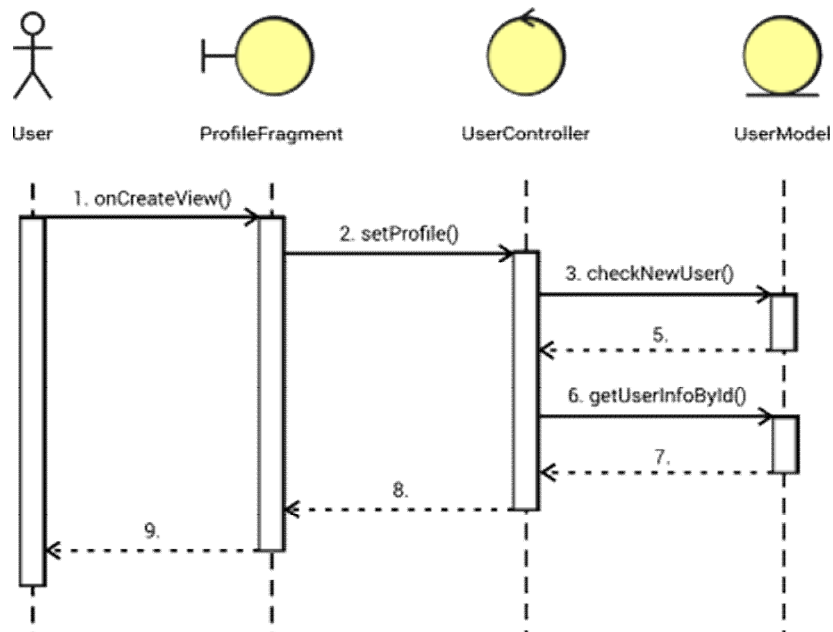
Gambar 3. Sequence Diagram : Login

#### 2.3.1.2 Logout



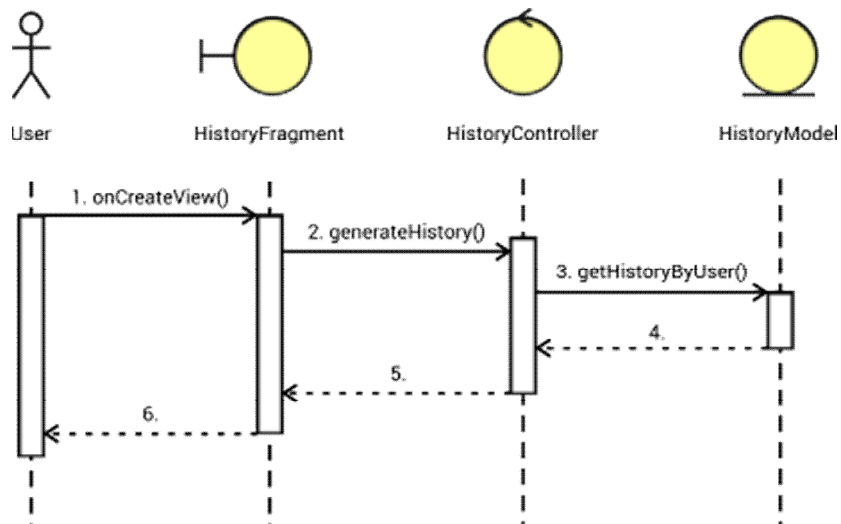
Gambar 4. Sequence Diagram : Logout

### 2.3.1.3 Menampilkan Profil Pengguna



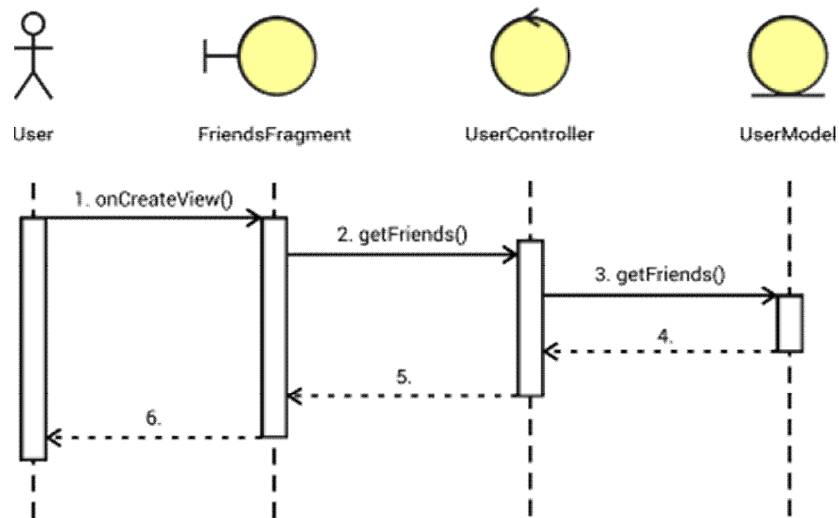
Gambar 5. Sequence Diagram : Menampilkan Profil Pengguna

### 2.3.1.4 Menampilkan Sejarah Pengguna



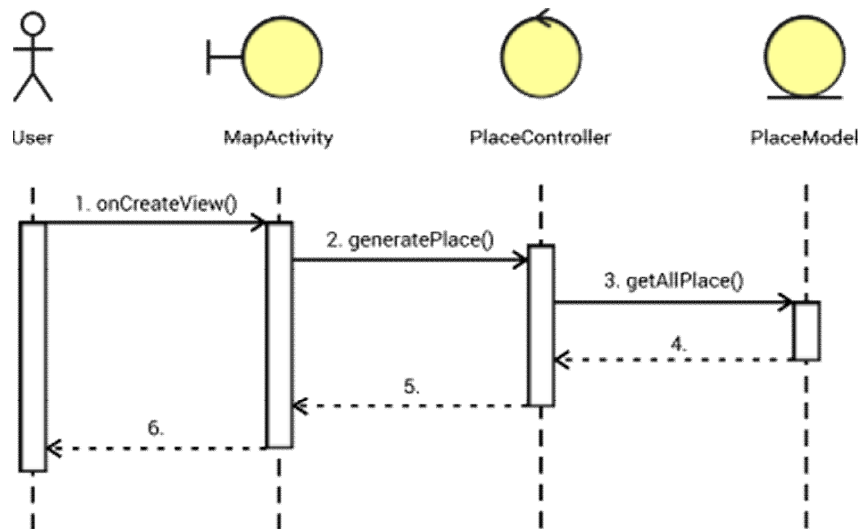
Gambar 6. Sequence Diagram : Menampilkan Sejarah Pengguna

### 2.3.1.5 Menampilkan Teman Pengguna



Gambar 7. Sequence Diagram : Menampilkan Teman Pengguna

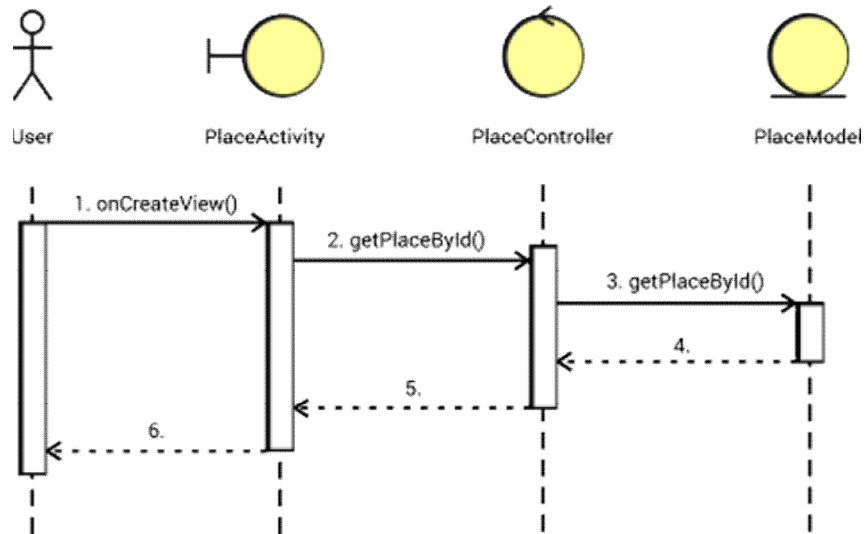
### 2.3.1.6 Menampilkan Tempat Wisata pada Peta



Gambar 8. Sequence Diagram : Menampilkan Tempat Wisata pada Peta

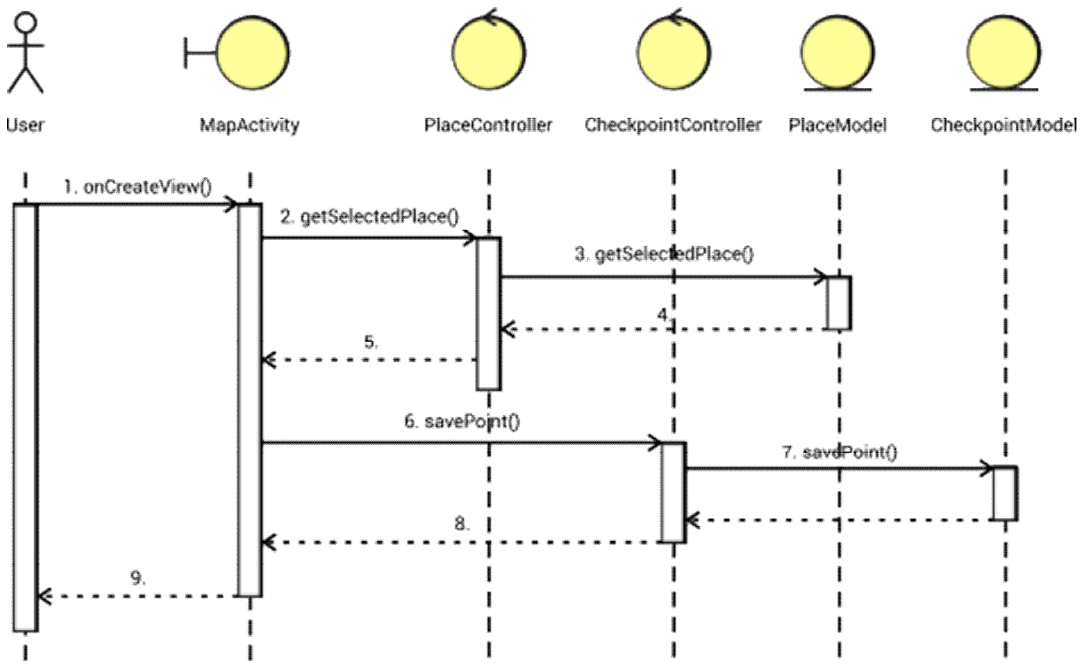


### 2.3.1.7 Menampilkan Profil Tempat Wisata



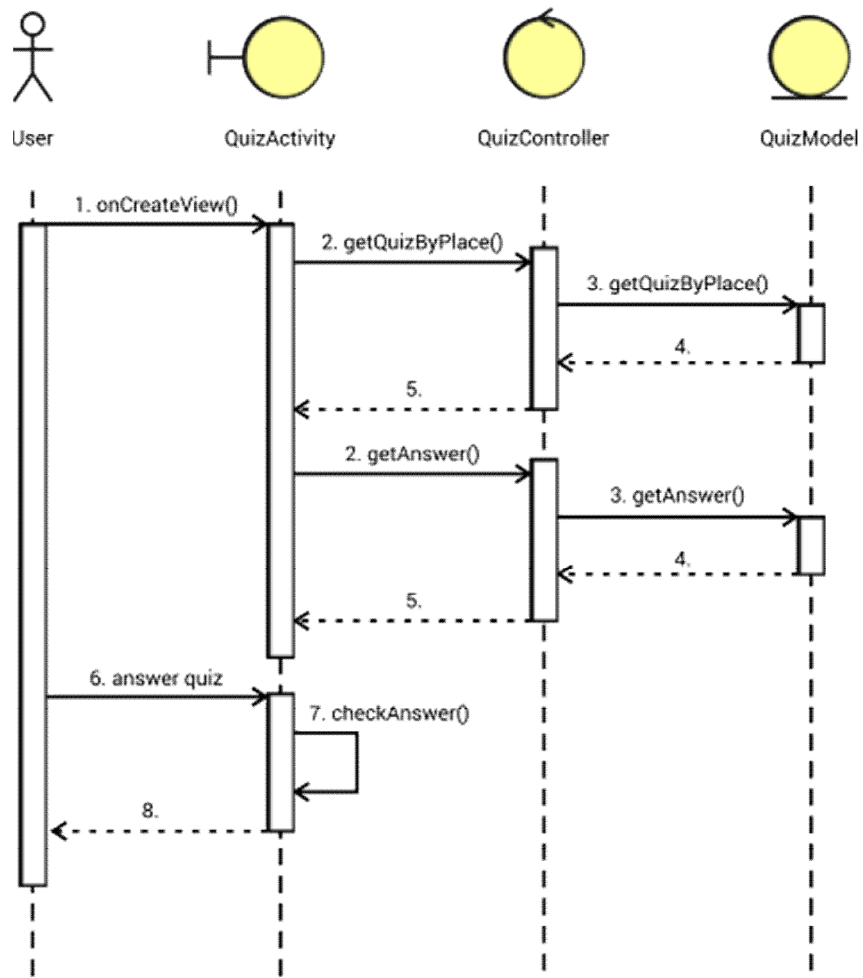
Gambar 9. Sequence Diagram : Menampilkan Profil Tempat Wisata

### 2.3.1.8 Menampilkan Tempat Wisata Tujuan pada Peta



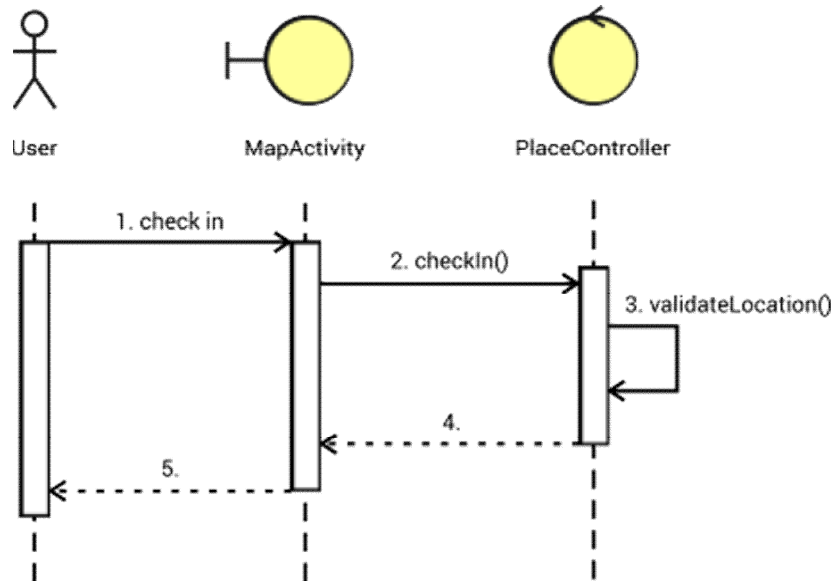
Gambar 10. Sequence Diagram : Menampilkan Tempat Wisata Tujuan pada Peta

### 2.3.1.9 Answer Quiz



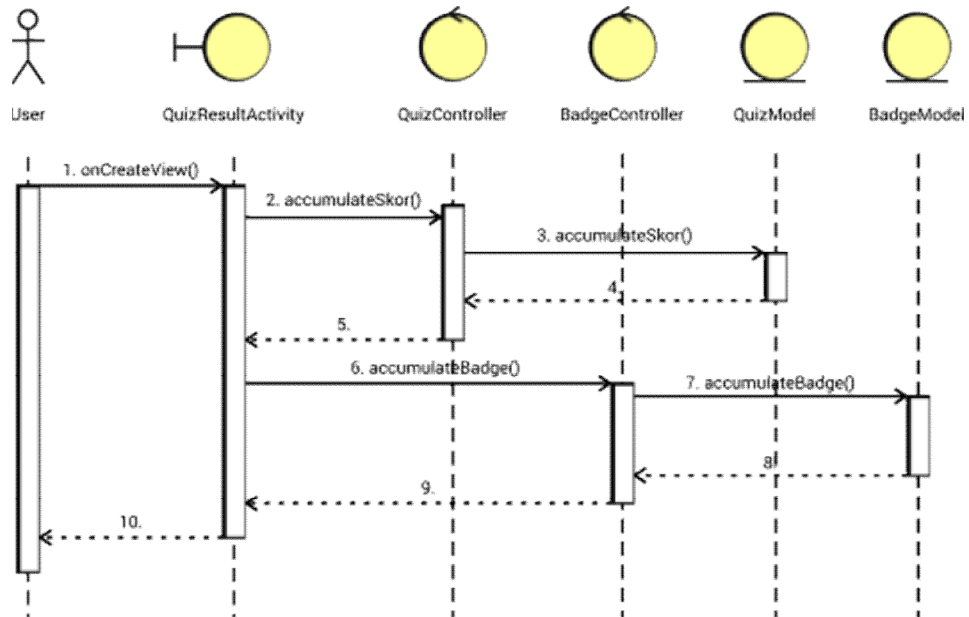
Gambar 10. Sequence Diagram : Answer Quiz

### 2.3.1.10 Check In



Gambar 11. Sequence Diagram : Check In

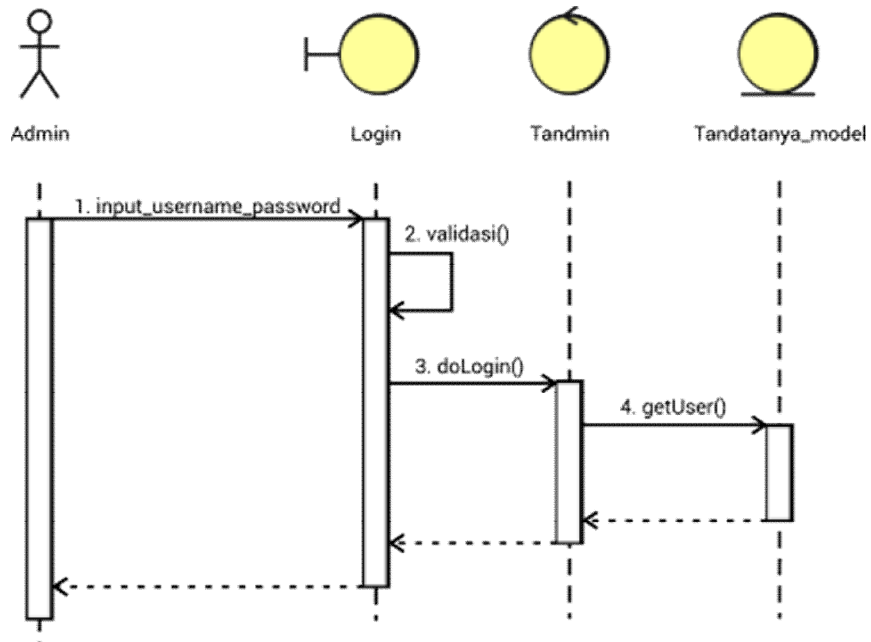
### 2.3.1.11 Menampilkan Skor



Gambar 12. Sequence Diagram : Menampilkan Skor

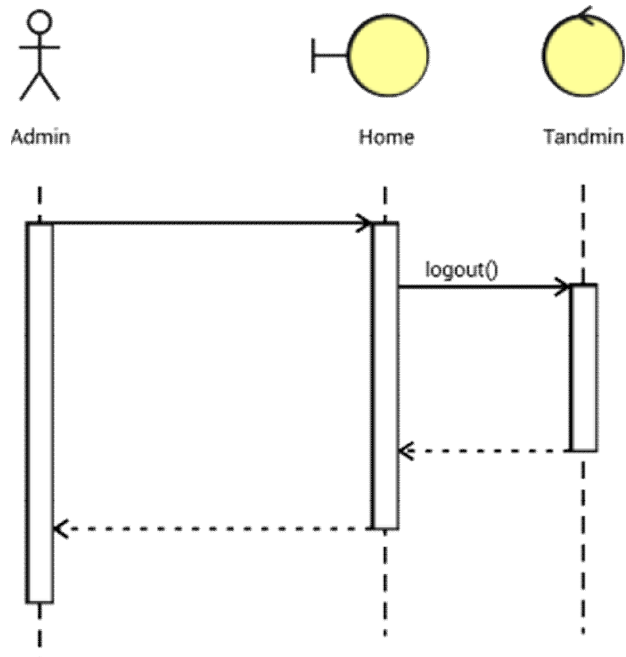
### 2.3.2 Sequence Diagram : Aplikasi TandaTanya untuk Administrator (Web)

#### 2.3.2.1 Login Admin



Gambar 13. Sequence Diagram : Login Admin

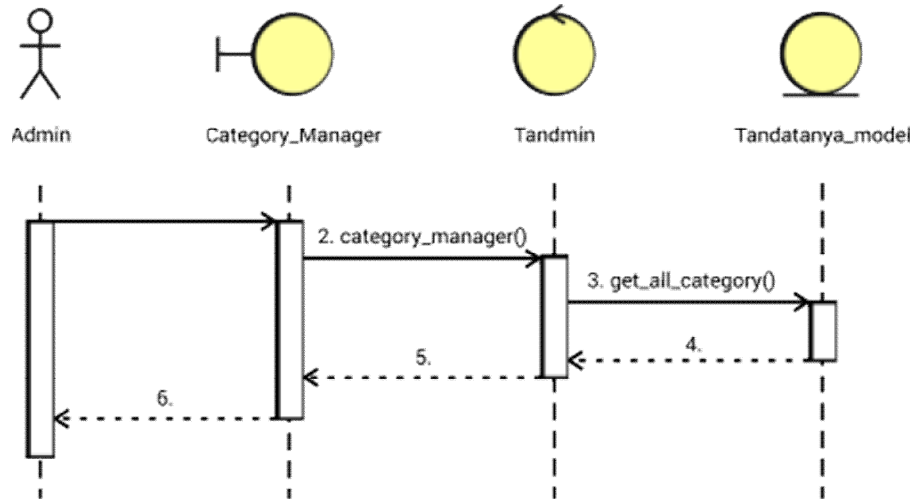
#### 2.3.2.2 Logout Admin



Gambar 14. Sequence Diagram : Logout Admin

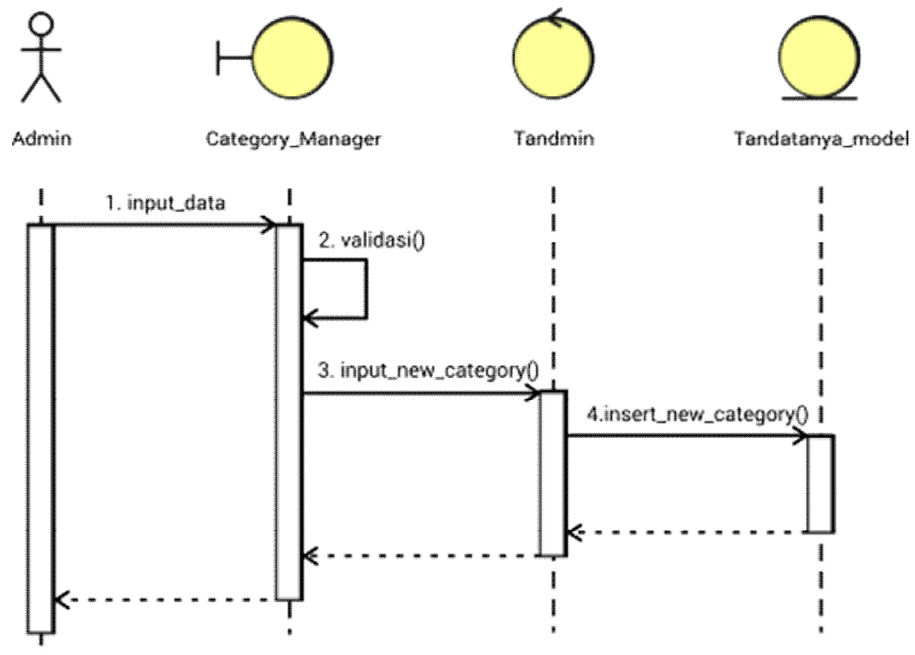
### 2.3.2.3 Pengelolaan Data Kategori Tempat Wisata

#### 2.3.2.3.1 Tampil Kategori Tempat Wisata



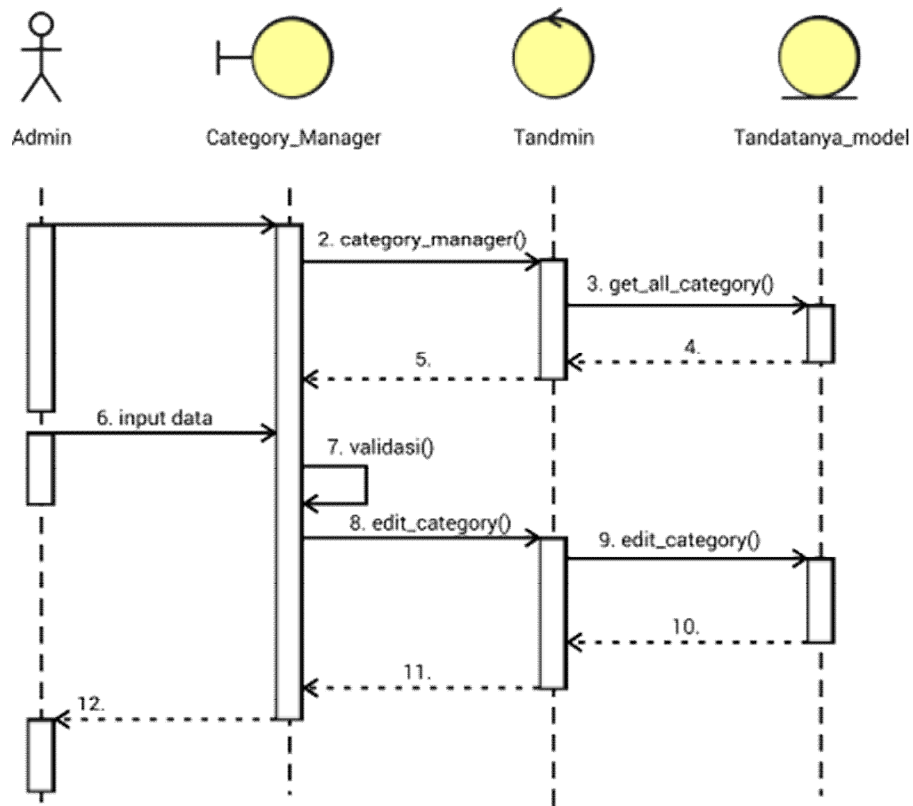
Gambar 15. Sequence Diagram : Tampil Kategori Tempat Wisata

#### 2.3.2.3.2 Tambah Kategori Tempat Wisata



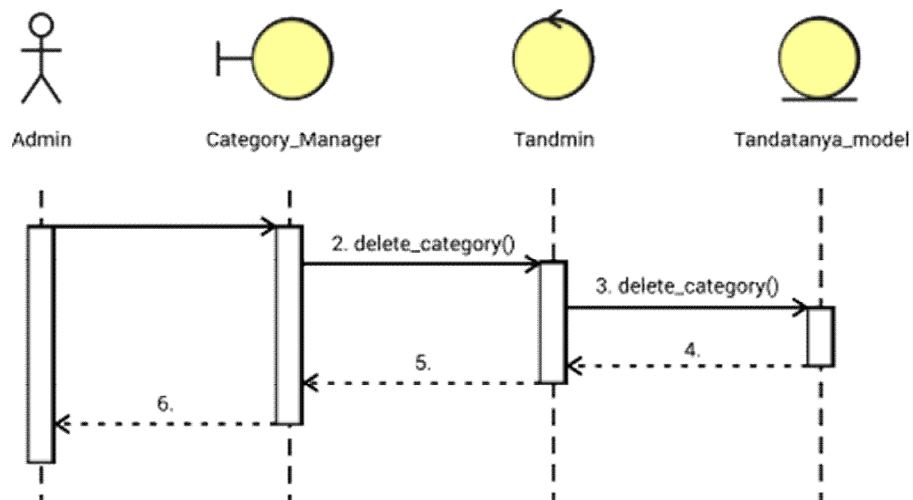
Gambar 16. Sequence Diagram : Tambah Kategori Tempat Wisata

### 2.3.2.3.3 Edit Kategori Tempat Wisata



Gambar 17. Sequence Diagram : Edit Kategori Tempat Wisata

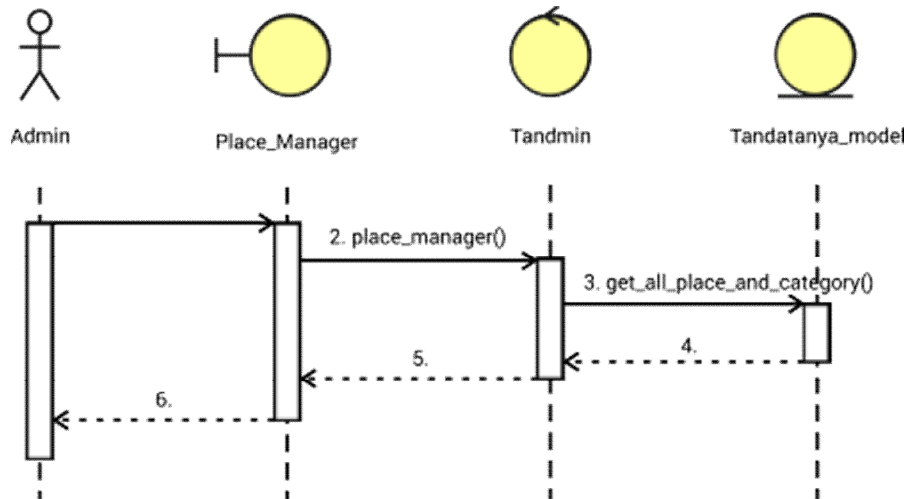
### 2.3.2.3.4 Hapus Kategori Tempat Wisata



Gambar 18. Sequence Diagram : Hapus Kategori Tempat Wisata

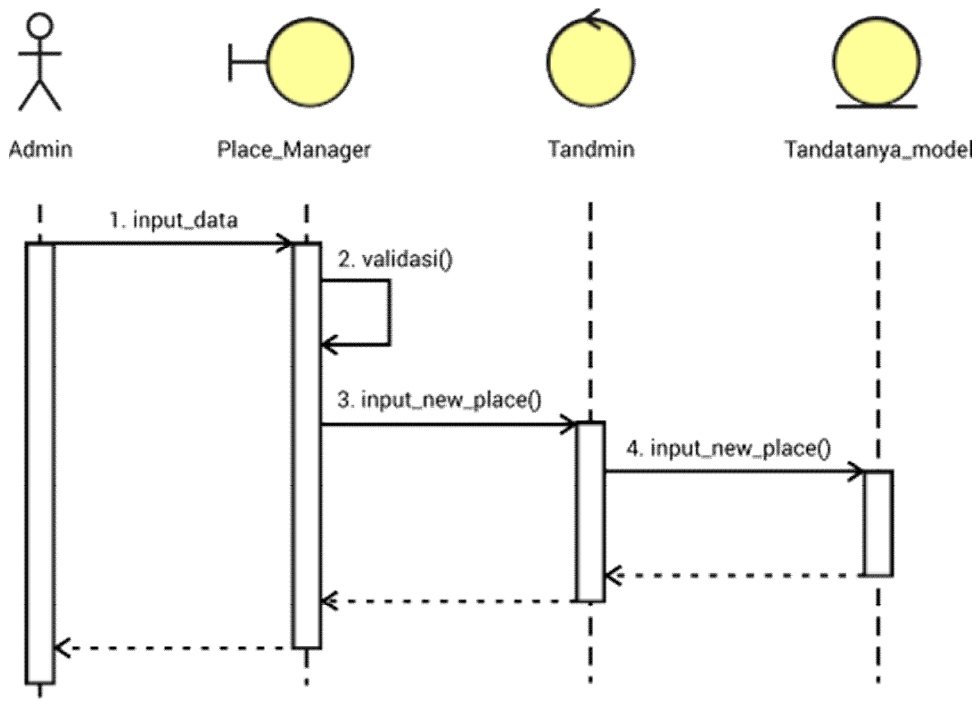
## 2.3.2.4 Pengelolaan Data Tempat Wisata

### 2.3.2.4.1 Tampil Tempat Wisata



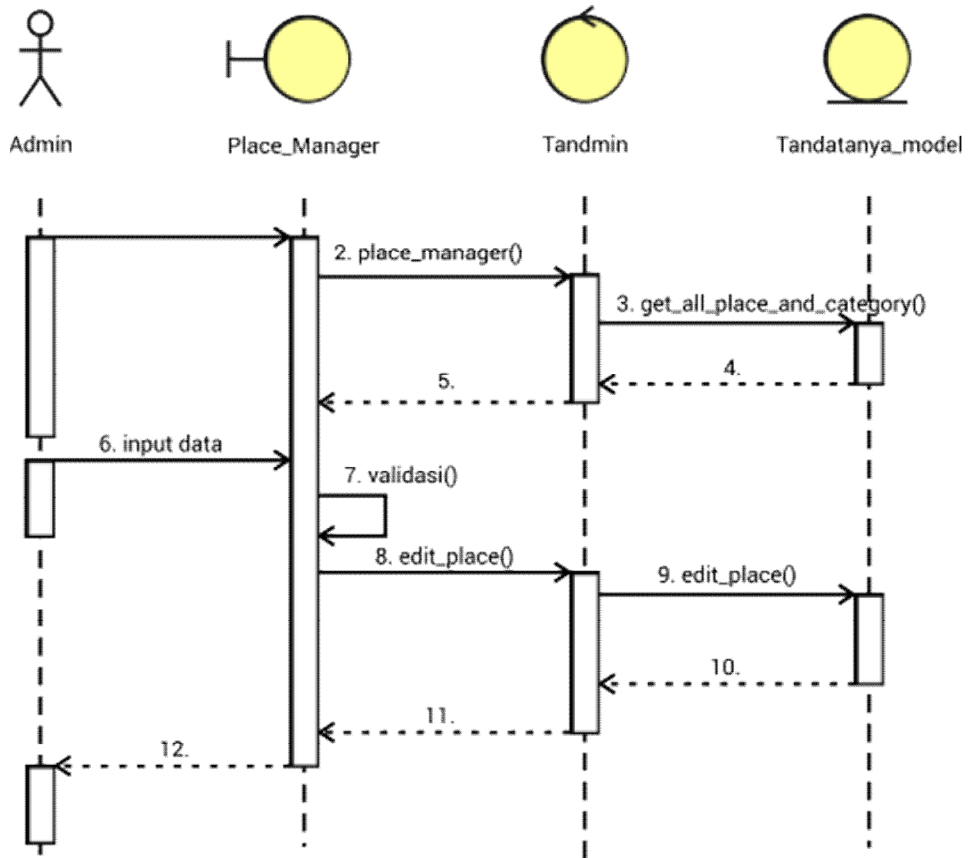
Gambar 19. Sequence Diagram : Tampil Tempat Wisata

### 2.3.2.4.2 Tambah Tempat Wisata



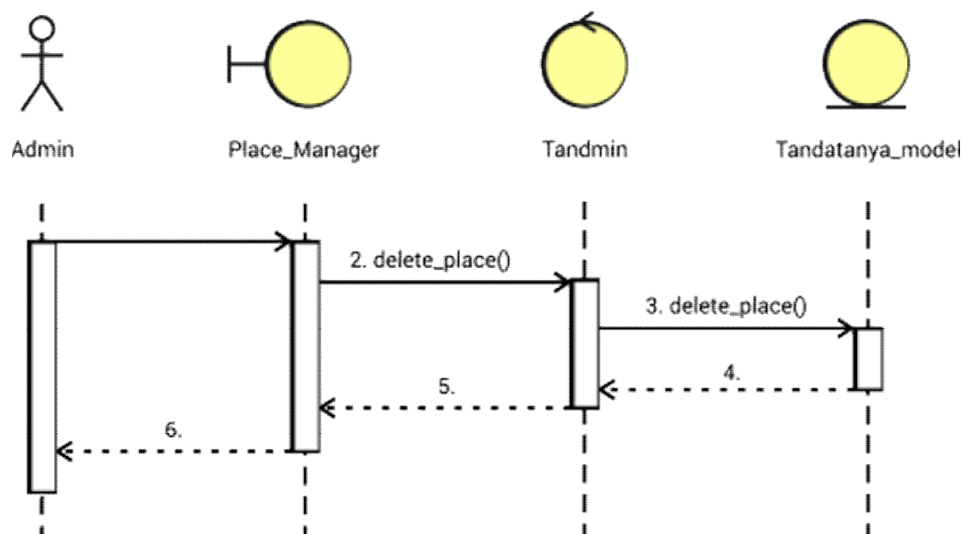
Gambar 20. Sequence Diagram : Tambah Tempat Wisata

### 2.3.2.4.3 Edit Tempat Wisata



Gambar 21. Sequence Diagram : Edit Tempat Wisata

### 2.3.2.4.4 Hapus Tempat Wisata

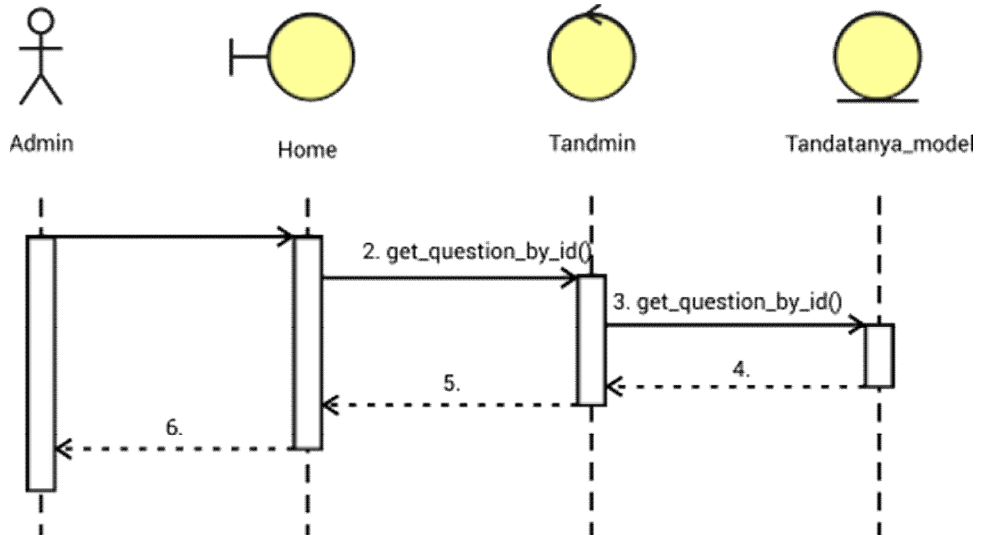


Gambar 22. Sequence Diagram : Hapus Tempat Wisata



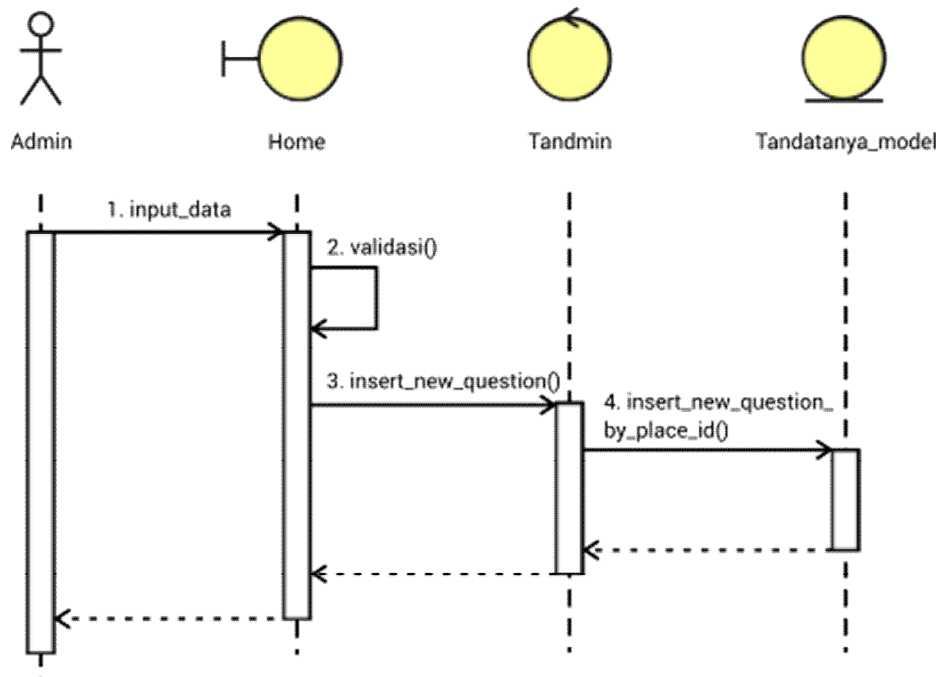
### 2.3.2.5 Pengelolaan Data Pertanyaan

#### 2.3.2.5.1 Tampil Pertanyaan



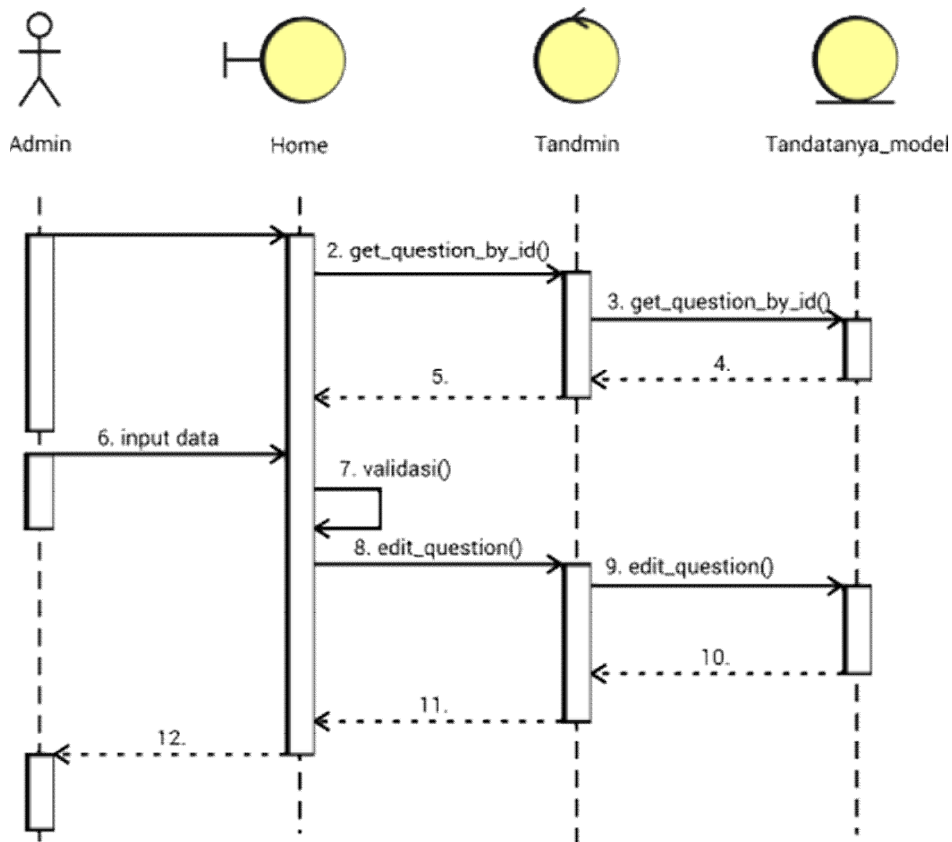
Gambar 23. Sequence Diagram : Tampil Pertanyaan

#### 2.3.2.5.2 Tambah Pertanyaan



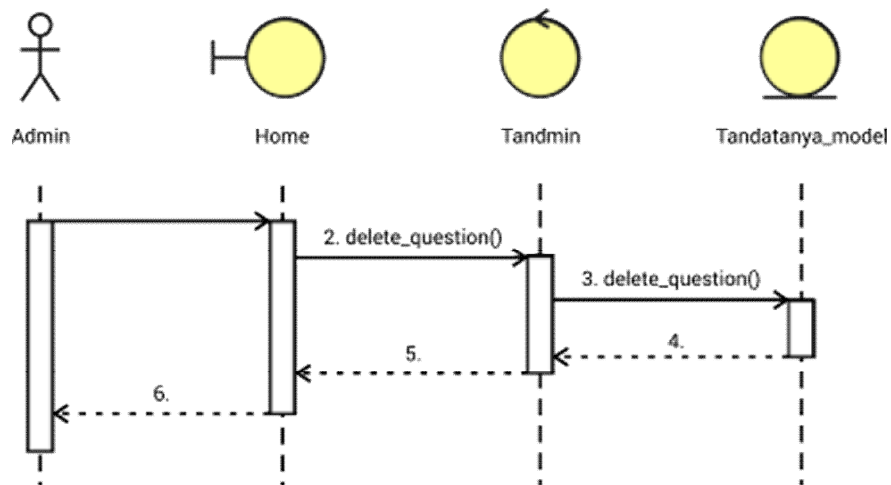
Gambar 24. Sequence Diagram : Tambah Pertanyaan

### 2.3.2.5.3 Edit Pertanyaan



Gambar 25. Sequence Diagram : Edit Pertanyaan

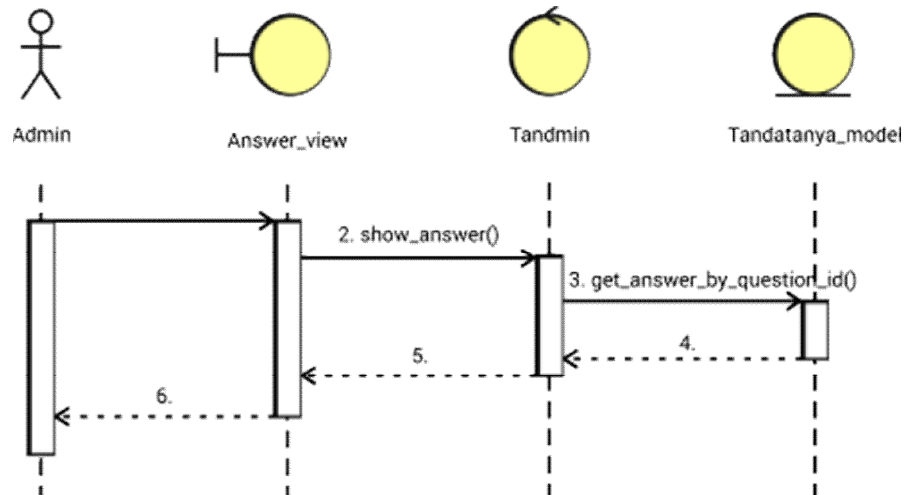
### 2.3.2.5.4 Hapus Pertanyaan



Gambar 26. Sequence Diagram : Hapus Pertanyaan

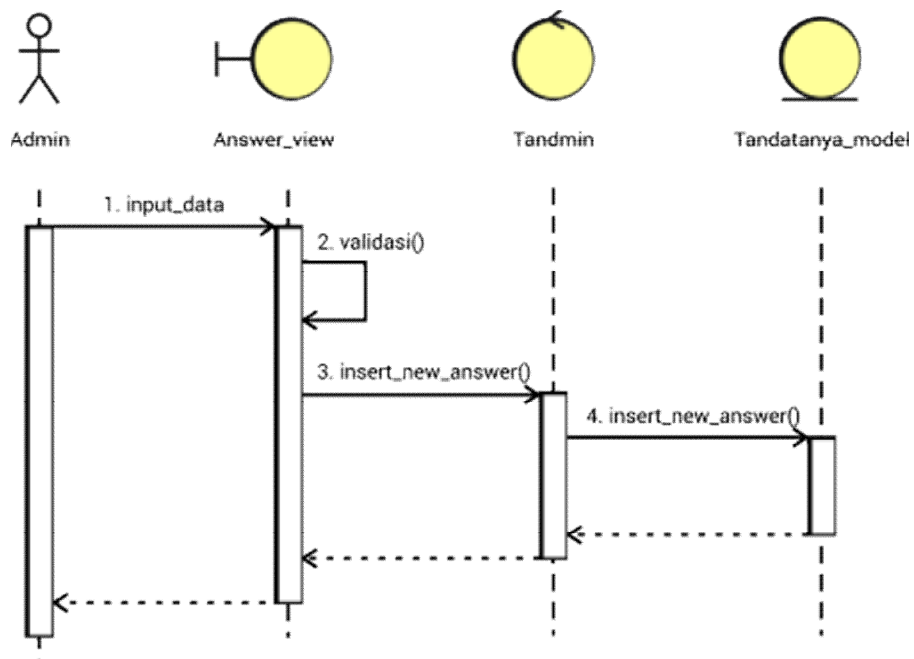
### 2.3.2.6 Pengelolaan Data Jawaban

#### 2.3.2.6.1 Tampil Jawaban



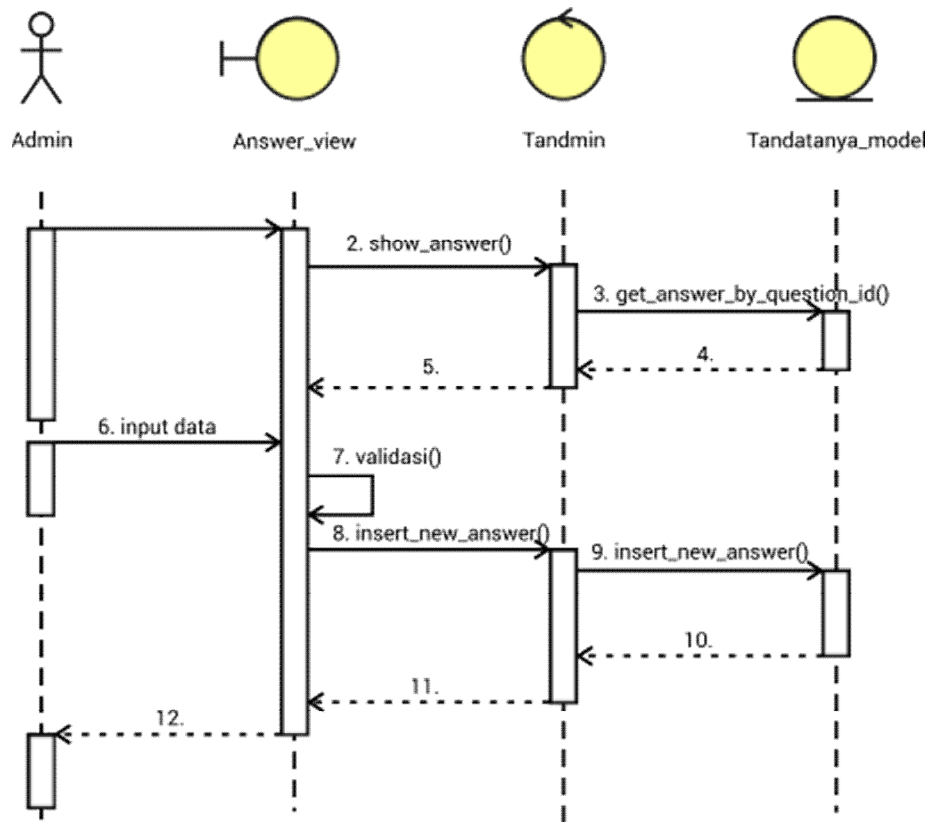
Gambar 27. Sequence Diagram : Tampil Jawaban

#### 2.3.2.6.2 Tambah Jawaban



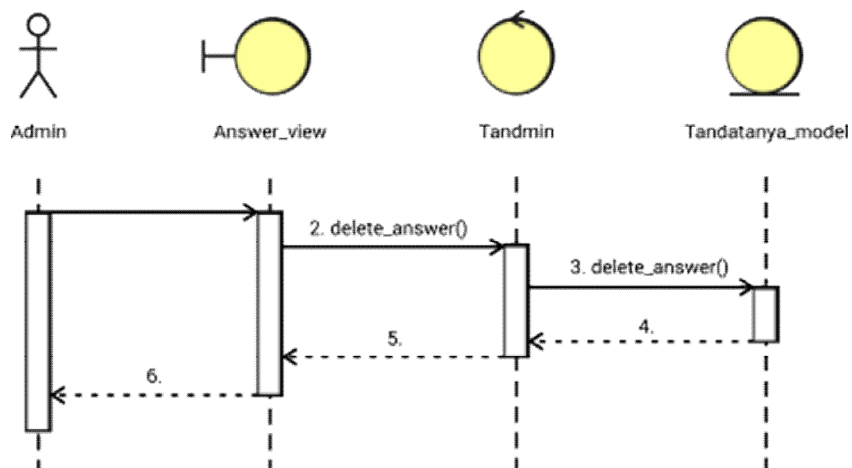
Gambar 28. Sequence Diagram : Tambah Jawaban

### 2.3.2.6.3 Edit Jawaban



Gambar 29. Sequence Diagram : Edit Jawaban

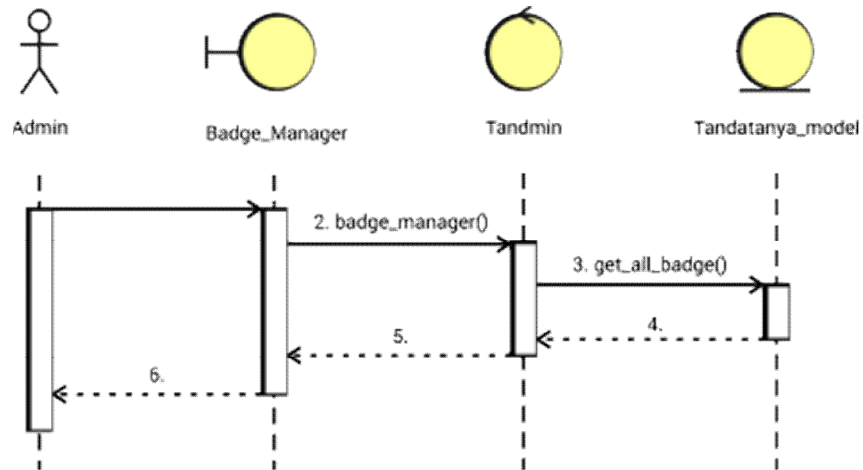
### 2.3.2.6.4 Hapus Jawaban



Gambar 30. Sequence Diagram : Hapus Jawaban

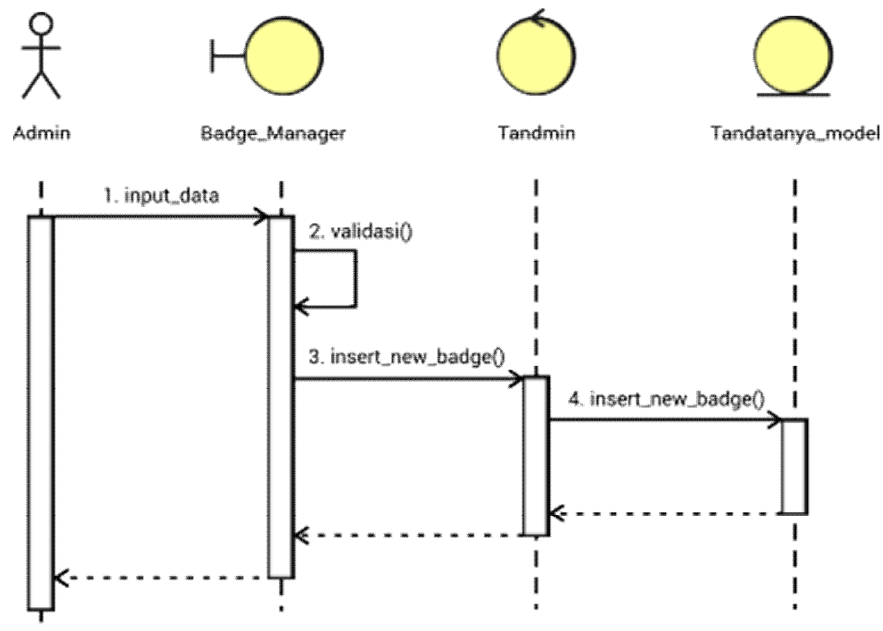
### 2.3.2.7 Pengelolaan Data Badge

#### 2.3.2.7.1 Tampil Badge



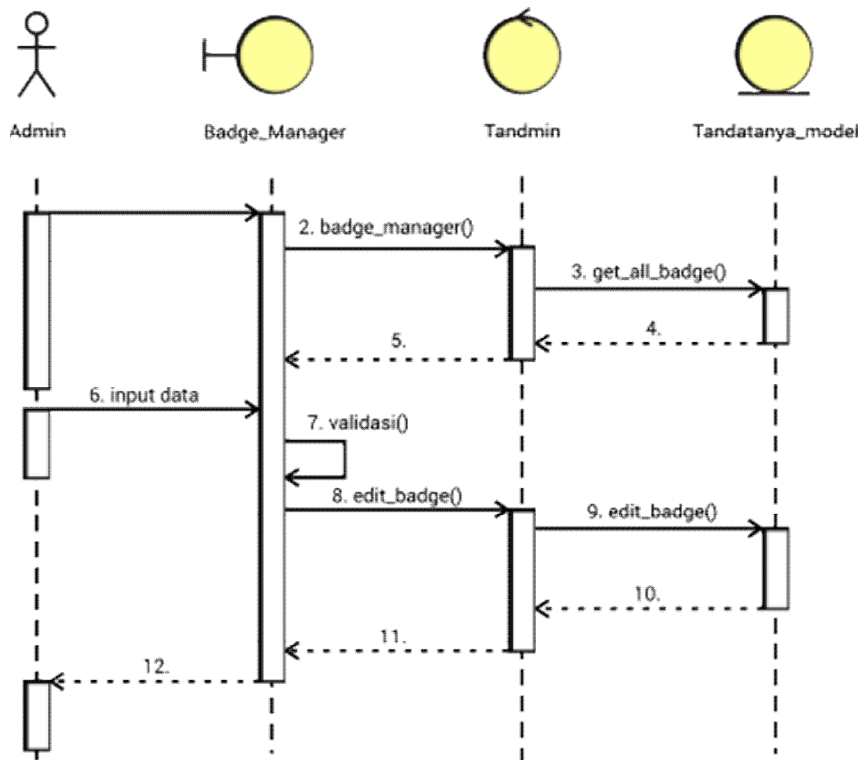
Gambar 31. Sequence Diagram : Tampil Badge

#### 2.3.2.7.2 Tambah Badge



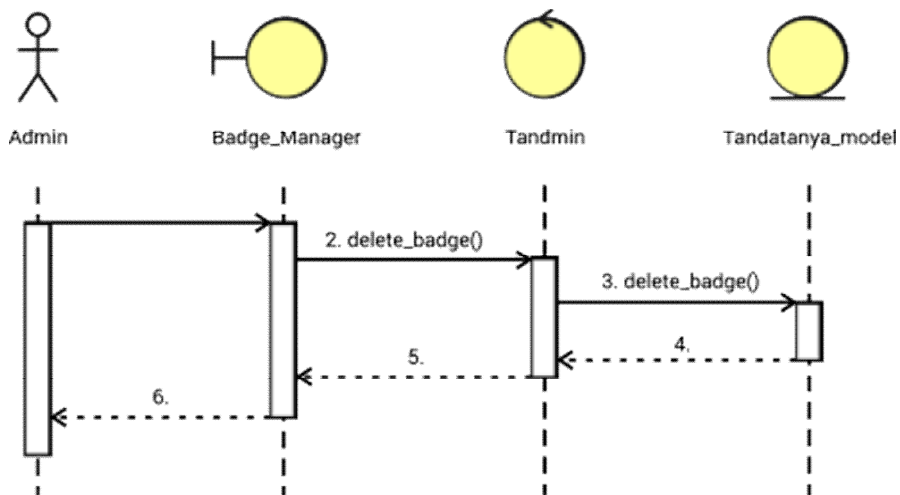
Gambar 32. Sequence Diagram : Tambah Badge

### 2.3.2.7.3 Edit Badge



Gambar 33. Sequence Diagram : Edit Badge

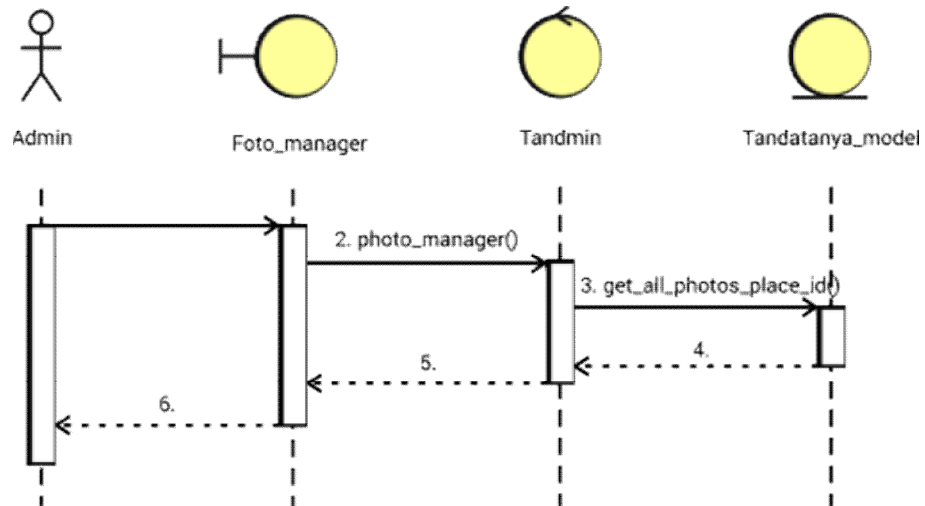
### 2.3.2.7.4 Hapus Badge



Gambar 34. Sequence Diagram : Hapus Badge

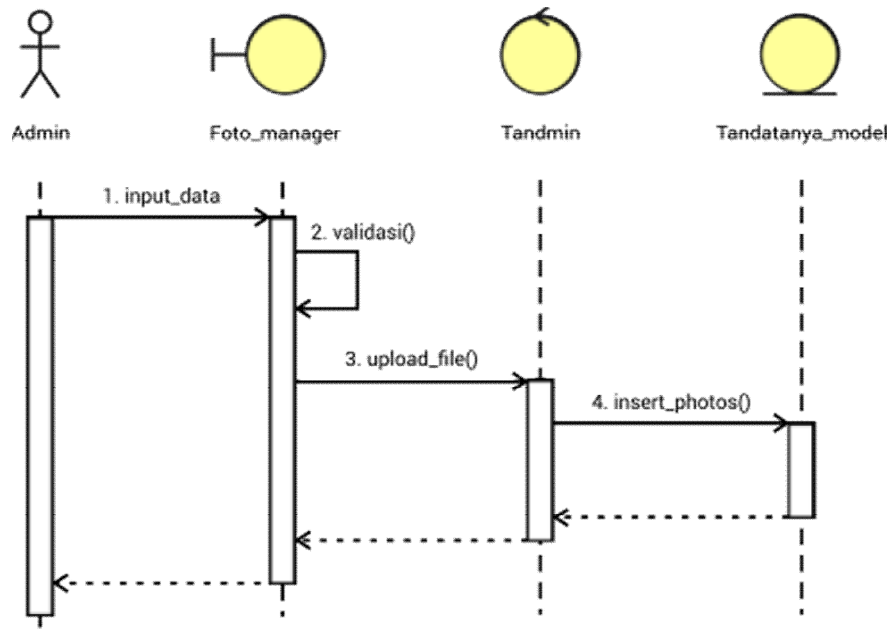
### 2.3.2.8 Pengelolaan Data Foto

#### 2.3.2.8.1 Tampil Foto



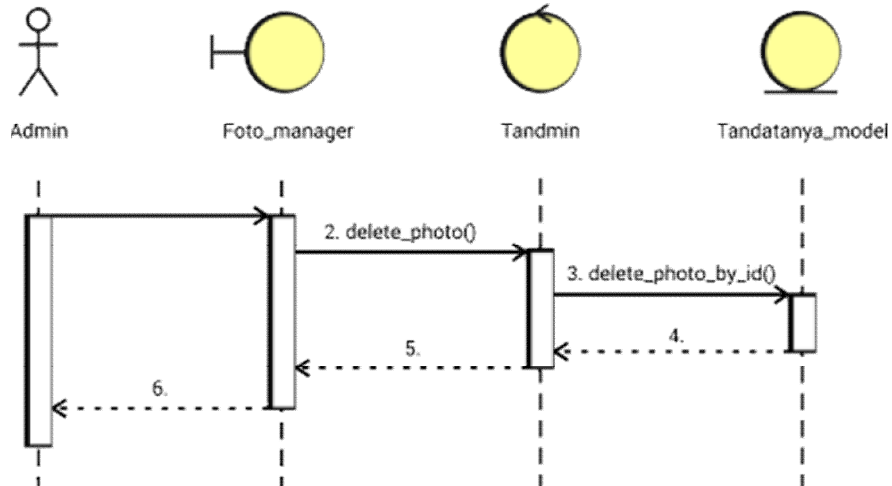
Gambar 35. Sequence Diagram : Tampil Foto

#### 2.3.2.8.2 Tambah Foto



Gambar 36. Sequence Diagram : Tambah Foto

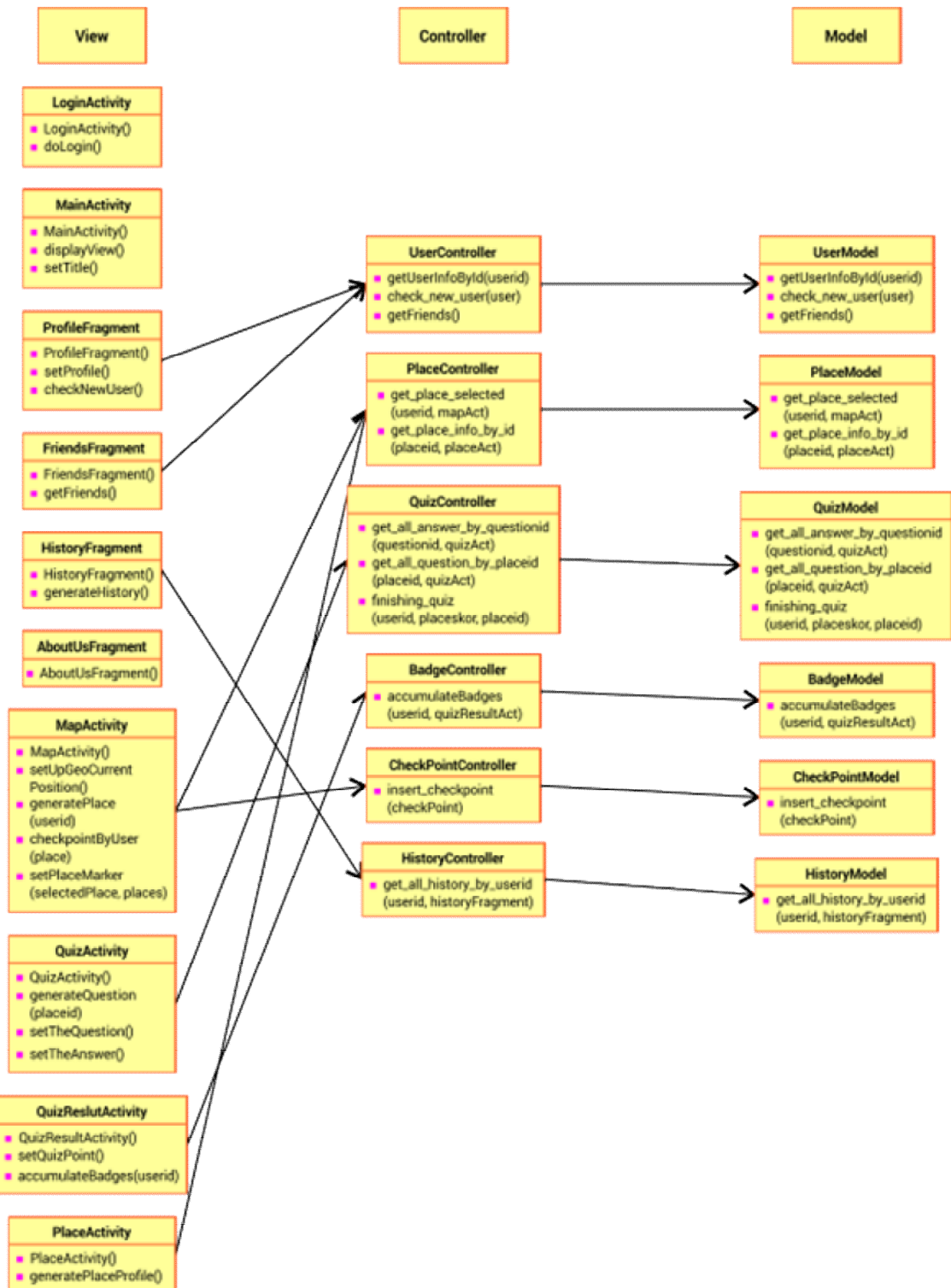
### 2.3.2.8.3 Hapus Foto



Gambar 37. Sequence Diagram : Hapus Foto

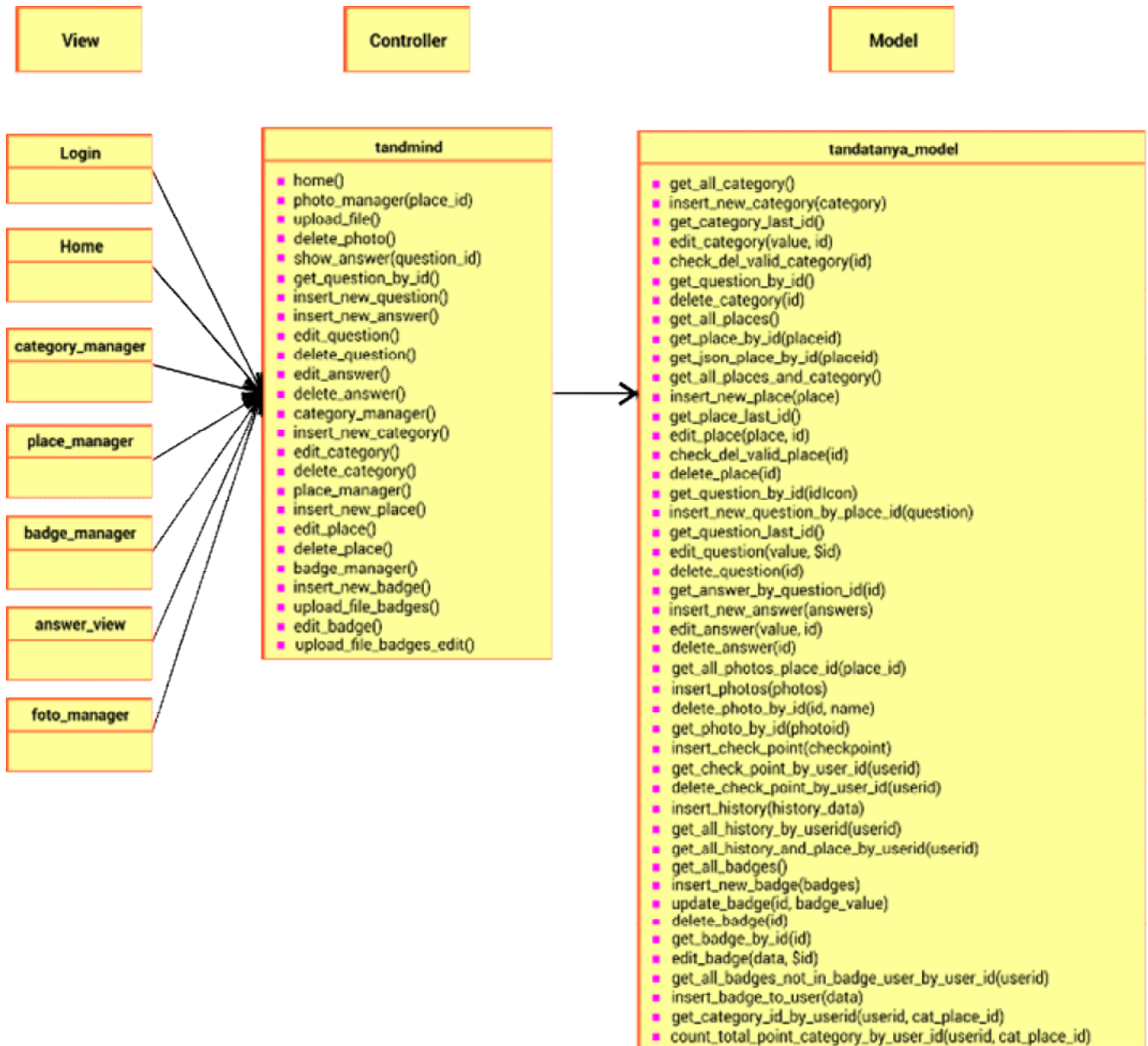


### 2.3.3 Class Diagram Aplikasi TandaTanya untuk User (Android)



Gambar 46. Class Diagram TandaTanya (Android)

### 2.3.4 Class Diagram Aplikasi TandaTanya untuk Administrator (Web)



Gambar 47. Class Diagram TandaTanya (Web)

### 2.3.5 Class Diagram Specific Descriptions TandaTanya (Android)

#### 2.3.5.1 Specific Design Class LoginActivity

<b>LoginActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<b>+LoginActivity():</b> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
<b>+doLogin():</b> Operasi ini dilakukan untuk masuk ke dalam system.	

#### 2.3.5.2 Spesific Design Class MainActivity

<b>MainActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<b>+MainActivity():</b> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
<b>+displayView():</b> Operasi ini dilakukan untuk menginisialisasi fragment-fragment yang ada.	
<b>+setTitle():</b> Operasi ini dilakukan untuk mengatur judul dari ActionBar.	

#### 2.3.5.3 Specific Design Class ProfileFragment

<b>ProfileFragment</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<b>+ProfileFragment():</b> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
<b>+setProfile():</b> Operasi ini digunakan untuk menginisialisasi nama, foto	

profil, dan total skor user.

+checkNewUser():

Operasi ini digunakan untuk mengecek apakah user adalah user baru pada aplikasi ini sekaligus menambahkan user aktif ke dalam basis data jika merupakan user baru.

#### 2.3.5.4 Specific Design Class FriendsFragment

<b>FriendsFragment</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+FriendsFragment(): Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getFriends(): Operasi ini digunakan untuk mendapatkan teman Facebook yang juga menggunakan aplikasi TandaTanya.	

#### 2.3.5.5 Specific Design Class HistoryFragment

<b>HistoryFragment</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+HistoryFragment(): Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+generateHistory(): Operasi ini dilakukan untuk mendapatkan sejarah user dan menampilkannya.	

#### 2.3.5.6 Specific Design Class AboutUsFragment

<b>AboutUsFragment</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+AboutUsFragment(): Default konstruktor, digunakan untuk inisialisasi semua	

attribute dari kelas ini.

### 2.3.5.7 Specific Design Class MapActivity

<b>MapActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+MapActivity(): Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +setUpGeoCurrentPosition(): Operasi ini digunakan untuk menginisialisasi lokasi user. +generatePlace(userid): Operasi ini digunakan untuk mengambil data tempat wisata yang ada pada basis data. +checkPointByUser(place): Operasi ini digunakan untuk memasukkan data user dan lokasi wisata ke dalam basis data sebagai <i>checkpoint</i>. +setPlaceMarker(selectedPlace, places): Operasi ini digunakan untuk menampilkan data lokasi tujuan pada peta dan sisanya.</pre>	

### 2.3.5.8 Specific Design Class QuizActivity

<b>QuizActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+QuizActivity(): Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +generateQuestion(placeid): Operasi ini digunakan untuk mengambil data pertanyaan dari basis data. +setTheQuestion(): Operasi ini digunakan untuk mengatur pertanyaan pada view. +setTheAnswer(): Operasi ini digunakan untuk mengatur jawaban pada view.</pre>	

### 2.3.5.9 Specific Design Class QuizResultActivity

<b>QuizResultActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+QuizResultActivity(): Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +setQuizPoint(): Operasi ini digunakan untuk mengatur perolehan point terhadap sebuah kuis.</pre>	

### 2.3.5.10 Specific Design Class PlaceActivity

<b>PlaceActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+PlaceActivity(): Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +generatePlaceProfile(): Operasi ini digunakan untuk menginisialisasi profil dari sebuah tempat wisata.</pre>	

### 2.3.5.11 Specific Design Class UserController

<b>UserController</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+getUserInfoById(): Operasi ini digunakan untuk mengambil data user dari basis data. +checkNewUser(): Operasi ini digunakan untuk mengecek apakah user adalah user baru pada aplikasi ini sekaligus menambahkan user aktif ke dalam basis data jika merupakan user baru. +getFriends(): Operasi ini digunakan untuk mendapatkan teman Facebook yang juga menggunakan aplikasi TandaTanya.</pre>	

### 2.3.5.12 Specific Design Class PlaceController

<b>PlaceController</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre>+get_place_selected(userid, mapAct)</pre> <p>Operasi ini digunakan untuk mendapatkan lokasi tempat wisata yang terpilih secara acak oleh aplikasi dan lokasi yang sisanya dari basis data.</p> <pre>+get_place_info_by_id(placeid, placeAct)</pre> <p>Operasi ini digunakan untuk mengambil data tempat wisata berdasarkan placeid.</p>	

### 2.3.5.13 Specific Design Class QuizController

<b>QuizController</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+get_all_answer_by_questionid(questioned, quizAct)</pre> <p>Operasi ini digunakan untuk mendapatkan jawaban-jawaban yang ada pada pertanyaan tertentu.</p> <pre>+get_all_question_by_placeid(placeid, quizAct)</pre> <p>Operasi ini digunakan untuk mendapatkan seluruh pertanyaan berdasarkan lokasi.</p> <pre>+finishing_quiz(userid, placeskor, placeid)</pre> <p>Operasi ini digunakan untuk menambahkan user id, skor, dan place id pada basis data ketika user telah menyelesaikan sebuah kuis.</p>	

### 2.3.5.14 Specific Design Class BadgeController

<b>BadgeController</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+accumulateBadges(userid, quizResultAct)</pre> <p>Operasi ini digunakan untuk mengakumulasi <i>badge</i> ketika user telah menyelesaikan sebuah kuis.</p>	

#### 2.3.5.15 Specific Design Class CheckPointController

CheckPointController	<<control>>
+insert_checkpoint(checkpoint) Operasi ini digunakan untuk menambahkan data <i>checkpoint</i> pada basis data ketika user memulai bermain kuis.	

#### 2.3.5.16 Specific Design Class HistoryController

HistoryController	<<control>>
+get_all_history_by_userid(userid, historyFragment(): Operasi ini digunakan untuk mendapatkan data sejarah pada user tertentu dari basis data.	

#### 2.3.5.17 Specific Design Class UserModel

UserModel	<<Model>>
+getUserInfoById(): Operasi ini digunakan untuk mengambil data user dari basis data.  +checkNewUser(): Operasi ini digunakan untuk mengecek apakah user adalah user baru pada aplikasi ini sekaligus menambahkan user aktif ke dalam basis data jika merupakan user baru.  +getFriends(): Operasi ini digunakan untuk mendapatkan teman Facebook yang juga menggunakan aplikasi TandaTanya.	

#### 2.3.5.18 Specific Design Class PlaceModel

PlaceModel	<<Model>>



<pre>+get_place_selected(userid, mapAct)</pre> <p>Operasi ini digunakan untuk mendapatkan lokasi tempat wisata yang terpilih secara acak oleh aplikasi dan lokasi yang sisanya dari basis data.</p> <pre>+get_place_info_by_id(placeid, placeAct)</pre> <p>Operasi ini digunakan untuk mengambil data tempat wisata berdasarkan placeid.</p>
--

### 2.3.5.19 Specific Design Class QuizModel

<b>QuizModel</b>	<b>&lt;&lt;Model&gt;&gt;</b>
<pre>+get_all_answer_by_questionid(questioned, quizAct)</pre> <p>Operasi ini digunakan untuk mendapatkan jawaban-jawaban yang ada pada pertanyaan tertentu.</p> <pre>+get_all_question_by_placeid(placeid, quizAct)</pre> <p>Operasi ini digunakan untuk mendapatkan seluruh pertanyaan berdasarkan lokasi.</p> <pre>+finishing_quiz(userid, placeskor, placeid)</pre> <p>Operasi ini digunakan untuk menambahkan user id, skor, dan place id pada basis data ketika user telah menyelesaikan sebuah kuis.</p>	

### 2.3.5.20 Specific Design Class BadgeModel

<b>BadgeModel</b>	<b>&lt;&lt;Model&gt;&gt;</b>
<pre>+accumulateBadges(userid, quizResultAct)</pre> <p>Operasi ini digunakan untuk mengakumulasi <i>badge</i> ketika user telah menyelesaikan sebuah kuis.</p>	

### 2.3.5.21 Specific Design Class CheckPointModel

<b>CheckPointModel</b>	<b>&lt;&lt;Model&gt;&gt;</b>
------------------------	------------------------------

+insert_checkpoint(checkpoint)
Operasi ini digunakan untuk menambahkan data <i>checkpoint</i> pada basis data ketika user memulai bermain kuis.

#### 2.3.5.22 Specific Design Class HistoryModel

HistoryModel	<<Model>>
+get_all_history_by_userid(userid, historyFragment():	
Operasi ini digunakan untuk mendapatkan data sejarah pada user tertentu dari basis data.	

#### 2.3.6 Class Diagram Specific Description TandaTanya (Web)

##### 2.3.6.1 Specific Design Class Login

Login	<<boundary>>

##### 2.3.6.2 Specific Design Class Home

Home	<<boundary>>

##### 2.3.6.3 Specific Design Class category\_manager

Category_manager	<<boundary>>

##### 2.3.6.4 Specific Design Class place\_manager

Place_manager	<<boundary>>
---------------	--------------


#### 2.3.6.5 Specific Design Class badge\_manager

badge_manager	<<boundary>>

#### 2.3.6.6 Specific Design Class answer\_view

Answer_view	<<boundary>>

#### 2.3.6.7 Specific Design Class tandmin

tandmin	<<control>>
<pre>+home(): Operasi ini digunakan untuk menampilkan halaman beranda beserta data tempat wisata dan pertanyaan. +photo_manager(place_id): Operasi ini digunakan untuk menampilkan halaman pengelolaan foto beserta data foto yang berdasarkan tempat wisata. +upload_file(): Operasi ini digunakan untuk upload file foto ke dalam basis data. +delete_photo(): Operasi ini digunakan untuk menghapus file foto dari basis data. +show_answer(question_id): Operasi ini digunakan untuk menampilkan halaman pengelolaan data jawaban beserta data jawaban yang berdasarkan data pertanyaan. +get_question_by_id():</pre>	

Program Studi Teknik Informatika	DPPL-TandaTanya	45/ 85
----------------------------------	-----------------	--------

Operasi ini digunakan untuk mendapatkan pertanyaan berdasarkan id.

+insert\_new\_question():

Operasi ini digunakan untuk menambahkan data pertanyaan baru ke dalam basis data.

+insert\_new\_answer():

Operasi ini digunakan untuk menambahkan data jawaban baru ke dalam basis data.

+edit\_question():

Operasi ini digunakan untuk merubah data pertanyaan yang ada di basis data.

+delete\_question():

Operasi ini digunakan untuk menghapus data pertanyaan yang ada di basis data.

+edit\_answer():

Operasi ini digunakan untuk merubah data jawaban yang ada di basis data.

+delete\_answer():

Operasi ini digunakan untuk menghapus data jawaban yang ada di basis data.

+category\_manager():

Operasi ini digunakan untuk menampilkan halaman pengelolaan data kategori tempat wisata beserta data kategori tempat wisata.

+insert\_new\_category():

Operasi ini digunakan untuk menambahkan data kategori baru ke dalam basis data.

+edit\_category():

Operasi ini digunakan untuk merubah data kategori yang ada di dalam basis data.

+delete\_category():

Operasi ini digunakan untuk menghapus data kategori yang ada di dalam basis data.

+place\_manager():

Operasi ini digunakan untuk menampilkan halaman pengelolaan tempat wisata beserta data tempat wisata yang ada di basis data.

+insert\_new\_place():

Operasi ini digunakan untuk menambahkan data tempat wisata baru ke dalam basis data.

+edit\_place():

Operasi ini digunakan untuk merubah data tempat wisata yang ada di basis data.

+delete\_place():

Operasi ini digunakan untuk menghapus data tempat wisata yang ada di basis data.

+badge\_manager():

Operasi ini digunakan untuk menampilkan halaman pengelolaan data *badge* beserta data *badge* yang ada di dalam basis data.

+insert\_new\_badges():

Operasi ini digunakan untuk menambahkan data *badge* baru ke dalam basis data.

+upload\_file\_badges():

Operasi ini digunakan untuk *upload* ikon *badge* ke dalam basis data.

+edit\_badges():

Operasi ini digunakan untuk merubah data *badge* yang ada di dalam basis data.

+upload\_file\_badges\_edit():

Operasi ini digunakan untuk menghapus dan menambahkan ikon *badge* baru ke dalam basis data.

### 2.3.6.8 Specific Design Class tandatanya\_model

Tandatanya_model	<<model>>
+get_all_category():	
Operasi ini digunakan untuk mengambil semua data kategori	

yang ada di dalam basis data.

+insert\_new\_category(category):

Operasi ini digunakan untuk menambahkan data kategori baru ke dalam basis data.

+get\_category\_last\_id():

Operasi ini digunakan untuk mendapatkan id pada baris paling bawah table kategori.

+edit\_category(value, id):

Operasi ini digunakan untuk merubah data kategori yang ada di dalam basis data.

+check\_del\_valid\_category(id):

Operasi ini digunakan untuk mengecek aman tidaknya proses penghapusan data kategori.

+get\_question\_by\_id():

Operasi ini digunakan untuk mendapatkan data kategori berdasarkan id.

+delete\_category(id):

Operasi ini digunakan untuk menghapus data kategori berdasarkan id pada basis data.

+get\_all\_places():

Operasi ini digunakan untuk mengambil semua data tempat wisata yang ada di basis data.

+get\_place\_by\_id(placeid):

Operasi ini digunakan untuk mendapatkan data tempat wisata berdasarkan id.

+get\_json\_place\_by\_id(placeid):

Operasi ini digunakan untuk mendapatkan data tempat wisata berdasarkan id yang memiliki keluaran bertipe data *array*.

+get\_all\_places\_and\_category():

Operasi ini digunakan untuk mendapatkan semua data tempat beserta data kategori

+insert\_new\_place(place):

Operasi ini digunakan untk menambahkan data tempat wisata baru ke dalam basis data.

+get\_place\_last\_id():

Operasi ini digunakan untuk mendapatkan id pada baris terakhir di table tempat wisata.

+edit\_place(place, id):

Operasi ini digunakan untuk merubah data tempat wisata berdasarkan id di basis data.

+check\_del\_valid\_place(id):

Operasi ini digunakan untuk mengecek aman tidaknya proses penghapusan data tempat wisata berdasarkan id.

+delete\_place(id):

Operasi ini digunakan untuk menghapus data tempat wisata berdasarkan id.

+get\_question\_by\_id(idicon):

Operasi ini digunakan untuk mendapatkan data pertanyaan berdasarkan id.

+insert\_new\_question\_by\_place\_id(question):

Operasi ini digunakan untuk menambahkan data pertanyaan baru ke dalam basis data berdasarkan id tempat wisata.

+get\_question\_last\_id():

Operasi ini digunakan untuk mendapatkan id pertanyaan pada baris paling bawah di dalam table pertanyaan.

+edit\_question(value, id):

Operasi ini digunakan untuk merubah data pertanyaan berdasarkan id.

+delete\_question(id):

Operasi ini digunakan untuk menghapus data pertanyaan berdasarkan id.

+get\_answer\_by\_question\_id(id):

Operasi ini digunakan untuk mendapatkan data jawaban berdasarkan id pertanyaan.

+insert\_new\_answer(answers):

Operasi ini digunakan untuk menambahkan data pertanyaan baru ke dalam basis data.

+edit\_answer(value, id):

Operasi ini digunakan untuk merubah data jawaban berdasarkan id.

+delete\_answer(id):

Operasi ini digunakan untuk menghapus data jawaban berdasarkan id.

+get\_all\_photos\_place\_id(place\_id):

Operasi ini digunakan untuk mendapatkan semua data foto berdasarkan id tempat wisata.

+insert\_photos(photos):

Operasi ini digunakan untuk menambahkan data foto baru ke dalam basis data.

+delete\_photo\_by\_id(id, name):

Operasi ini digunakan untuk menghapus data foto berdasarkan id.

+get\_photo\_by\_id(photoid):

Operasi ini digunakan untuk mendapatkan data foto berdasarkan id.

+insert\_check\_point(checkpoint):

Operasi ini digunakan untuk menambahkan data *checkpoint* baru ke dalam basis data.

+get\_check\_point\_by\_user\_id(userid):

Operasi ini digunakan untuk mendapatkan data *checkpoint* berdasarkan id user.

+delete\_check\_point\_by\_uswer\_id(userid):

Operasi ini digunakan untuk menghapus data *checkpoint* berdasarkan id user.

+insert\_history(history\_data):

Operasi ini digunakan untuk menambahkan data sejarah ke dalam basis data.

+get\_all\_history\_by\_userid(userid):

Operasi ini digunakan untuk mendapatkan semua data sejarah user berdasarkan id user.

+get\_all\_history\_and\_place\_by\_userid(userid):

Operasi ini digunakan untuk mendapatkan data sejarah dan



tempat wisata berdasarkan id user.

+get\_all\_badges():

Operasi ini digunakan untuk mendapatkan semua data badges yang ada di basis data.

+insert\_new\_badge(badges):

Operasi ini digunakan untuk menambahkan data *badge* ke dalam basis data.

+update\_badge(id, badge\_value):

Operasi ini digunakan untuk merubah data *badge* berdasarkan id.

+delete\_badge(id):

Operasi ini digunakan untuk menghapus data *badges* berdasarkan id.

+get\_badge\_by\_id(id):

Operasi ini digunakan untuk mendapatkan data *badges* berdasarkan id.

+edit\_badge(data, id):

Operasi ini digunakan untuk merubah data *badge* berdasarkan id.

+get\_all\_badges\_not\_in\_badge\_user\_by\_user\_id(userid):

Operasi ini digunakan untuk mendapatkan semua data *badge* yang tidak ada pada table *badgeuser* berdasarkan id user.

+insert\_badge\_to\_user(data):

Operasi ini digunakan untuk menambahkan data *badgeuser* baru ke dalam basis data.

+get\_category\_id\_by\_userid(userid, cat\_place\_id):

Operasi ini digunakan untuk mendapatkan data id kategori yang ada pada table *sejarah* berdasarkan id user.

+count\_total\_point\_category\_by\_user\_id(userid, cat\_place\_id):

Operasi ini digunakan untuk menghitung jumlah skor yang diperoleh user oada kategori tempat wisata tertentu.

### 3 Perancangan Data

#### 3.1 Dekomposisi Data

##### 3.1.1 Deskripsi Entitas Users

Nama	Tipe	Panjang	Keterangan
USER_ID	Varchar	100	User id, <i>Primary key</i>
USERNAME	Varchar	50	Username
TOTAL_POINT	Float	-	Total poin

##### 3.1.2 Deskripsi Entitas Category\_places

Nama	Tipe	Panjang	Keterangan
CAT_PLACE_ID	Int	-	Id kategori tempat wisata, <i>Foreign Key, Primary Key</i>
CAT_NAME	Varchar	100	Nama kategori tempat wisata

##### 3.1.3 Deskripsi Entitas Places

Nama	Tipe	Panjang	Keterangan
PLACE_ID	Integer	-	Id event, <i>Foreign Key, Primary Key</i>
CAT_PLACE_ID	Integer	-	Id kategori, <i>Foreign Key, Primary Key</i>
PLACE_NAME	Varchar	50	Nama tempat wisata
PLACE_ADDRESS	Varchar	200	Alamat tempat wisata
PLACE_DESCRIPTION	Varchar	250	Deskripsi tempat wisata
PLACE_LAT	Varchar	100	Latitude lokasi tempat wisata
PLACE_LON	Varchar	100	Longitude lokasi tempat wisata

##### 3.1.4 Deskripsi Entitas Questions

Nama	Tipe	Panjang	Keterangan
QUESTION_ID	Integer	-	Id pertanyaan, <i>Primary key, Foreign Key</i>
PLACE_ID	Integer	-	Id tempat wisata, <i>Primary key, Foreign Key</i>
QUESTION	Varchar	300	Pertanyaan

##### 3.1.5 Deskripsi Entitas Answers

Nama	Tipe	Panjang	Keterangan
------	------	---------	------------

ANSWER_ID	Integer	-	Id jawaban, <i>Primary key</i>
QUESTION_ID	Integer	-	Id pertanyaan, <i>Foreign Key</i>
ANSWER	Varchar	200	Jawaban
CORRECT	Varchar	50	Status jawaban

### 3.1.6 Deskripsi Entitas Badges

Nama	Tipe	Panjang	Keterangan
BADGE_ID	Integer	-	Id badge, <i>Primary key, Foreign Key</i>
BADGE_NAME	Varchar	50	Nama Badge
DESCRIPTION	Varchar	500	Deskripsi Badge
ICON	Varchar	500	Ikon Badge
POINT_MIN	Float	-	Poin minimal
CAT_SELECT	Integer	-	Id Kategori terpilih
CHECK_IN_COUNT	Integer	-	Jumlah kunjungan minimal

### 3.1.7 Deskripsi Entitas Badgeuser

Nama	Tipe	Panjang	Keterangan
BADGE_ID	Integer	-	Id badge user
USER_ID	Varchar	100	Id User

### 3.1.8 Deskripsi Entitas Checkpoint

Nama	Tipe	Panjang	Keterangan
CHECKPOIN_ID	Integer	-	Id <i>checkpoint</i> , <i>Primary Key</i>
USER_ID	Varchar	100	Id User, <i>Foreign Key</i>
PLACE_ID	Integer	-	Id Place, <i>Foreign Key</i>

### 3.1.9 Deskripsi Entitas History

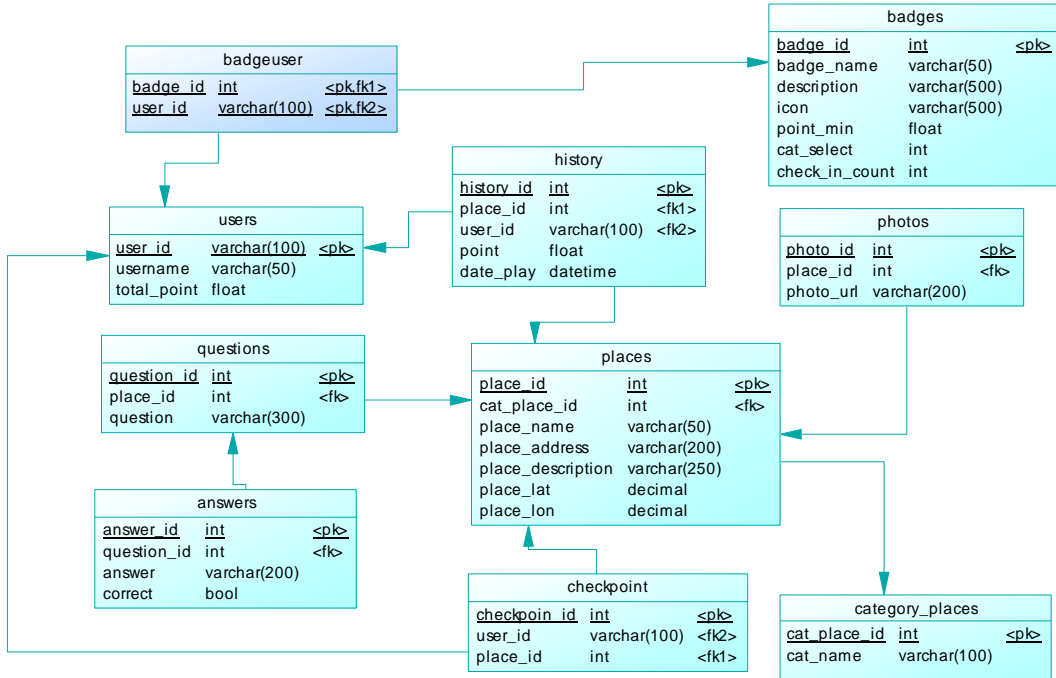
Nama	Tipe	Panjang	Keterangan
HISTORY_ID	Integer	-	Id sejarah, <i>Primary Key</i>
PLACE_ID	Integer	-	Id tempat wisata, <i>Foreign Key</i>
USER_ID	Varchar	100	Id user, <i>Foreign Key</i>
POINT	Float	-	Jumlah skor pada lokasi

### 3.1.10 Deskripsi Entitas Photos

Nama	Tipe	Panjang	Keterangan
------	------	---------	------------

PHOTO_ID	Integer	-	Id foto, <i>Primary Key</i>
PLACE_ID	Integer	-	Id tempat wisata , <i>Foreign Key</i>
PHOTO_URL	Varchar	200	Url foto

### 3.2 Physical Data Model

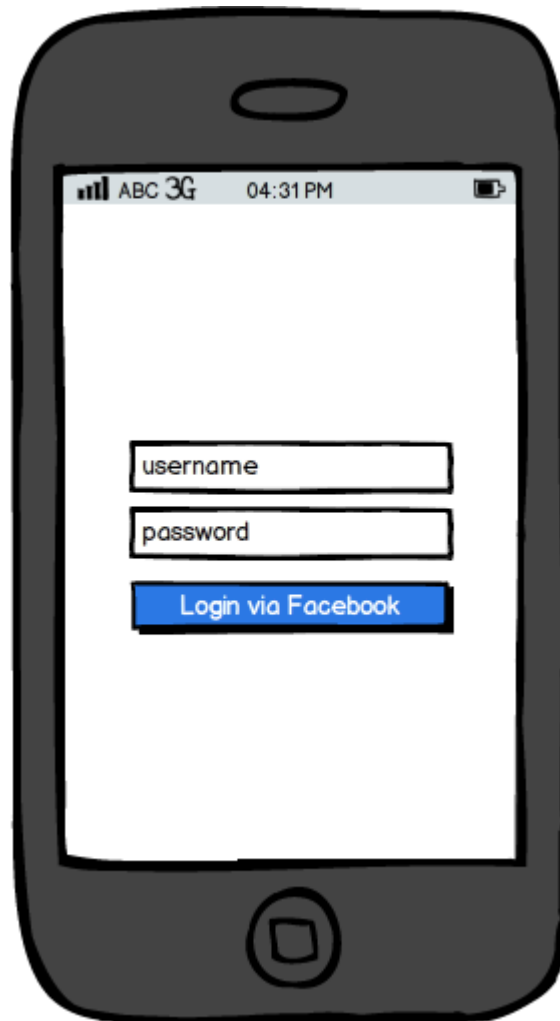


Gambar 48. Physical Data Model

## 4 Perancangan Antarmuka

### 4.1 Sketsa UI Aplikasi TandaTanya untuk User (Android)

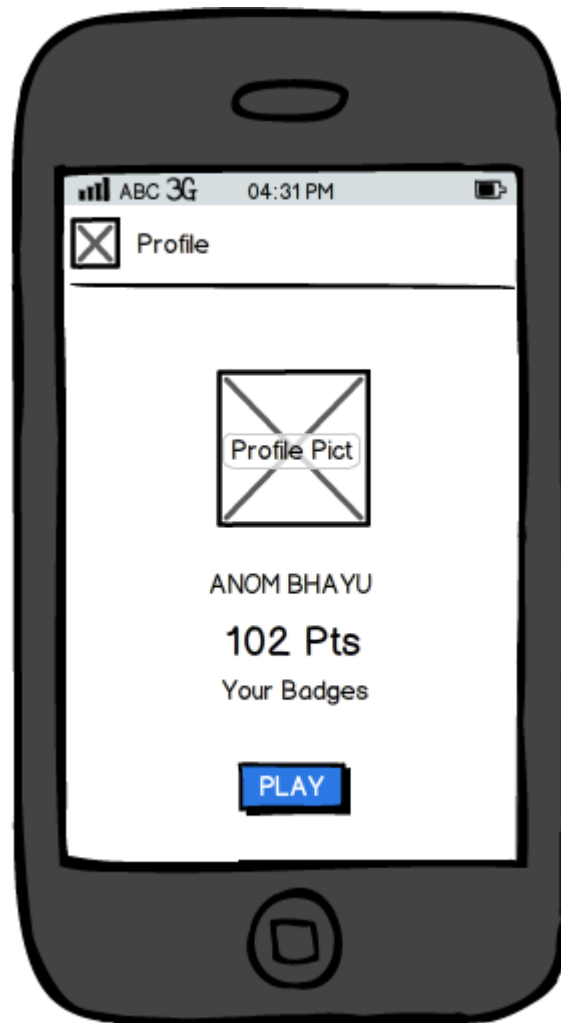
#### 4.1.1 Login



Gambar 49. Antarmuka Login

Antarmuka ini digunakan oleh User untuk masuk ke dalam aplikasi dengan menginputkan *username* dan *password* akun Facebook.

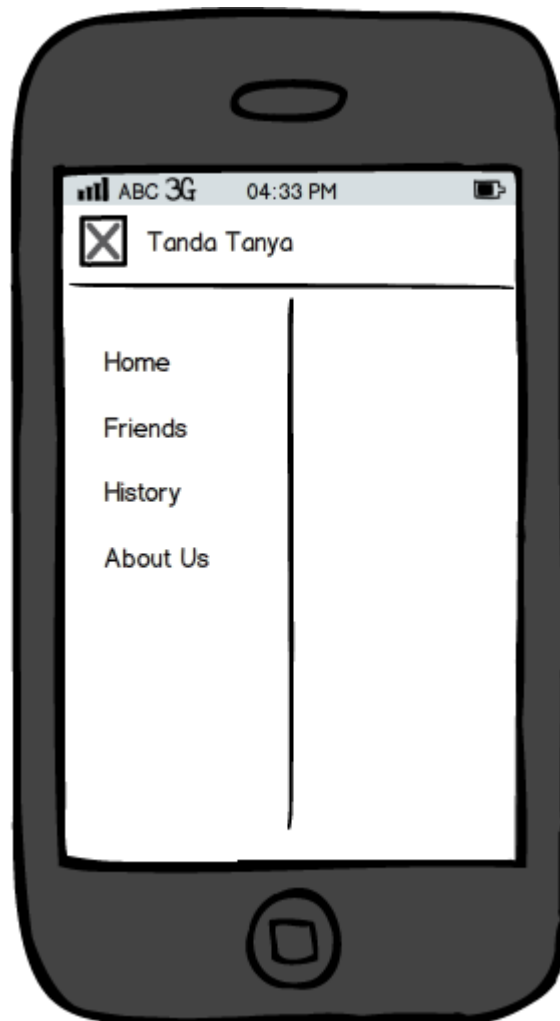
#### 4.1.2 Home



Gambar 50. Antarmuka Home

Antarmuka ini merupakan halaman utama User. Antarmuka ini menampilkan foto, nama, total skor yang dimiliki oleh user dan tombol 'PLAY' untuk bermain kuis.

#### 4.1.3 Menu Navigasi

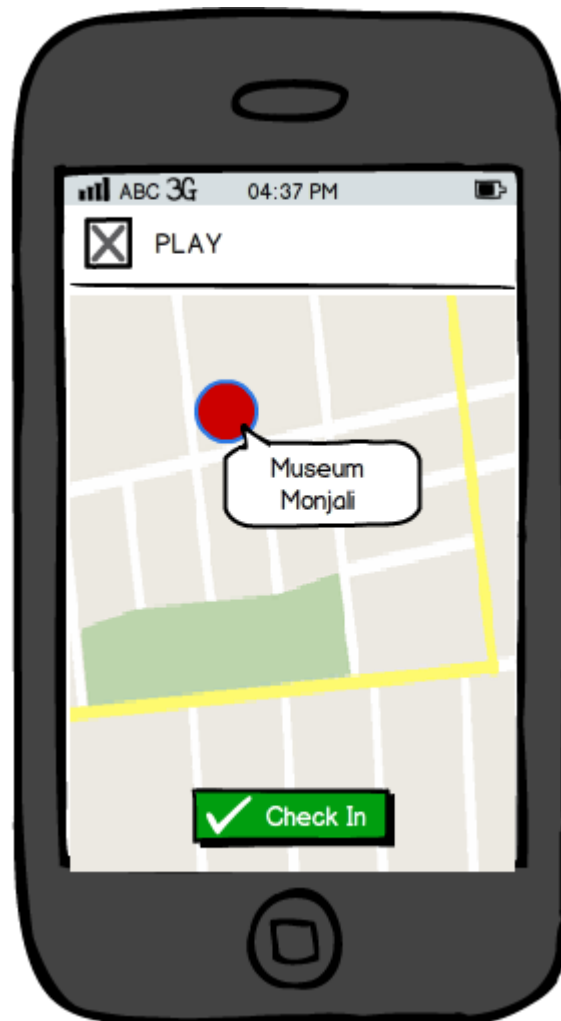


**Gambar 51. Antarmuka Navigasi**

Di halaman utama, jika User melakukan *slide* ke arah kanan, maka akan muncul antarmuka menu, dimana terdapat tombol 'Home' untuk menuju halaman utama, 'Friends' untuk menuju halaman yang menunjukkan list teman Facebook User yang menggunakan aplikasi TandaTanya, 'History' untuk menuju halaman yang menunjukkan sejarah User, dan 'About Us' yang menuju halaman tentang aplikasi TandaTanya.



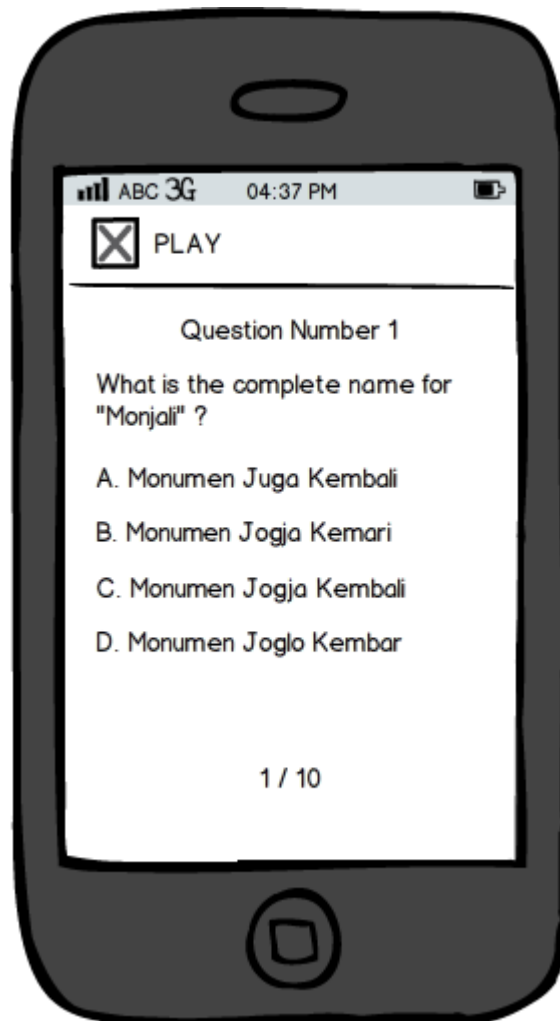
#### 4.1.4 Menampilkan Lokasi Tempat Wisata



**Gambar 52. Antarmuka Tampil Lokasi Tempat Wisata**

Antarmuka ini digunakan untuk menampilkan lokasi-lokasi tempat wisata yang ada di dalam basis data ke dalam peta berupa marker, dan juga antarmuka ini digunakan untuk menampilkan lokasi tujuan tempat wisata yang harus di tuju oleh User. Tombol 'Check In' digunakan untuk melakukan *check in* di lokasi tempat wisata dan memulai kuis.

#### 4.1.5 Kuis



**Gambar 53. Antarmuka Upload Gambar**

Antarmuka ini digunakan untuk menampilkan pertanyaan dan jawaban seputar tempat wisata sehingga User dapat mengerjakan kuis.

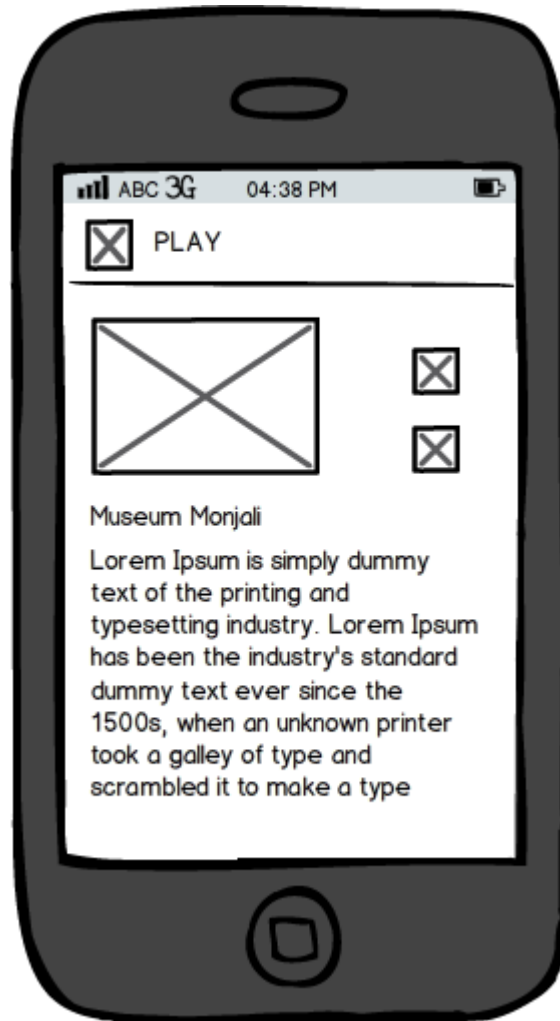
#### 4.1.6 Menampilkan Skor Kuis



**Gambar 54. Antarmuka Menampilkan Skor Kuis**

Antarmuka ini digunakan untuk menampilkan skor yang diperoleh User dalam mengerjakan kuis, dan juga User dapat segera melihat profil dari tempat yang sudah dikunjungi dengan menekan tombol 'View'.

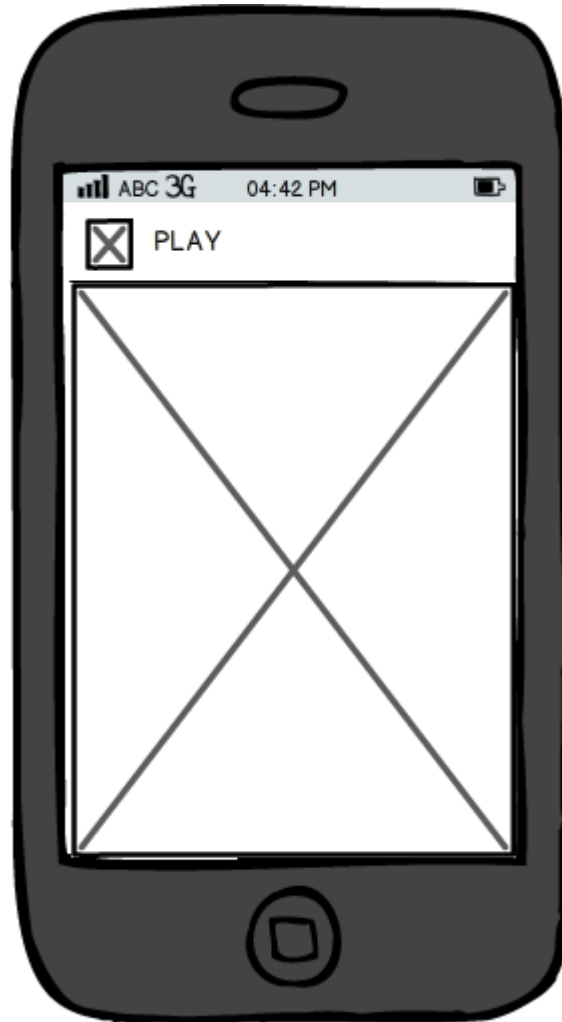
#### 4.1.7 Menampilkan Profil Tempat Wisata



**Gambar 55. Antarmuka Ubah Menampilkan Profil Tempat Wisata**

Antarmuka ini digunakan untuk menampilkan profil tempat wisata yang berupa foto, nama, dan deskripsi tempat wisata.

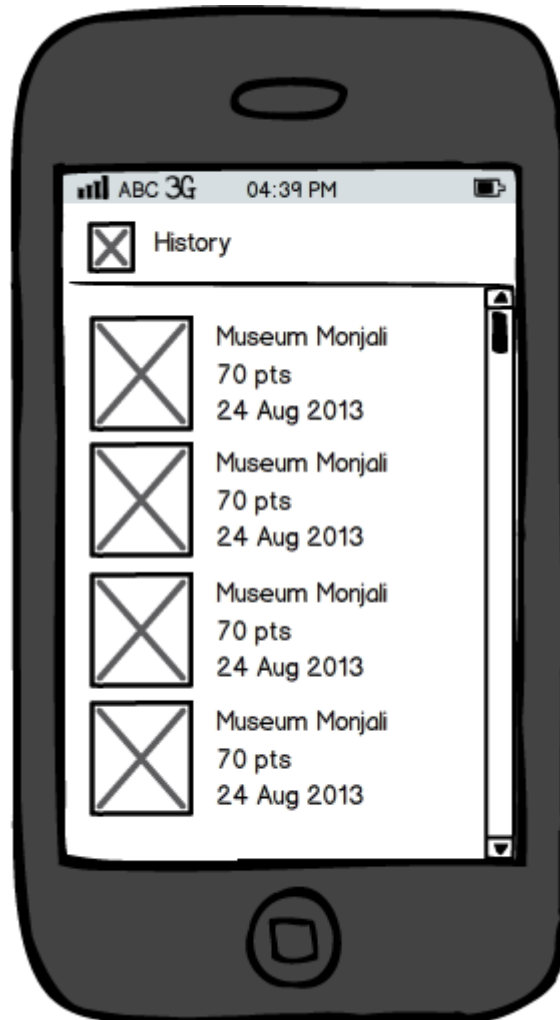
#### 4.1.8 Menampilkan Foto Galeri



**Gambar 56. Antarmuka Menampilkan Foto Galeri**

Antarmuka ini digunakan untuk menampilkan foto galari tempat wisata, user dapat menggeser layar / *slide* kanan atau kiri untuk melihat gambar berikutnya.

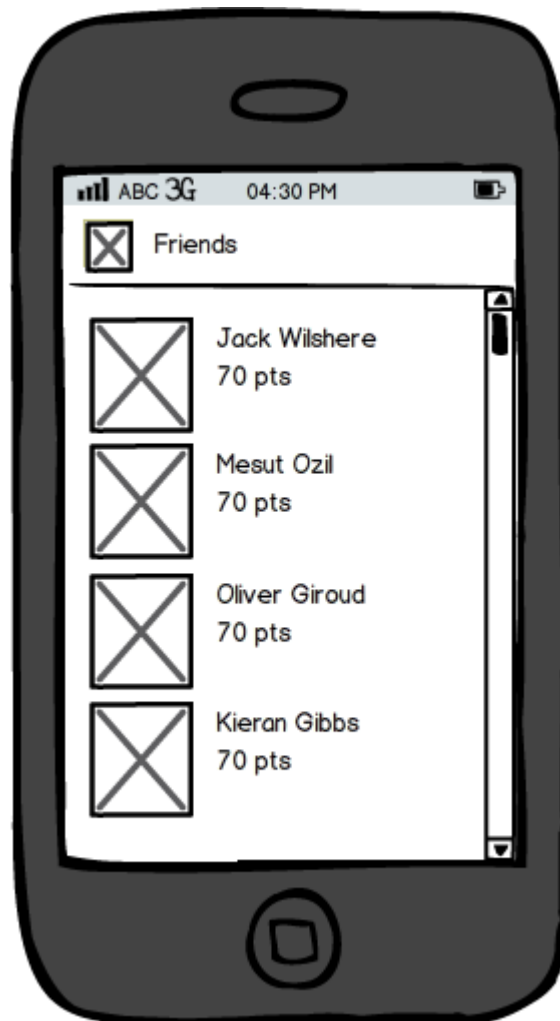
#### 4.1.9 Menampilkan Sejarah Pengguna



**Gambar 57. Antarmuka Sejarah Pengguna**

Antarmuka ini digunakan untuk menampilkan sejarah pengguna. Sejarah pengguna ditampilkan dalam bentuk list yang terdiri dari foto profil tempat wisata, nama tempat wisata, skor User ketika mengerjakan kuis pada tempat wisata tersebut dan tanggal selesai mengerjakan kuis.

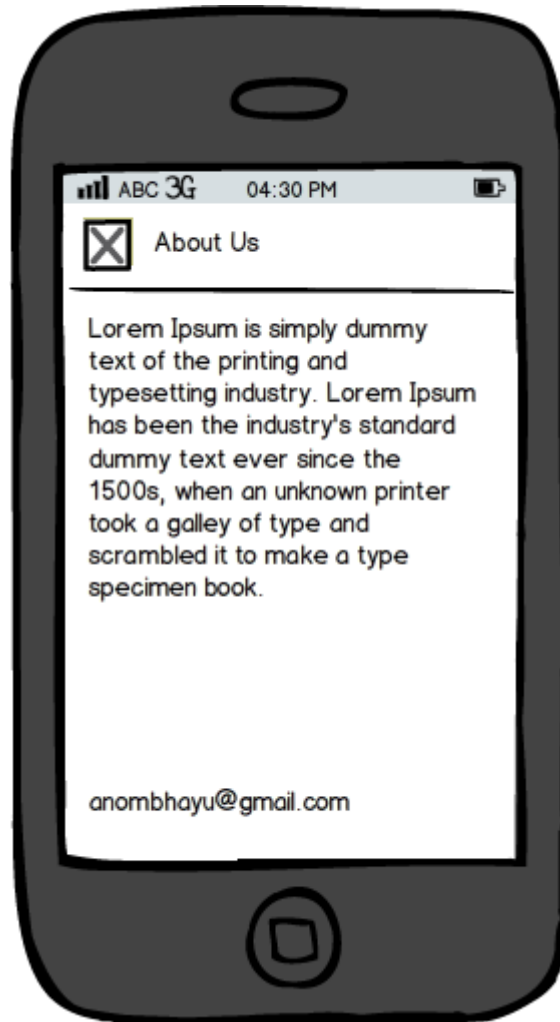
#### 4.1.10 Menampilkan Teman



**Gambar 58. Antarmuka Menampilkan Teman**

Antarmuka di atas digunakan untuk menampilkan teman User dari Facebook yang juga menggunakan aplikasi TandaTanya. Teman ditampilkan dalam bentuk list yang terdiri dari foto profil, nama, dan total skor.

#### 4.1.11 Menampilkan Tentang Aplikasi TandaTanya



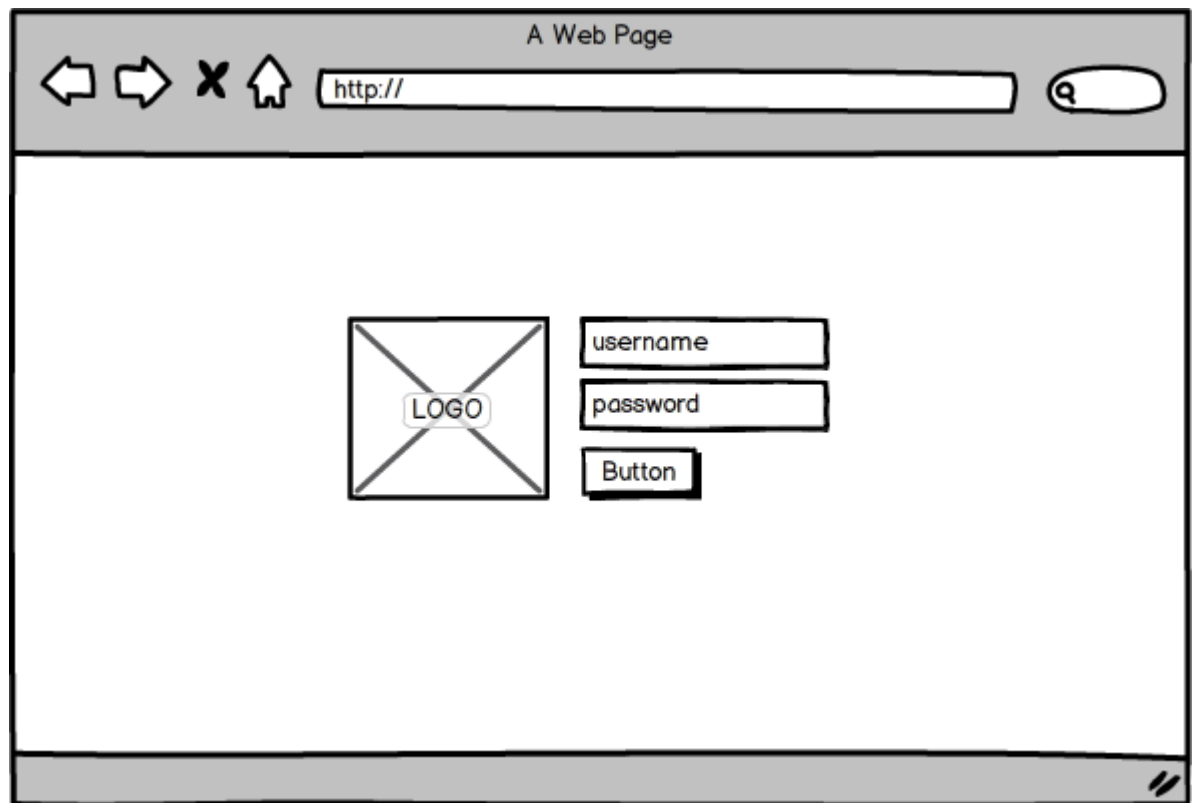
**Gambar 60. Menampilkan Tentang Aplikasi TandaTanya**

Antarmuka ini digunakan untuk menampilkan tentang aplikasi TandaTanya.



## 4.2 Sketsa UI Aplikasi TandaTanya untuk Administrator (Web)

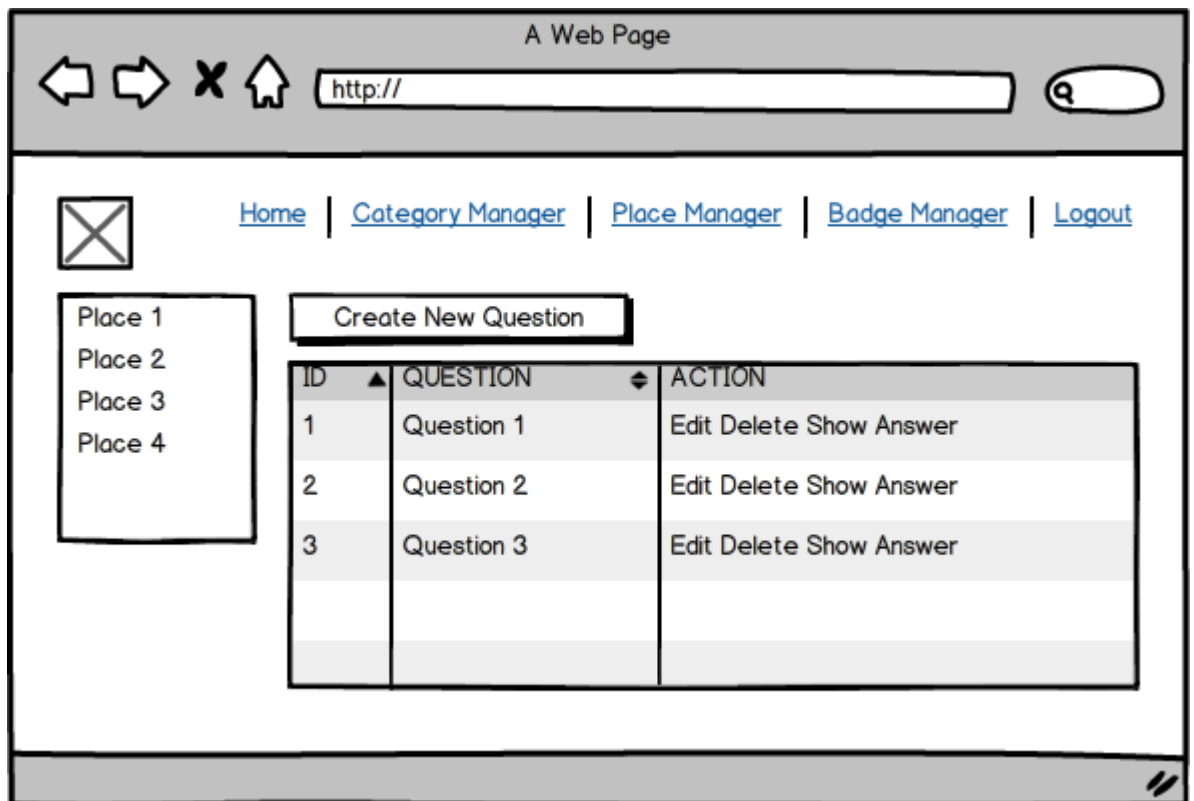
### 4.2.1 Login



Gambar 64. Antarmuka Login Web

Antarmuka ini digunakan oleh Admin untuk masuk ke dalam sistem. Admin mengisikan username dan password kemudian sistem akan melakukan validasi data.

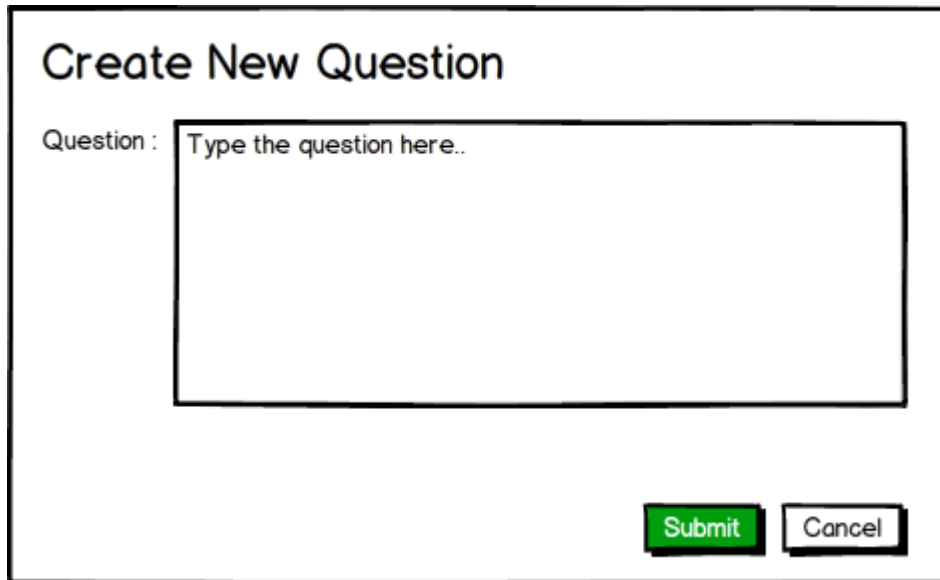
### 4.2.2 Antarmuka Kelola Data Pertanyaan



**Gambar 65. Antarmuka Kelola Data Pertanyaan**

Antarmuka ini digunakan Admin menampilkan data pertanyaan sesuai tempat wisata yang dipilih.

#### 4.2.3 Antarmuka Tambah Data Pertanyaan



**Create New Question**

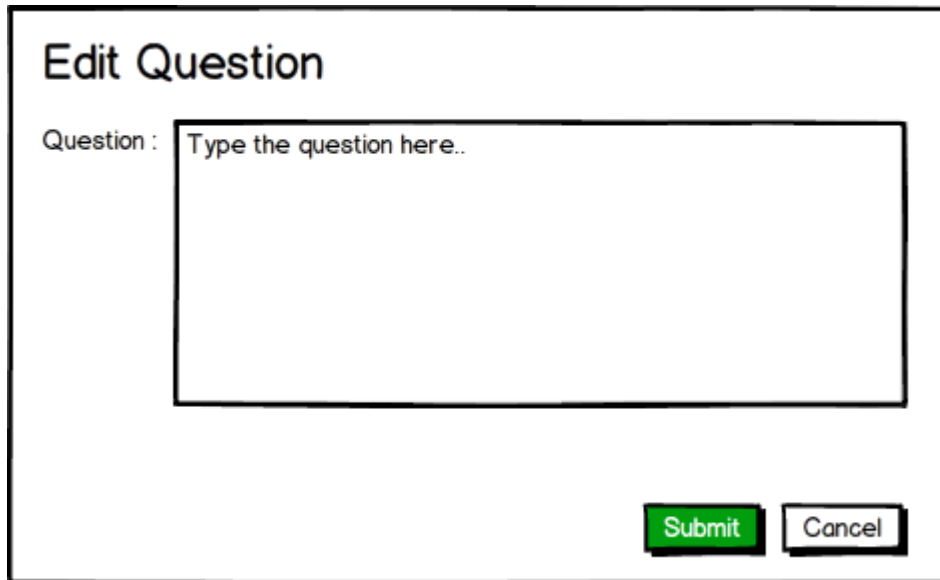
Question : Type the question here..

**Submit** **Cancel**

**Gambar 66. Antarmuka Tambah Data Pertanyaan**

Antarmuka ini digunakan oleh admin untuk menambah data pertanyaan. Administrator memasukkan data pertanyaan melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan data inputan user ke dalam basis data.

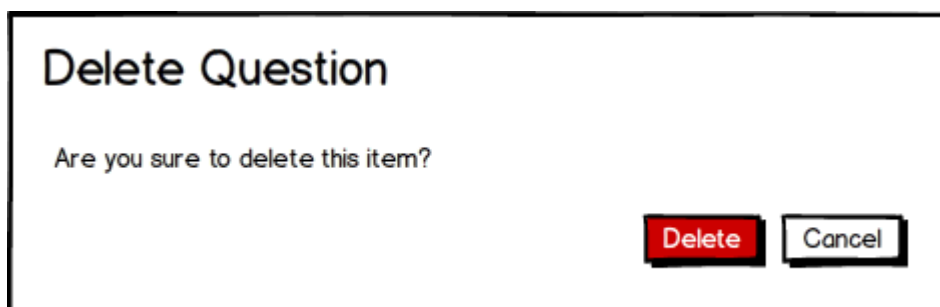
#### 4.2.4 Antarmuka Edit Data Pertanyaan



Gambar 68. Antarmuka Edit Data Pertanyaan

Antarmuka ini digunakan untuk mengubah data pertanyaan yang dipilih administrator dari table yang ditampilkan. Administrator memasukkan data pertanyaan melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan hasil rubahan data pertanyaan ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan perubahan data pertanyaan.

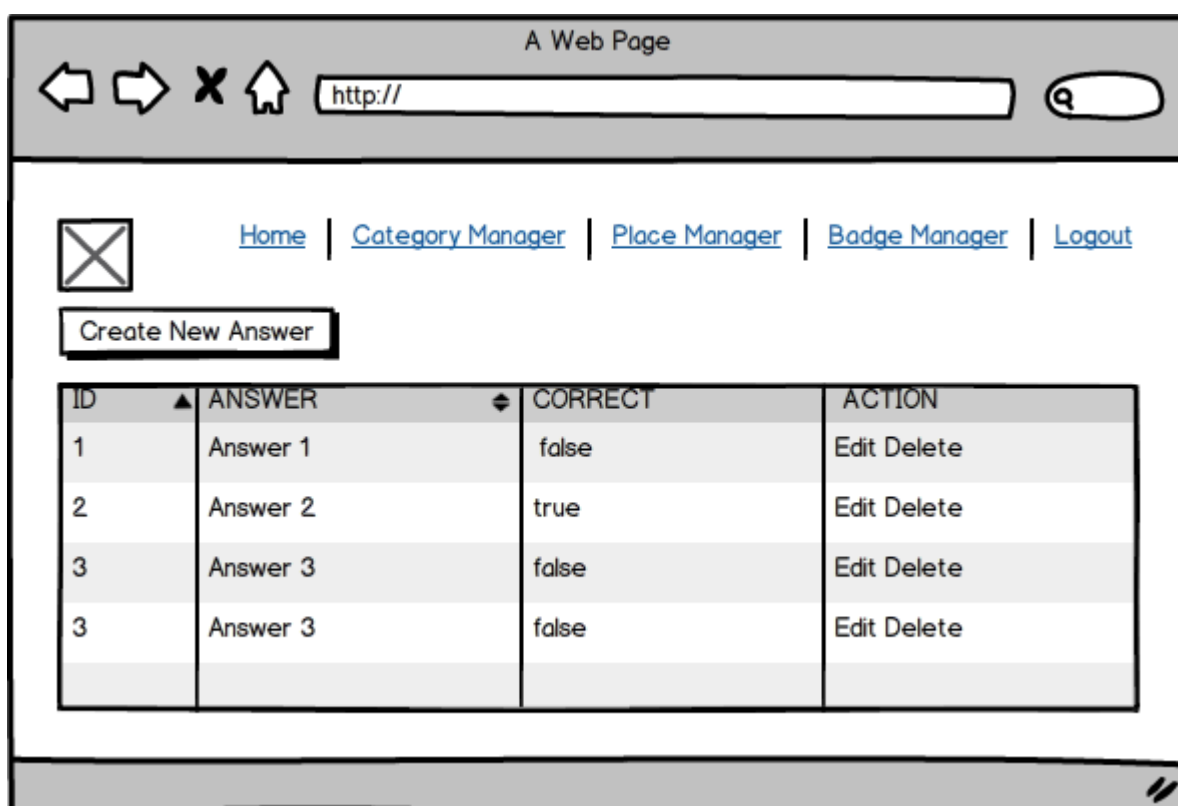
#### 4.2.5 Antarmuka Hapus Data Pertanyaan



Gambar 68. Antarmuka Hapus Data Pertanyaan

Antarmuka ini digunakan untuk menghapus data pertanyaan yang dipilih administrator dari table yang ditampilkan. Tombol 'Delete' digunakan untuk menghapus data pertanyaan dari basis data, sedangkan tombol 'Cancel' untuk membatalkan hapus data pertanyaan.

#### 4.2.6 Antarmuka Kelola Data Jawaban

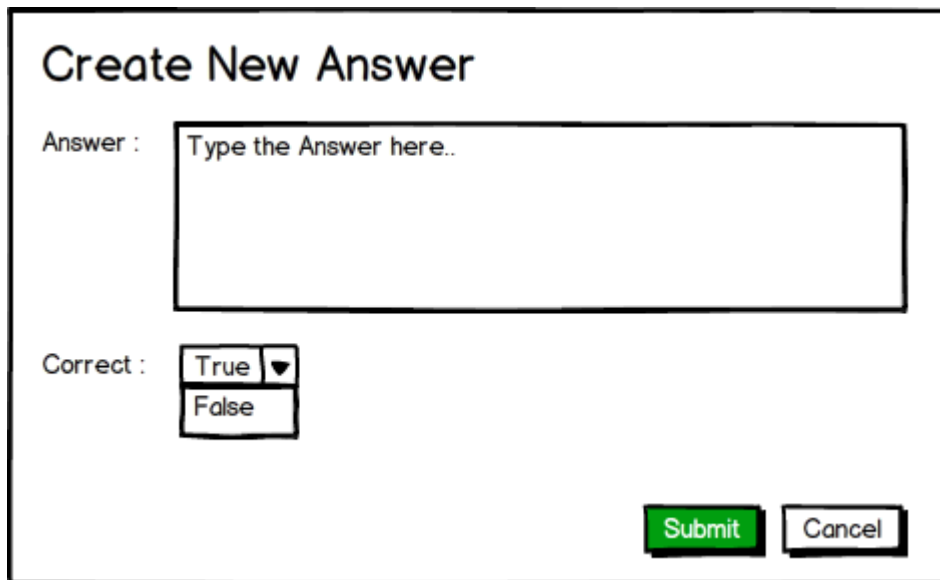


Gambar 68. Antarmuka Kelola Data Jawaban

Antarmuka ini digunakan mengelola data jawaban yang ada di basis data sesuai pertanyaan yang dipilih Administrator. Tombol 'Create New Answer' digunakan untuk menampilkan form penambahan data jawaban, 'Edit' pada kolom 'Action' digunakan untuk menampilkan form

perubahan data jawaban, dan 'Delete' pada kolom 'Action' digunakan untuk menghapus data jawaban.

#### 4.2.7 Antarmuka Tambah Data Jawaban

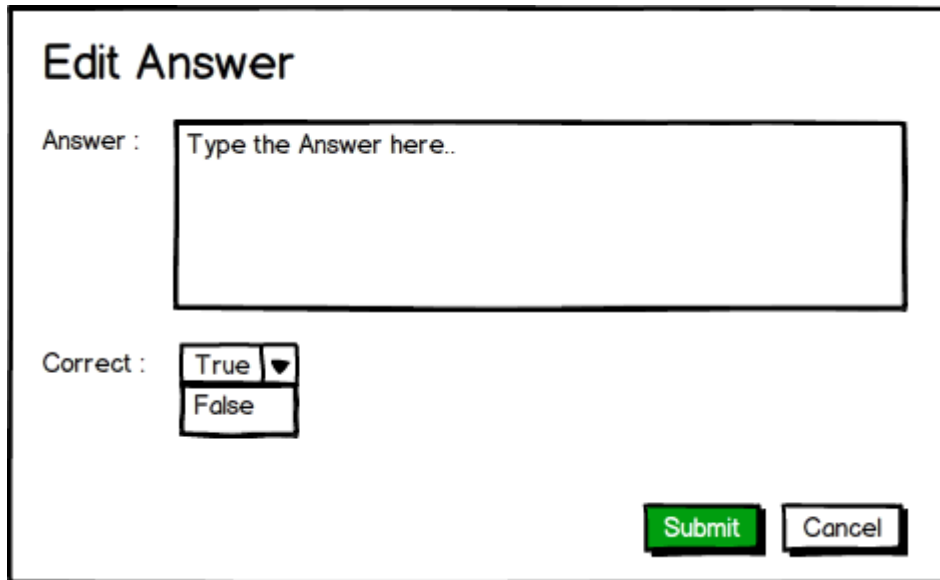


The image shows a web form titled "Create New Answer". It contains a text input field for the answer, a dropdown menu for the correctness (currently set to "True"), and two buttons: "Submit" and "Cancel".

**Gambar 68. Antarmuka Tambah Data Jawaban**

Antarmuka ini digunakan untuk menambahkan data jawaban ke dalam basis data sesuai data pertanyaan yang dipilih oleh Administrator. Administrator memasukkan data jawaban melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan data jawaban baru ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan penambahan data jawaban.

#### 4.2.8 Antarmuka Edit Data Jawaban



**Edit Answer**

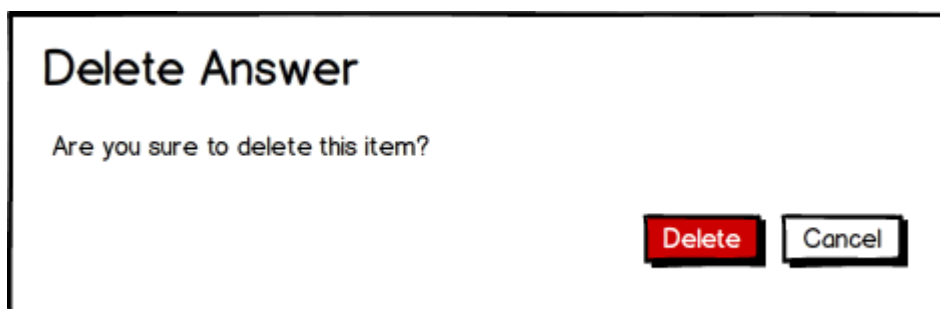
Answer :

Correct :

**Gambar 68. Antarmuka Edit Data Jawaban**

Antarmuka ini digunakan untuk mengubah data jawaban yang dipilih administrator dari table yang ditampilkan. Administrator memasukkan data jawaban melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan hasil rubahan data jawaban ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan perubahan data jawaban.

#### 4.2.9 Antarmuka Hapus Data Jawaban



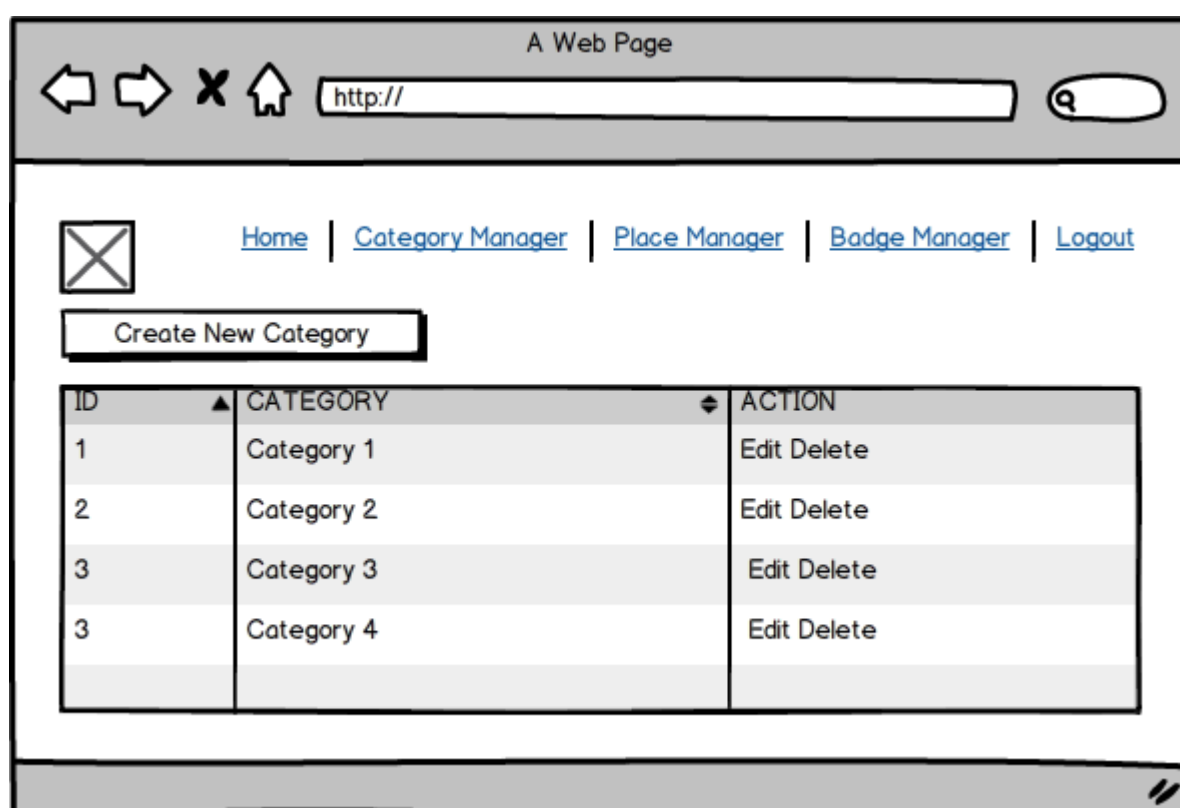
**Delete Answer**

Are you sure to delete this item?

**Gambar 68. Antarmuka Hapus Data Jawaban**

Antarmuka ini digunakan untuk menghapus data jawaban yang dipilih administrator dari table yang ditampilkan. Tombol 'Delete' digunakan untuk menghapus data jawaban dari basis data, sedangkan tombol 'Cancel' untuk membatalkan hapus data pertanyaan.

#### 4.2.10 Antarmuka Kelola Data Kategori Tempat Wisata



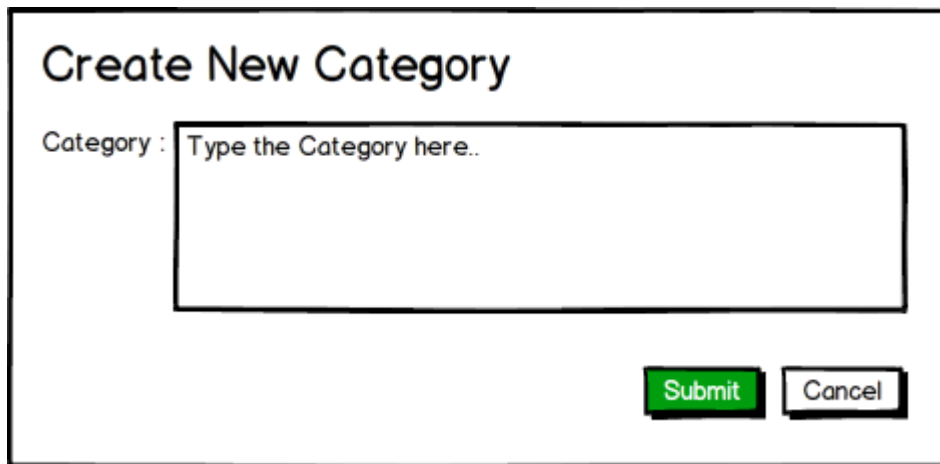
Gambar 68. Antarmuka Kelola Data Kategori Tempat Wisata

Antarmuka ini digunakan mengelola data kategori tempat wisata yang ada di basis data. Tombol 'Create New Category' digunakan untuk menampilkan form penambahan data kategori tempat wisata, 'Edit' pada kolom 'Action' digunakan untuk menampilkan form perubahan data kategori tempat wisata, dan 'Delete'



pada kolom 'Action' digunakan untuk menghapus data kategori tempat wisata.

#### 4.2.11 Antarmuka Tambah Data Kategori Tempat Wisata

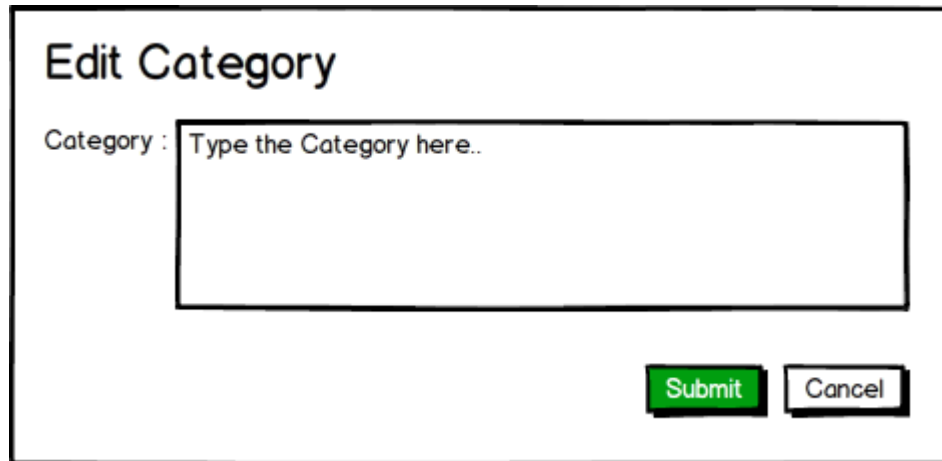


The image shows a web form titled "Create New Category". It contains a text input field with the label "Category :" and the placeholder text "Type the Category here..". Below the input field are two buttons: "Submit" and "Cancel".

Gambar 68. Antarmuka Tambah Data Kategori Tempat Wisata

Antarmuka ini digunakan untuk menambahkan data kategori tempat wisata ke dalam basis data. Administrator memasukkan data kategori tempat wisata melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan data kategori tempat wisata baru ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan penambahan data kategori tempat wisata.

#### 4.2.12 Antarmuka Edit Data Kategori Tempat Wisata



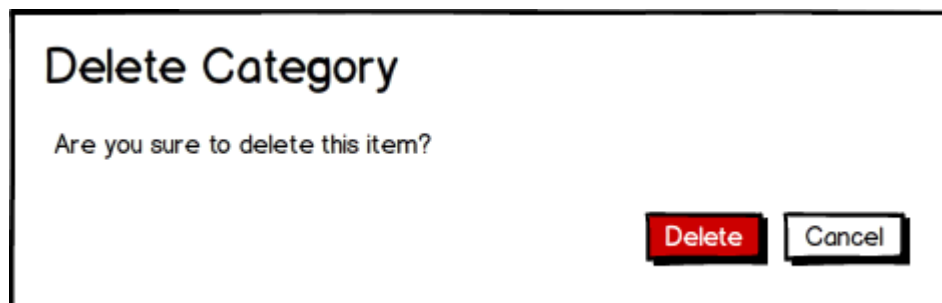
**Edit Category**

Category :

Gambar 68. Antarmuka Edit Data Kategori Tempat Wisata

Antarmuka ini digunakan untuk mengubah data kategori tempat wisata yang dipilih administrator dari table yang ditampilkan. Administrator memasukkan data kategori tempat wisata melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan hasil rubahan data kategori tempat wisata ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan perubahan data kategori tempat wisata.

#### 4.2.13 Antarmuka Hapus Data Kategori Tempat Wisata



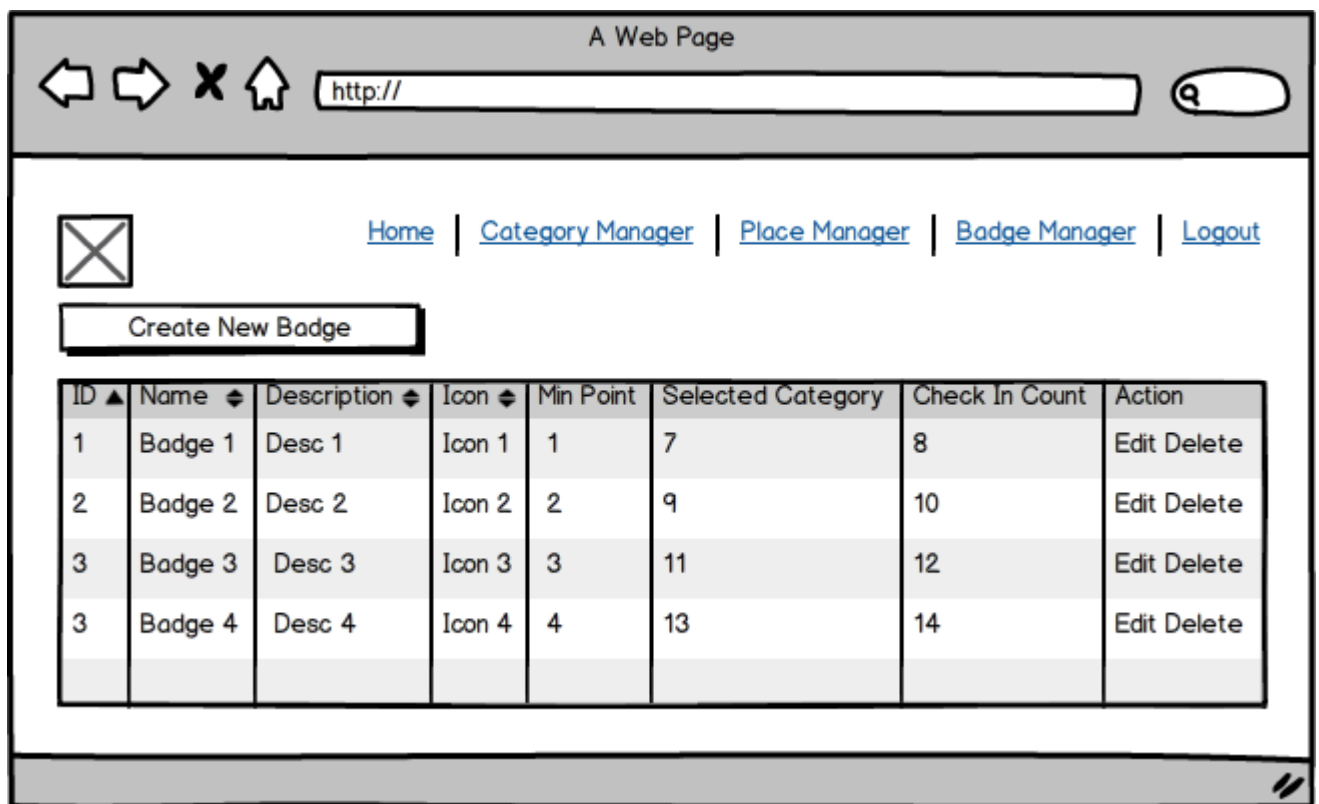
**Delete Category**

Are you sure to delete this item?

Gambar 68. Antarmuka Hapus Data Kategori Tempat Wisata

Antarmuka ini digunakan untuk menghapus data kategori tempat wisata yang dipilih administrator dari table yang ditampilkan. Tombol 'Delete' digunakan untuk menghapus data kategori tempat wisata dari basis data, sedangkan tombol 'Cancel' untuk membatalkan hapus data kategori tempat wisata.

#### 4.2.14 Antarmuka Kelola Data Badge



Gambar 68. Antarmuka Kelola Data Badge

Antarmuka ini digunakan mengelola data badge yang ada di basis data. Tombol 'Create New Badge' digunakan untuk menampilkan form penambahan data badge, 'Edit' pada kolom 'Action' digunakan untuk menampilkan form perubahan data badge, dan 'Delete' pada kolom 'Action' digunakan untuk menghapus data badge.

#### 4.2.15 Antarmuka Tambah Data Badge

The screenshot shows a web form titled "Create New Badge". It contains the following fields and controls:

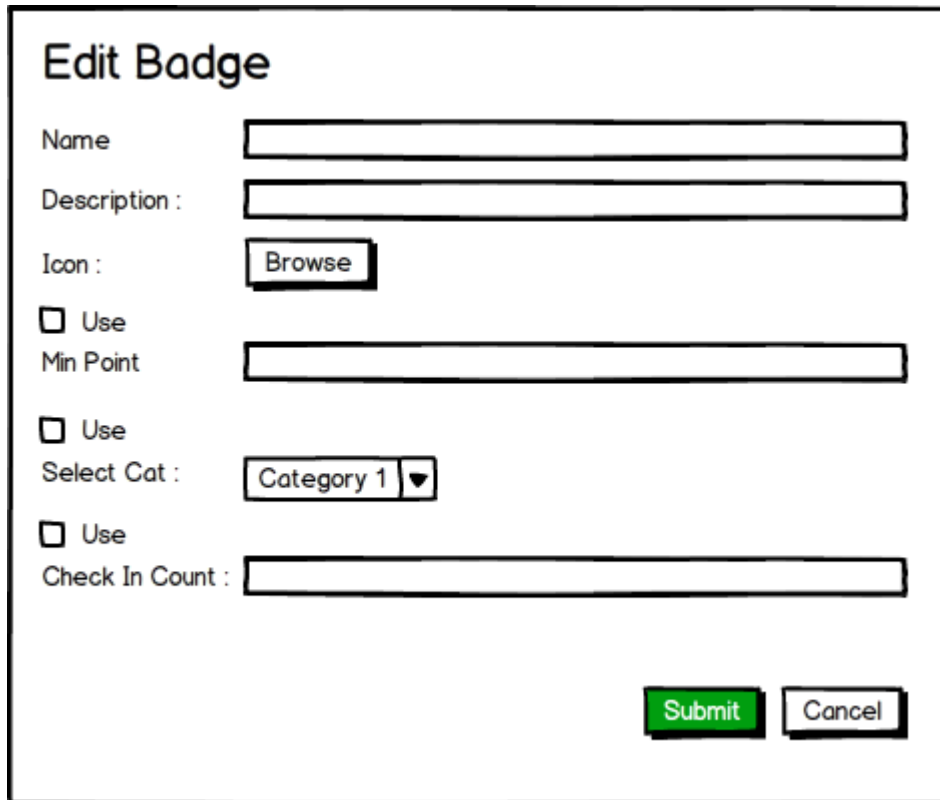
- Name:** A text input field.
- Description:** A text input field.
- Icon:** A "Browse" button.
- Use:** A checkbox.
- Min Point:** A text input field.
- Use:** A checkbox.
- Select Cat:** A dropdown menu currently showing "Category 1".
- Use:** A checkbox.
- Check In Count:** A text input field.

At the bottom right of the form, there are two buttons: a green "Submit" button and a "Cancel" button.

Gambar 68. Antarmuka Tambah Data Badge

Antarmuka ini digunakan untuk menambahkan data badge ke dalam basis data. Administrator memasukkan data badge melalui form yang ditampilkan yang terdiri dari nama badge, deskripsi badge, ikon, minimal point, kategori terpilih, jumlah Check In dan beberapa *checkbox* yang bila mana diberikan centang berarti data di bawahnya diikuti sertakan dalam penambahan badge. Tombol 'Submit' digunakan untuk menyimpan data badge baru ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan penambahan data badge.

#### 4.2.16 Antarmuka Edit Data Badge



The image shows a web form titled "Edit Badge". The form contains the following fields and controls:

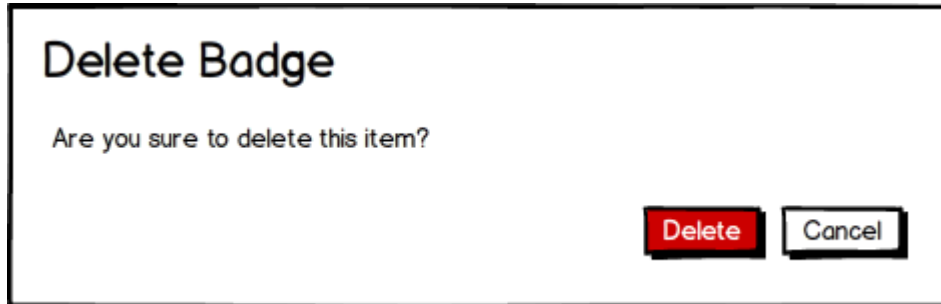
- Name:** A text input field.
- Description:** A text input field.
- Icon:** A "Browse" button.
- Use:** A checkbox.
- Min Point:** A text input field.
- Use:** A checkbox.
- Select Cat:** A dropdown menu showing "Category 1".
- Use:** A checkbox.
- Check In Count:** A text input field.

At the bottom right of the form, there are two buttons: a green "Submit" button and a "Cancel" button.

**Gambar 68. Antarmuka Edit Data Badge**

Antarmuka ini digunakan untuk mengubah data badge yang dipilih administrator dari table yang ditampilkan. Administrator memasukkan data badge melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan hasil rubahan data badge ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan perubahan data badge.

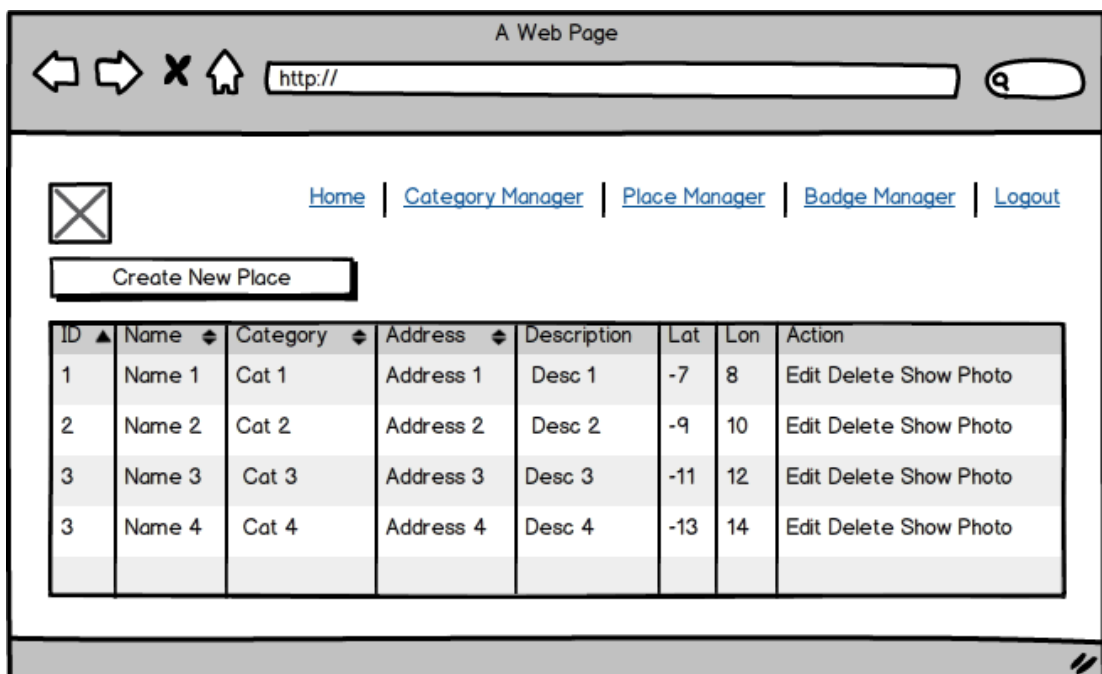
#### 4.2.17 Antarmuka Hapus Data Badge



Gambar 68. Antarmuka Hapus Data Badge

Antarmuka ini digunakan untuk menghapus data badge yang dipilih administrator dari table yang ditampilkan. Tombol 'Delete' digunakan untuk menghapus data badge dari basis data, sedangkan tombol 'Cancel' untuk membatalkan hapus data badge.

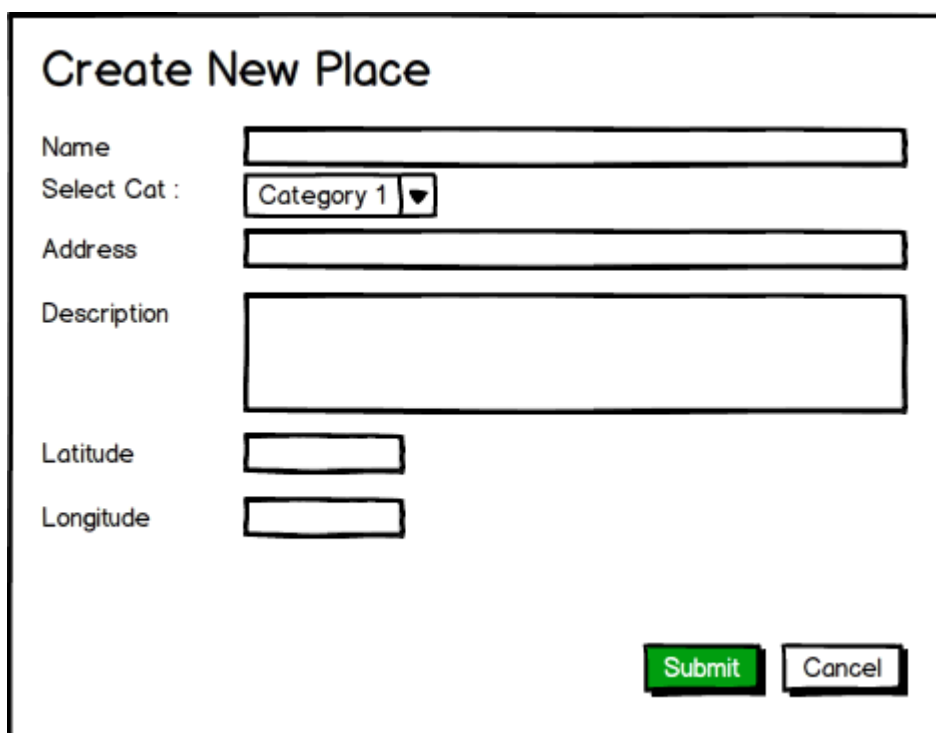
#### 4.2.18 Antarmuka Kelola Data Tempat Wisata



Gambar 68. Antarmuka Kelola Data Tempat wisata

Antarmuka ini digunakan mengelola data tempat wisata yang ada di basis data. Tombol 'Create New Place' digunakan untuk menampilkan form penambahan data tempat wisata, 'Edit' pada kolom 'Action' digunakan untuk menampilkan form perubahan data tempat wisata, dan 'Delete' pada kolom 'Action' digunakan untuk menghapus data tempat wisata.

#### 4.2.19 Antarmuka Tambah Data Tempat wisata



The image shows a web form titled "Create New Place". It contains the following fields and controls:

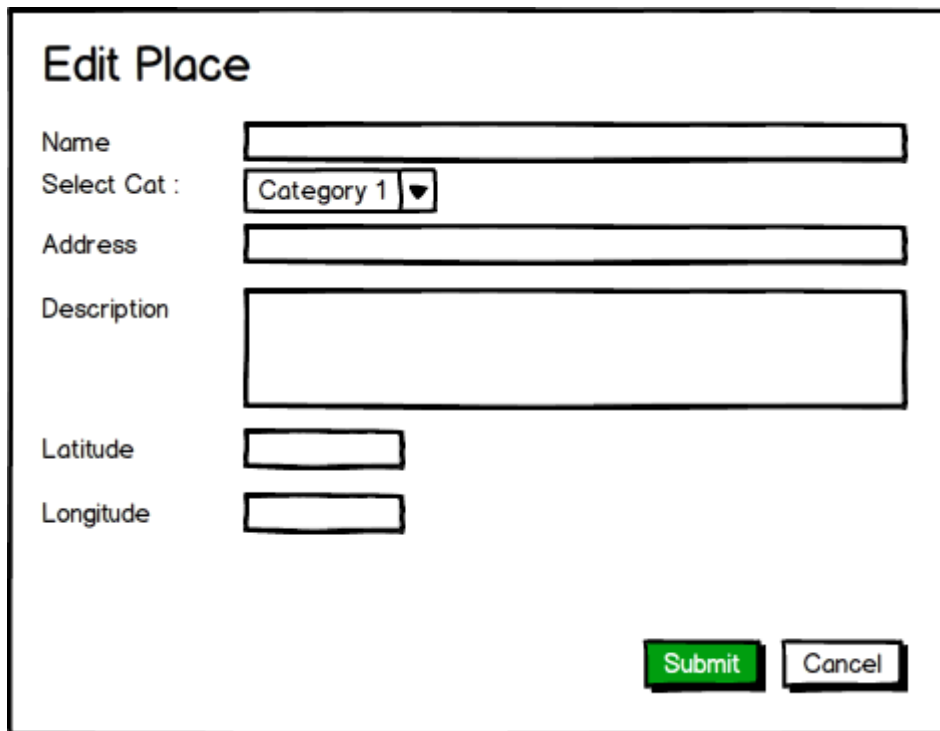
- Name:** A single-line text input field.
- Select Cat:** A dropdown menu currently displaying "Category 1".
- Address:** A single-line text input field.
- Description:** A multi-line text area.
- Latitude:** A single-line text input field.
- Longitude:** A single-line text input field.
- Buttons:** Two buttons at the bottom right: a green "Submit" button and a white "Cancel" button with a black border.

Gambar 68. Antarmuka Tambah Data Tempat wisata

Antarmuka ini digunakan untuk menambahkan data tempat wisata ke dalam basis data. Administrator memasukkan data tempat wisata melalui form yang ditampilkan yang terdiri dari nama tempat wisata, kategori, alamat, deskripsi tempat wisata, latitude dan longitude tempat wisata. Tombol 'Submit' digunakan

untuk menyimpan data tempat wisata baru ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan penambahan data tempat wisata.

#### 4.2.20 Antarmuka Edit Data Tempat wisata



**Edit Place**

Name

Select Cat :

Address

Description

Latitude

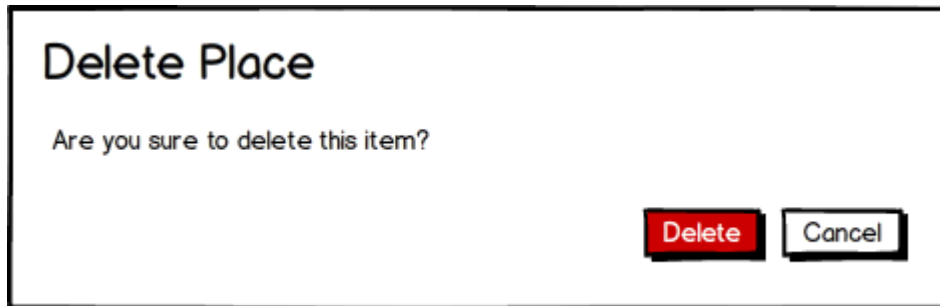
Longitude

**Gambar 68. Antarmuka Edit Data Tempat wisata**

Antarmuka ini digunakan untuk mengubah data tempat wisata yang dipilih administrator dari table yang ditampilkan. Administrator memasukkan data tempat wisata melalui form yang ditampilkan, dan tombol 'Submit' digunakan untuk menyimpan hasil rubahan data tempat wisata ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan perubahan data tempat wisata.



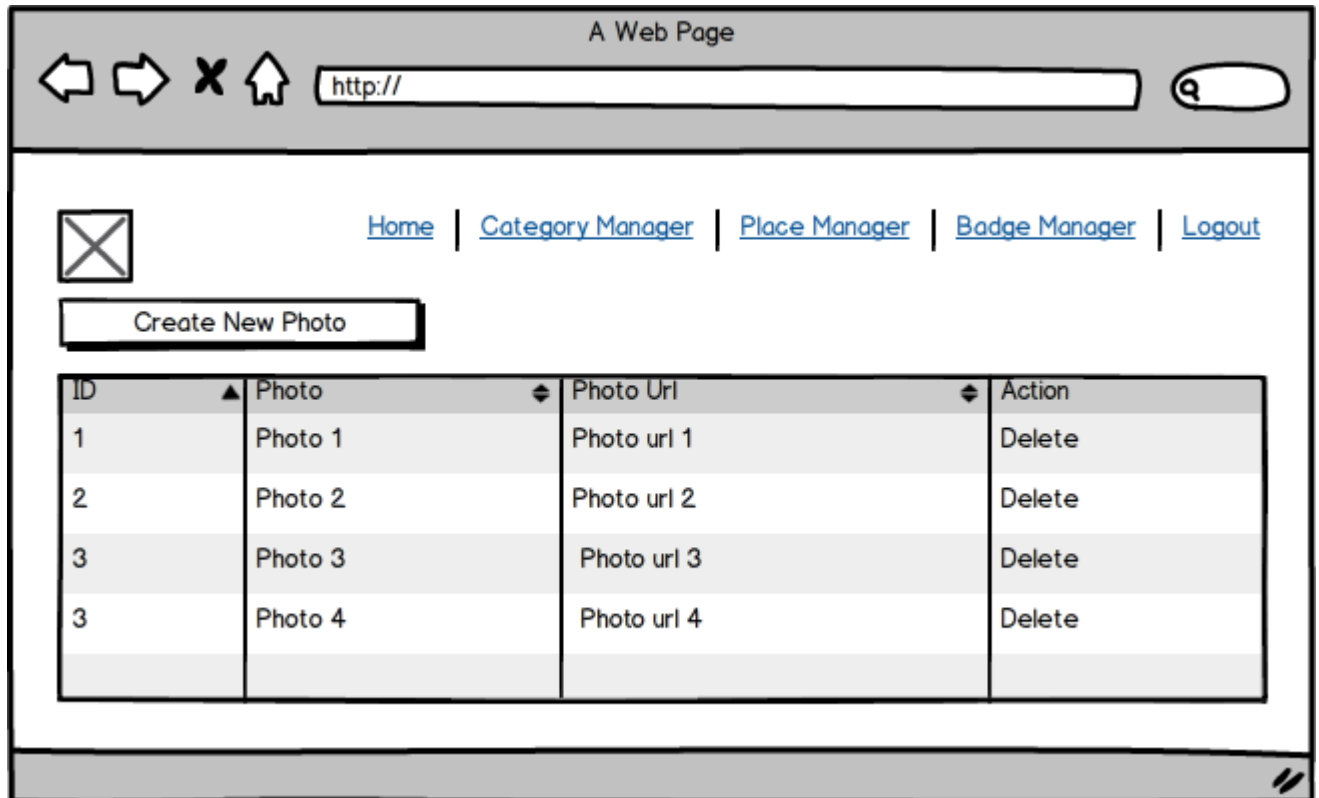
#### 4.2.21 Antarmuka Hapus Data Tempat wisata



**Gambar 68. Antarmuka Hapus Data Tempat wisata**

Antarmuka ini digunakan untuk menghapus data tempat wisata yang dipilih administrator dari table yang ditampilkan. Tombol 'Delete' digunakan untuk menghapus data tempat wisata dari basis data, sedangkan tombol 'Cancel' untuk membatalkan hapus data tempat wisata.

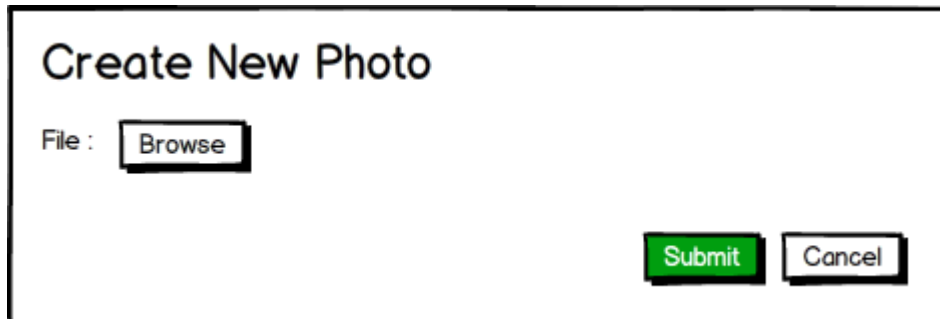
#### 4.2.22 Antarmuka Kelola Data Foto



Gambar 68. Antarmuka Kelola Data Foto

Antarmuka ini digunakan mengelola data foto yang ada di basis data. Tombol 'Create New Photo' digunakan untuk menampilkan form penambahan data foto, dan 'Delete' pada kolom 'Action' digunakan untuk menghapus data foto.

#### 4.2.23 Antarmuka Tambah Data Foto

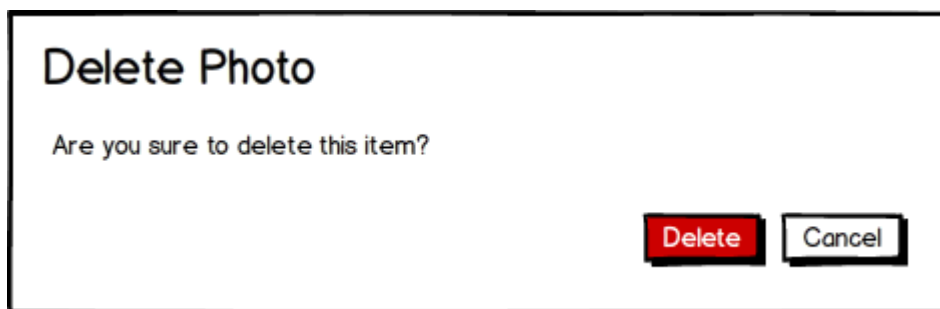


The screenshot shows a web form titled "Create New Photo". It contains a "File:" label followed by a "Browse" button. At the bottom right, there are two buttons: a green "Submit" button and a "Cancel" button.

Gambar 68. Antarmuka Tambah Data Foto

Antarmuka ini digunakan untuk menambahkan data foto ke dalam basis data. Administrator memasukkan data foto melalui form yang. Tombol 'Submit' digunakan untuk menyimpan data foto baru ke dalam basis data, sedangkan tombol 'Cancel' untuk membatalkan penambahan data foto.

#### 4.2.24 Antarmuka Hapus Data Foto



The screenshot shows a web form titled "Delete Photo". It contains the question "Are you sure to delete this item?". At the bottom right, there are two buttons: a red "Delete" button and a "Cancel" button.

Gambar 68. Antarmuka Hapus Data Foto

Antarmuka ini digunakan untuk menghapus data foto yang dipilih administrator dari table yang ditampilkan. Tombol 'Delete' digunakan untuk menghapus data foto dari basis data, sedangkan tombol 'Cancel' untuk membatalkan hapus data foto.