



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### METODE DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi Literatur

Mencari informasi atau referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Referensi ini dapat dicari melalui buku, jurnal, artikel penelitian, dan situs-situs di internet. Studi literatur dilakukan untuk memperkuat permasalahan dan sebagai dasar teori dalam melakukan studi.

2. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data-data yang akan digunakan untuk keperluan aplikasi, baik mencari melalui internet maupun bertemu para ahli.

3. Analisis Aplikasi

Pada tahap ini akan dianalisis algoritma yang digunakan, yaitu algoritma *Analytical Hierarchy Process*. Hal-hal yang akan dianalisis meliputi cara kerja algoritma dan detail kegunaan algoritma, sehingga dapat diimplementasikan dengan optimal pada aplikasi ini.

4. Perancangan Program

Perancangan program yang dilakukan meliputi *interface*, *database*, dan alur kerja program. Akan disertai juga berbagai diagram seperti *Flowchart Diagram*, *Data Flow Diagram*, dan *Entity Relationship Diagram*.

### 5. Pembuatan Program

Pada tahap ini dilakukan pembuatan program dengan mengaplikasikan rancangan yang telah dibuat sebelumnya dengan bahasa pemrograman dan *tool* yang dipilih.

### 6. *Testing* dan Perbaiki Program

Pada tahap ini akan dilakukan *testing* pada program yang telah dibuat. *Testing* dilakukan oleh *user* secara umum dari berbagai kalangan. Apabila ada kesalahan atau *error* yang terdeteksi, maka program akan dianalisis kembali dan diperbaiki.

### 7. Penulisan Skripsi

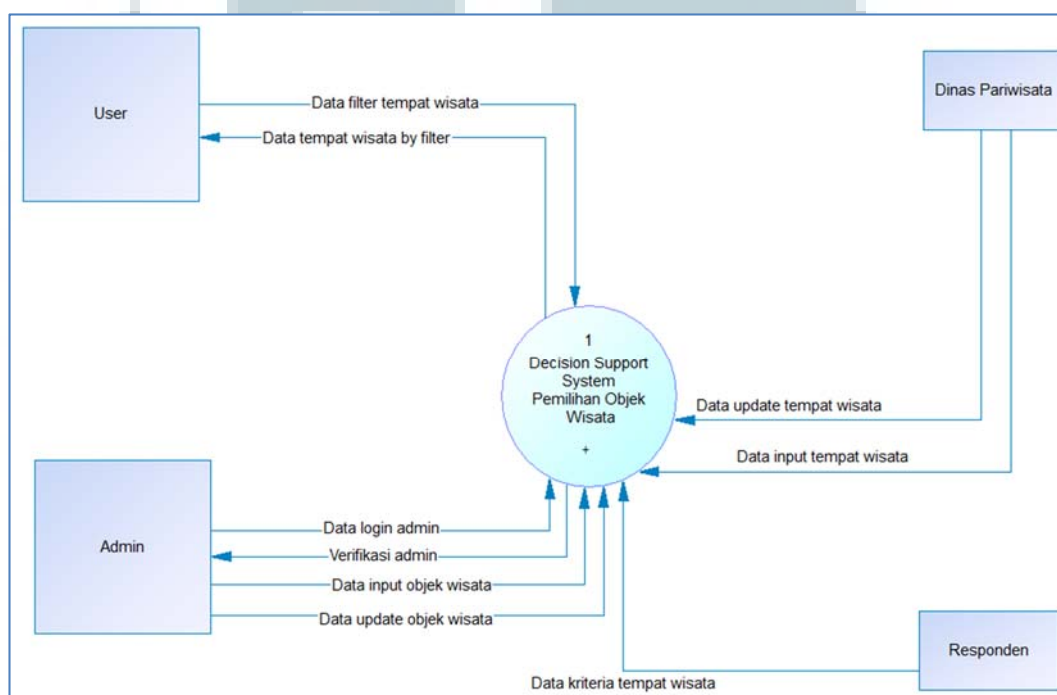
Setelah semua langkah-langkah di atas telah selesai, maka langkah selanjutnya adalah penyusunan laporan skripsi sebagai dokumentasi.

## 3.2 Perancangan Aplikasi

Dalam pembuatan aplikasi, digunakan *Data Flow Diagram*, *Flowchart Diagram* dan *Entity Relationship Diagram*. *Data Flow Diagram* adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. *Flowchart Diagram* adalah sebuah diagram yang menggambarkan algoritma atau alur jalan sebuah proses dengan proses lainnya yang terjadi dalam aplikasi tersebut hingga dihasilkan sebuah solusi atas permasalahan tersebut. *Entity Relationship Diagram* adalah sebuah diagram yang menggambarkan struktur dan hubungan antar tabel, data, dan informasi yang ada pada sebuah *relational database* yang digunakan oleh aplikasi.

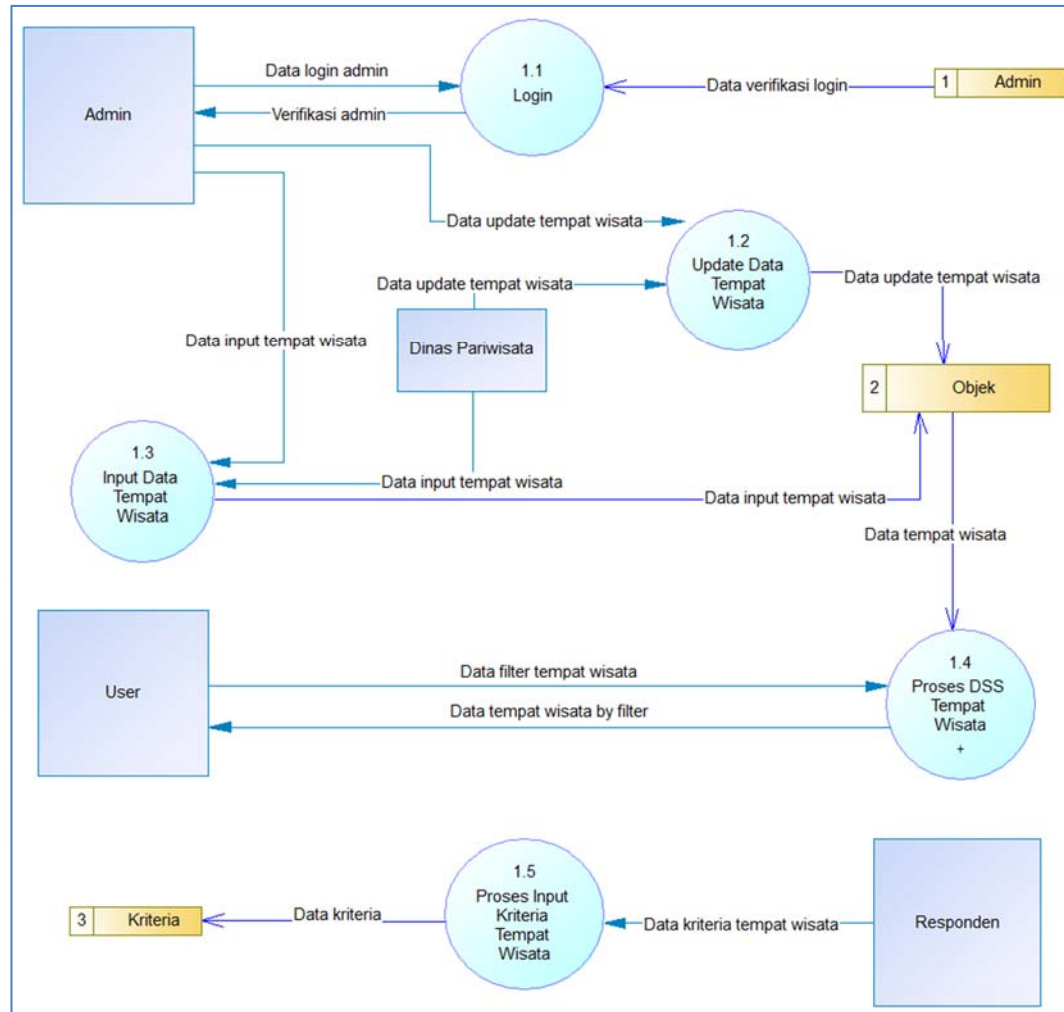
### 3.2.1 Data Flow Diagram

Sistem yang dibangun memiliki *context* diagram, dimana terdapat satu proses utama dan empat entitas yang terlibat dalam sistem yaitu *user*, *admin*, dinas pariwisata, dan responden. *User* hanya terbatas dalam melakukan *filter* untuk menentukan objek wisata, *admin* memiliki akses ke bagian *back-end* sistem untuk melakukan *management* data, dinas pariwisata dan responden sebagai sumber data.



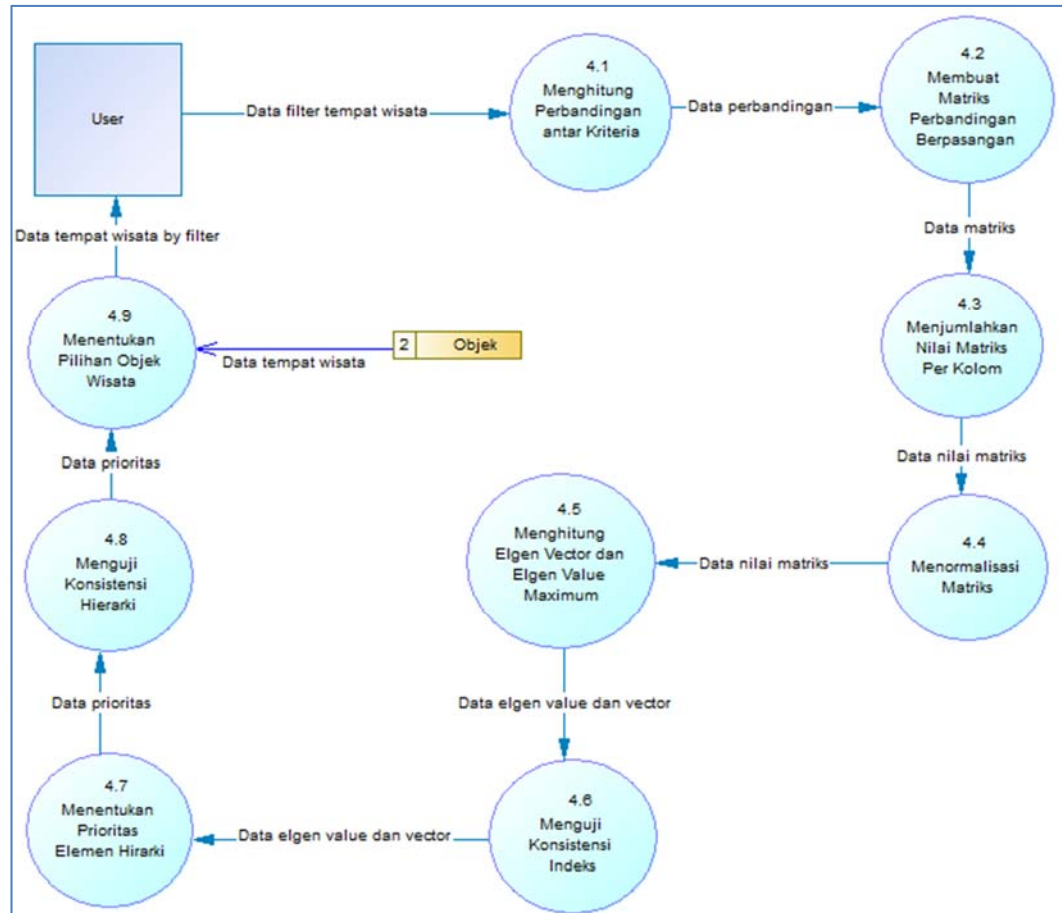
Gambar 3.8 *Context* Diagram

Pada Diagram *level 1*, proses utama yang terdapat pada *context* diagram dipecah menjadi lima proses, yaitu *login*, *update* data tempat wisata, *input* data tempat wisata, proses DSS tempat wisata, dan proses *input* kriteria tempat wisata. Pada *level* ini juga terdapat tiga data *store* yang terdiri dari *admin*, objek, dan kriteria.



Gambar 3.9 DFD Level 1 *Decision Support System* Pemilihan Objek Wisata

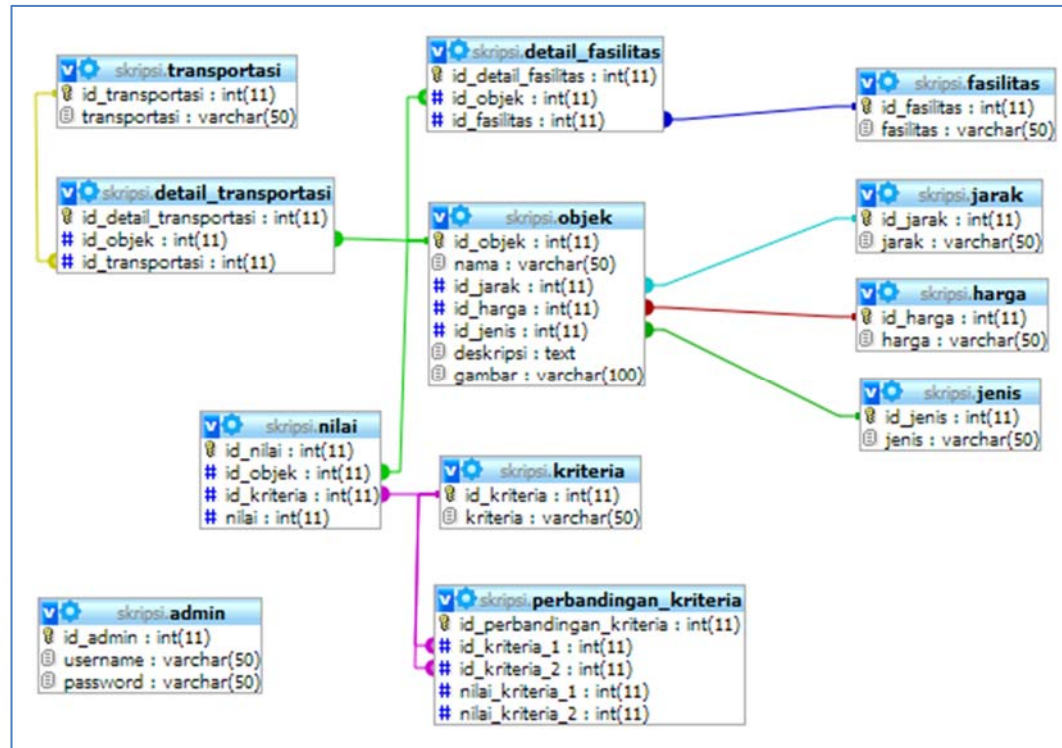
Pada diagram *level 2*, proses AHP dipecah menjadi beberapa proses yaitu, menghitung perbandingan antar kriteria, membuat matriks perbandingan berpasangan, menjumlahkan nilai matriks per kolom, menormalisasi matriks, menghitung *eigen vector* dan *eigen value* maksimum, menguji konsistensi indeks, menentukan prioritas elemen hirarki, menguji konsistensi hirarki, dan menentukan pilihan objek wisata. Di sini hanya terdapat satu data *store* yaitu objek.



Gambar 3.10 DFD Level 2 Proses DSS Tempat Wisata

### 3.2.2 Entity Relationship Diagram

Pada sistem ini, terdapat 12 tabel data, dimana 11 dari tabel tersebut saling berhubungan. Tabel-tabel tersebut terdiri dari tabel OBJEK, tabel nilai, tabel KRITERIA, tabel PERBANDINGAN\_KRITERIA, tabel JARAK, tabel HARGA, tabel JENIS, tabel FASILITAS, tabel DETAIL\_FASILITAS, tabel TRANSPORTASI, tabel DETAIL\_TRANSPORTASI, serta tabel ADMIN yang tidak memiliki hubungan dengan tabel lainnya. Tabel ADMIN hanya digunakan untuk menyimpan data *username* dan *password admin* untuk masuk ke *back-end* sistem.



Gambar 3.11 ERD *Decision Support System* Pemilihan Objek Wisata

Hubungan antara tabel OBJEK dengan tabel JARAK, tabel OBJEK dengan tabel HARGA, tabel OBJEK dengan tabel JENIS adalah *one-to-one* relationship, dimana satu OBJEK hanya memiliki satu JARAK, HARGA, dan JENIS. Hubungan antara tabel KRITERIA dengan tabel NILAI adalah *one-to-many relationship*, dimana satu KRITERIA dapat memiliki satu atau lebih NILAI. Hubungan antara tabel KRITERIA dengan tabel PERBANDINGAN\_KRITERIA adalah *one-to-many relationship*, dimana satu KRITERIA dapat memiliki satu atau lebih PERBANDINGAN\_KRITERIA. Hubungan antara tabel OBJEK dengan tabel NILAI adalah *one-to-many relationship*, dimana satu OBJEK dapat memiliki satu atau lebih NILAI. Hubungan antara tabel OBJEK dengan tabel DETAIL\_TRANSPORTASI adalah *one-to-many relationship*, dimana satu OBJEK dapat memiliki satu atau lebih DETAIL\_TRANSPORTASI. Hubungan antara tabel OBJEK dengan tabel DETAIL\_FASILITAS adalah *one-to-many*

*relationship*, dimana satu OBJEK dapat memiliki satu atau lebih DETAIL\_FASILITAS. Hubungan antara tabel FASILITAS dengan tabel DETAIL\_FASILITAS adalah *one-to-many relationship*, dimana satu DETAIL\_FASILITAS dapat memiliki satu atau lebih FASILITAS. Hubungan antara tabel TRANSPORTASI dengan tabel DETAIL\_TRANSPORTASI adalah *one-to-many relationship*, dimana satu DETAIL\_TRANSPORTASI dapat memiliki satu atau lebih TRANSPORTASI.

### 3.2.3 Struktur Tabel

Adapun keterangan serta detail dari tabel-tabel yang dipergunakan adalah sebagai berikut.

Nama tabel : ADMIN

Fungsi : menyimpan informasi *admin* yang dapat *login* ke *back-end*

Tabel 3.1 Struktur Tabel ADMIN

| Nama Field | Tipe    | Panjang | Keterangan         |
|------------|---------|---------|--------------------|
| id_admin   | int     |         | <i>Primary Key</i> |
| username   | varchar | 50      |                    |
| password   | varchar | 50      |                    |

Nama tabel : OBJEK

Fungsi : menyimpan informasi objek wisata

Tabel 3.2 Struktur Tabel OBJEK

| Nama Field | Tipe    | Panjang | Keterangan         |
|------------|---------|---------|--------------------|
| id_objek   | int     |         | <i>Primary Key</i> |
| nama       | varchar | 50      |                    |
| id_jarak   | int     |         | <i>Foreign Key</i> |
| id_harga   | int     |         | <i>Foreign Key</i> |
| id_jenis   | int     |         | <i>Foreign Key</i> |
| deskripsi  | text    |         |                    |
| gambar     | varchar | 100     |                    |



Nama tabel : KRITERIA

Fungsi : menyimpan informasi kriteria objek wisata

Tabel 3.3 Struktur Tabel KRITERIA

| Nama Field  | Tipe    | Panjang | Keterangan  |
|-------------|---------|---------|-------------|
| id_kriteria | int     |         | Primary key |
| kriteria    | varchar | 50      |             |

Nama tabel : JENIS

Fungsi : menyimpan informasi jenis objek wisata

Tabel 3.4 Struktur Tabel JENIS

| Nama Field | Tipe    | Panjang | Keterangan  |
|------------|---------|---------|-------------|
| id_jenis   | int     |         | Primary Key |
| jenis      | varchar | 50      |             |

Nama tabel : JARAK

Fungsi : menyimpan informasi jarak objek wisata

Tabel 3.5 Struktur Tabel JARAK

| Nama Field | Tipe    | Panjang | Keterangan  |
|------------|---------|---------|-------------|
| id_jarak   | int     |         | Primary key |
| jarak      | varchar | 50      |             |

Nama tabel : HARGA

Fungsi : menyimpan informasi harga objek wisata

Tabel 3.6 Struktur Tabel HARGA

| Nama Field | Tipe    | Panjang | Keterangan  |
|------------|---------|---------|-------------|
| id_harga   | int     |         | Primary Key |
| harga      | varchar | 50      |             |

Nama tabel : FASILITAS

Fungsi : menyimpan informasi fasilitas objek wisata

Tabel 3.7 Struktur Tabel FASILITAS

| Nama Field   | Tipe    | Panjang | Keterangan  |
|--------------|---------|---------|-------------|
| id_fasilitas | int     |         | Primary Key |
| fasilitas    | varchar | 50      |             |

Nama tabel : TRANSPORTASI

Fungsi : menyimpan informasi transportasi objek wisata

Tabel 3.8 Struktur Tabel TRANSPORTASI

| Nama Field      | Tipe    | Panjang | Keterangan  |
|-----------------|---------|---------|-------------|
| id transportasi | int     |         | Primary Key |
| transportasi    | varchar | 50      |             |

Nama tabel : DETAIL\_FASILITAS

Fungsi : menyimpan informasi detail fasilitas objek wisata

Tabel 3.9 Struktur Tabel DETAIL\_FASILITAS

| Nama Field          | Tipe | Panjang | Keterangan  |
|---------------------|------|---------|-------------|
| id_detail_fasilitas | int  |         | Primary Key |
| id_objek            | int  |         | Foreign Key |
| id_fasilitas        | int  |         | Foreign Key |

Nama tabel : DETAIL\_TRANSPORTASI

Fungsi : menyimpan informasi detail transportasi objek wisata

Tabel 3.10 Struktur Tabel DETAIL\_TRANSPORTASI

| Nama Field             | Tipe | Panjang | Keterangan  |
|------------------------|------|---------|-------------|
| id_detail_transportasi | int  |         | Primary Key |
| id_objek               | int  |         | Foreign Key |
| id_transportasi        | int  |         | Foreign Key |

Nama tabel : nilai

Fungsi : menyimpan informasi nilai dari kriteria objek wisata

Tabel 3.11 Struktur Tabel Nilai

| Nama Field  | Tipe | Panjang | Keterangan  |
|-------------|------|---------|-------------|
| id_nilai    | int  |         | Primary Key |
| id_objek    | int  |         | Foreign Key |
| id_kriteria | int  |         | Foreign Key |
| nilai       | int  |         |             |

Nama tabel : PERBANDINGAN\_KRITERIA

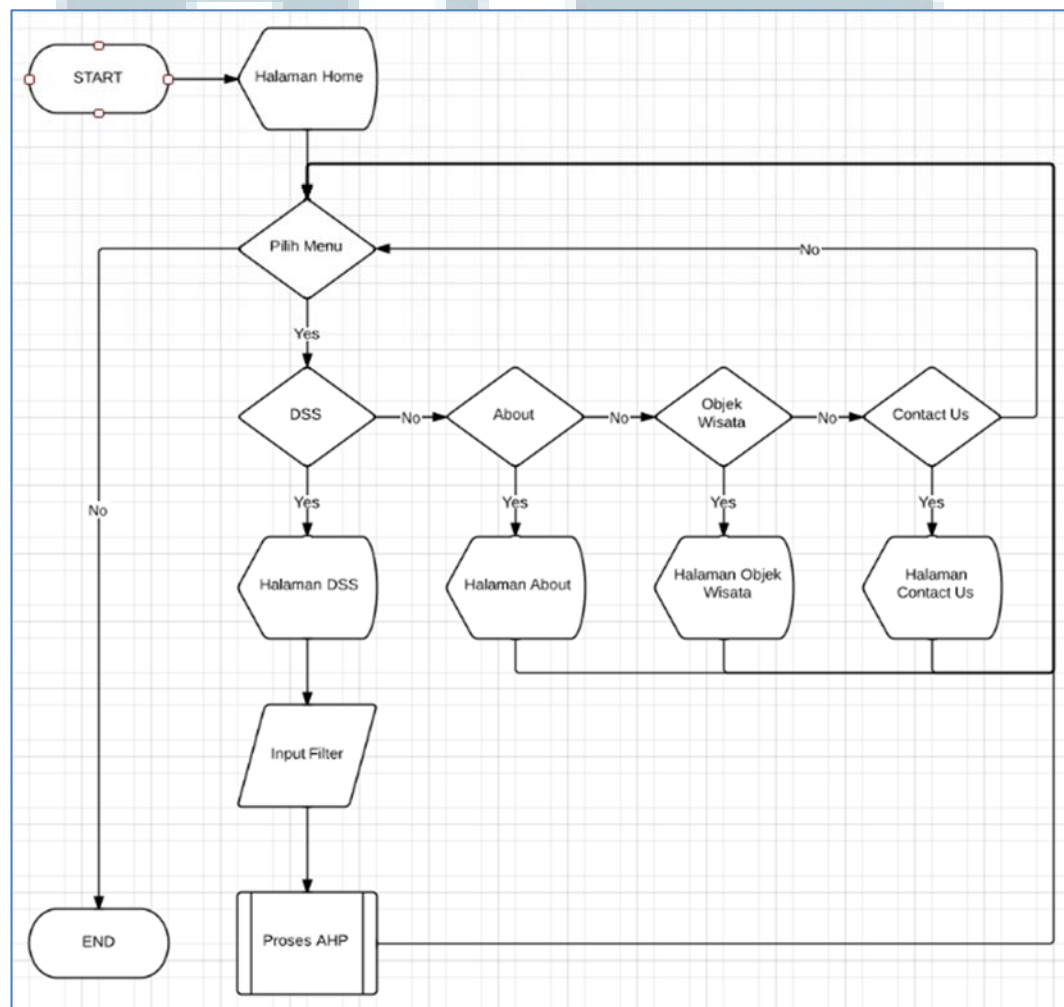
Fungsi : menyimpan informasi nilai perbandingan kriteria objek wisata

Tabel 3.12 Struktur Tabel PERBANDINGAN KRITERIA

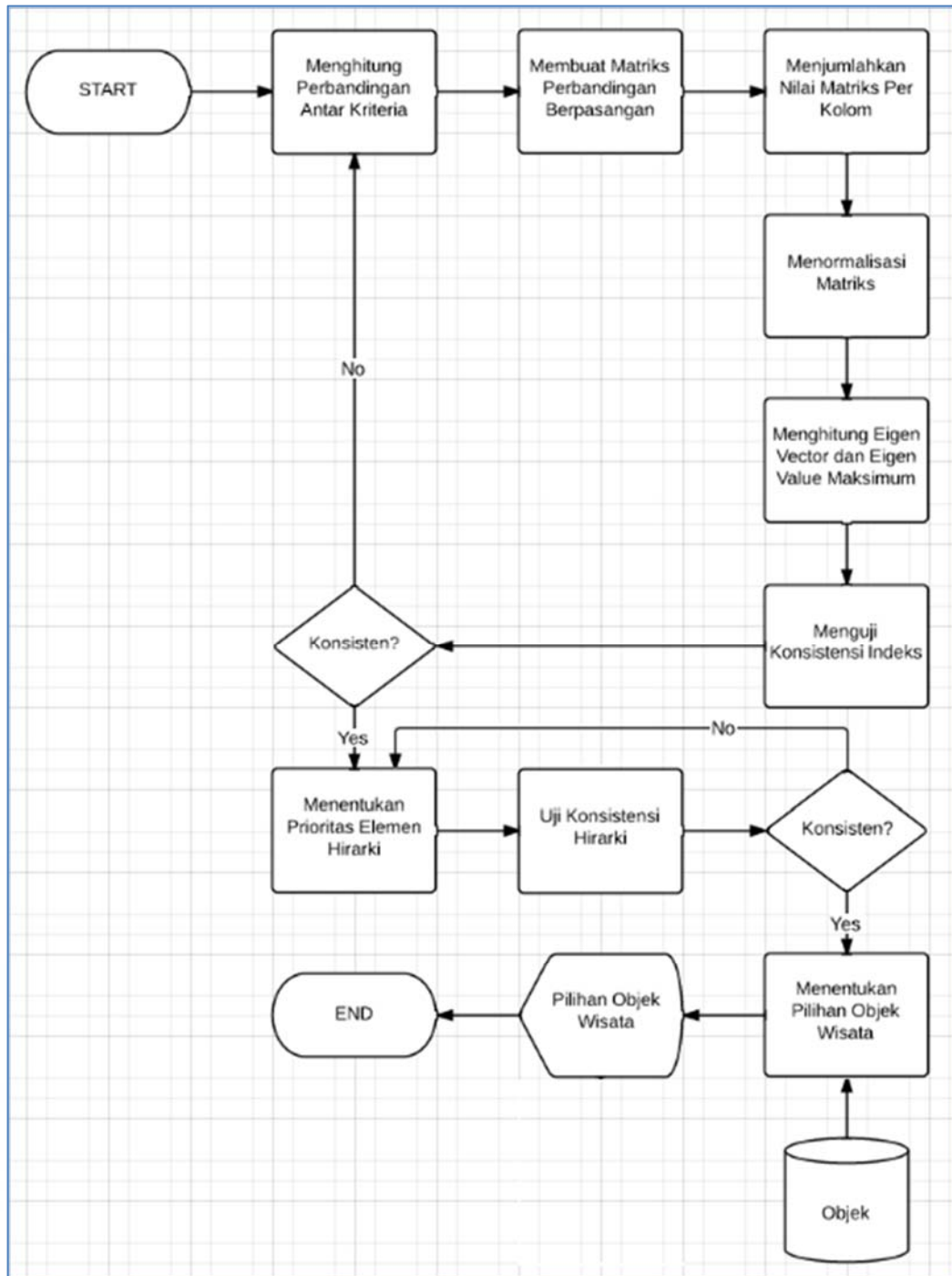
| Nama Field               | Tipe | Panjang | Keterangan  |
|--------------------------|------|---------|-------------|
| id_perbandingan_kriteria | int  |         | Primary key |
| id_kriteria_1            | int  |         | Foreign Key |
| id_kriteria_2            | int  |         | Foreign Key |
| nilai_kriteria_1         | int  |         |             |
| nilai_kriteria_2         | int  |         |             |

### 3.2.4 Flowchart

Sistem yang dibangun terdiri dari bagian *front-end* dan bagian *back-end*. Bagian *front-end* merupakan bagian yang digunakan oleh *user*, sedangkan bagian *back-end* merupakan bagian yang digunakan oleh *admin*.

Gambar 3.1 *Flowchart Front-End*

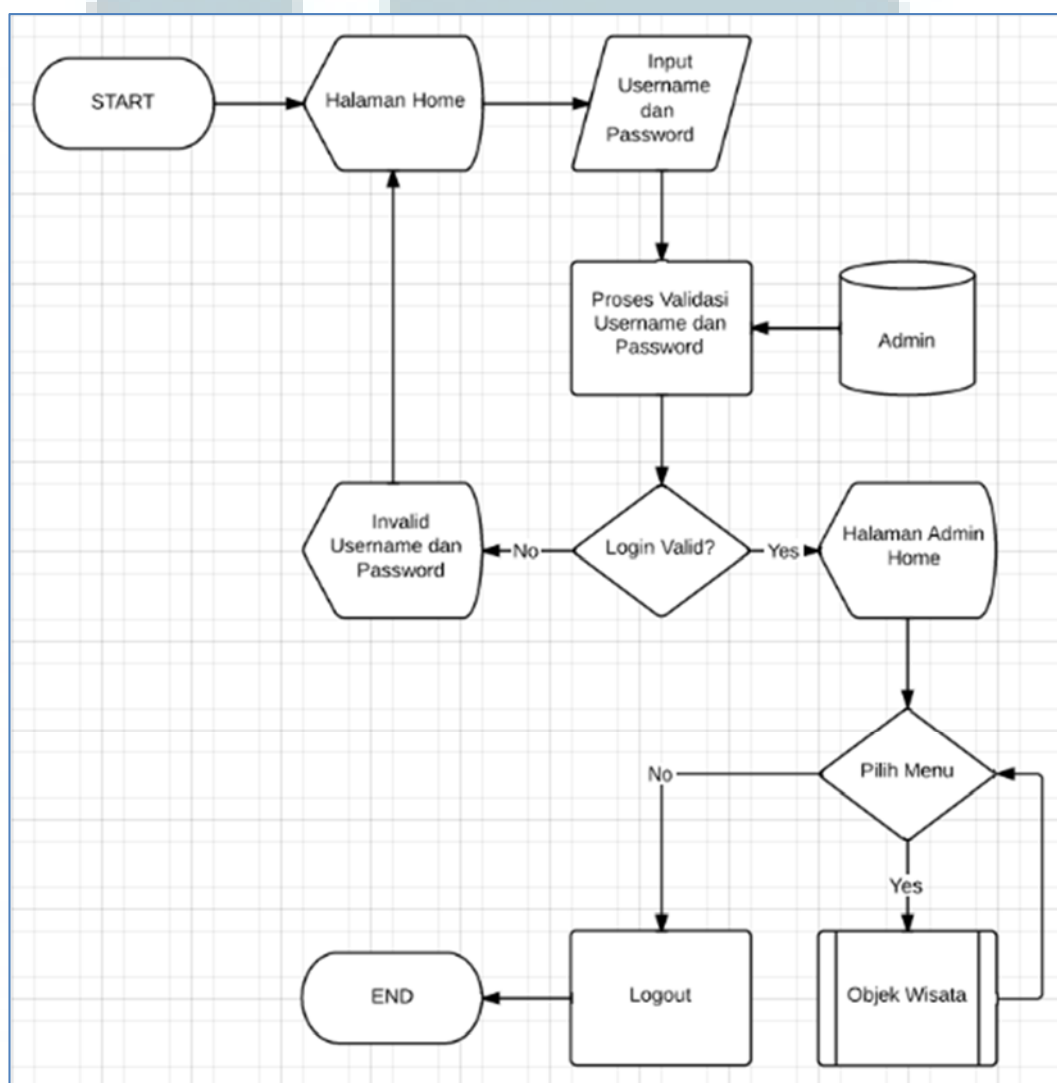
Pertama-tama *user* akan masuk ke halaman *home*, dimana terdapat beberapa *menu* seperti *DSS*, *About*, *Objek Wisata*, dan *Contact Us*. Untuk *DSS* sendiri, *user* diharuskan untuk menentukan *filter* yang akan digunakan pada proses *AHP*.



Gambar 3.2 *Flowchart Analytical Hierarchy Process*

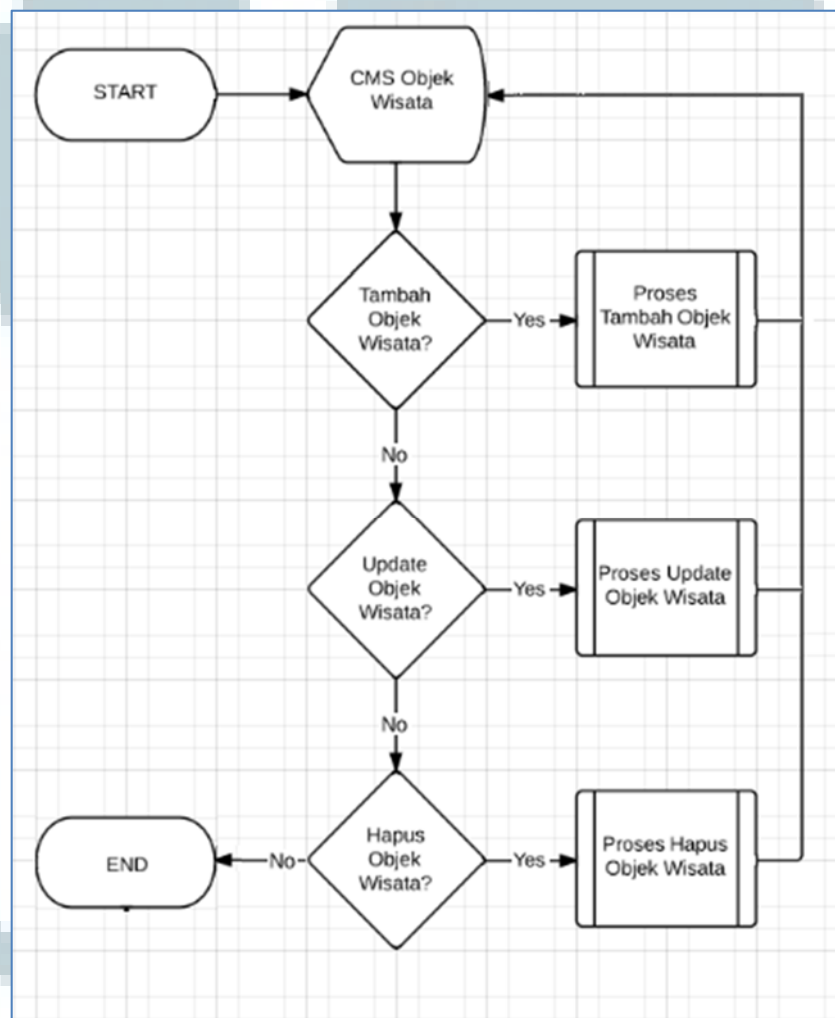
Gambar 3.2 di atas menunjukkan tahap-tahap dari algoritma *Analytical Hierarchy Process* yang dimulai dari perhitungan perbandingan antar kriteria sampai akhirnya dapat menentukan pilihan dari objek wisata yang sesuai dengan *filter* dari *user*.

Pada aplikasi ini juga terdapat *back-end* aplikasi, dimana *admin* dapat melakukan *management content* dari aplikasi. Gambar 3.3 menunjukkan *flowchart* dari *back-end* aplikasi.



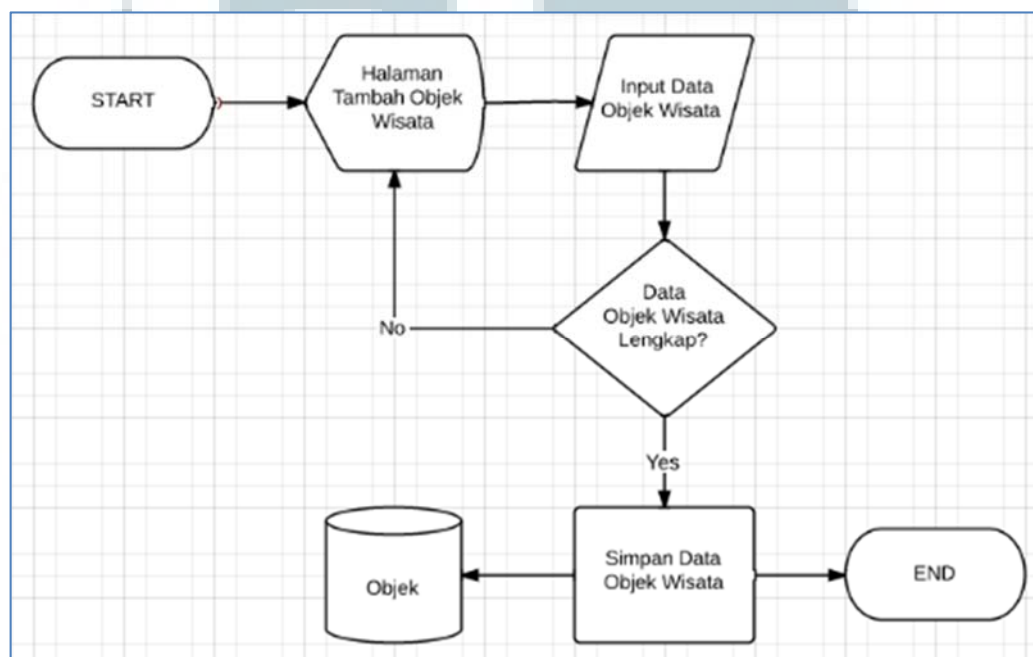
Gambar 3.3 *Flowchart Back-End*

Pertama-tama *admin* harus melakukan *login* untuk dapat masuk ke *back-end* yang akan mengvalidasi *username* dan *password* yang dimasukkan. Apabila tidak valid, akan diminta *login* kembali. Jika valid, maka *admin* akan masuk ke *home admin*, dimana terdapat menu CMS objek wisata. *Admin* dapat mengolah data objek wisata seperti menambahkan, mengubah, ataupun menghapus objek wisata. *Admin* juga dapat *logout* jika telah selesai melakukan pengolahan data objek wisata yang dilakukan. Untuk *flowchart* CMS objek wisata dapat dilihat pada gambar 3.4 di bawah ini.



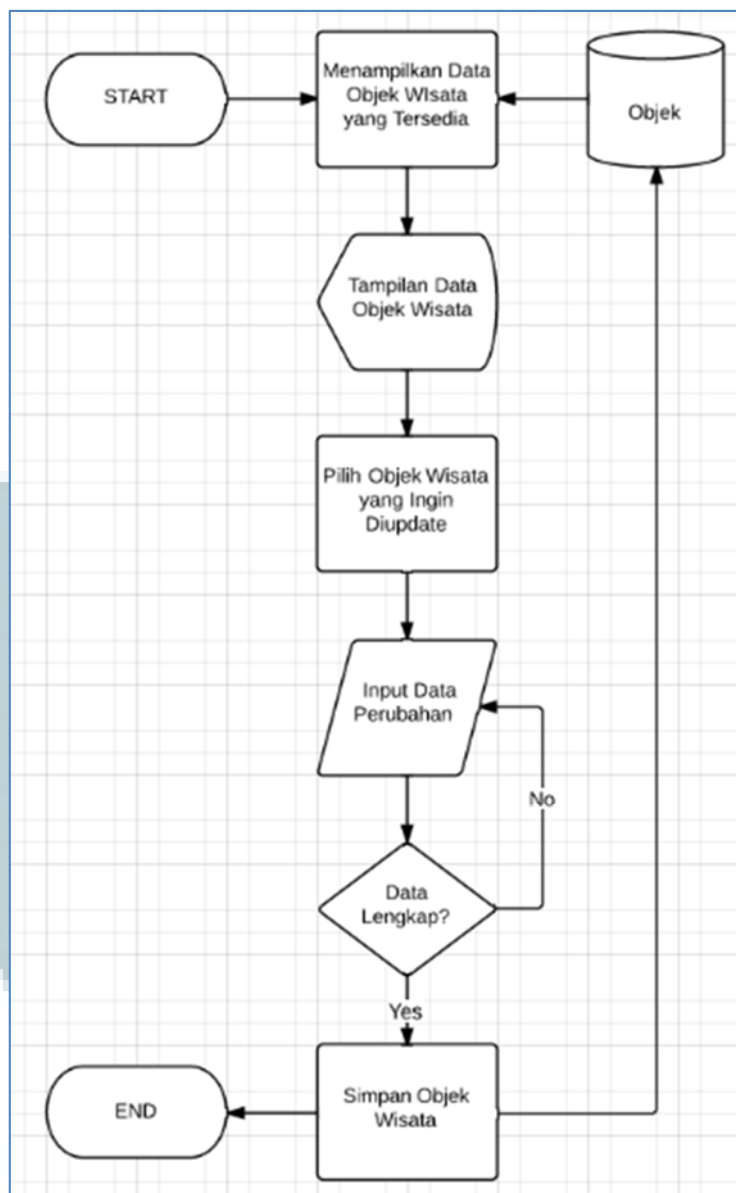
Gambar 3.4 *Flowchart* CMS Objek Wisata

Untuk melakukan penambahan objek wisata baru, *admin* harus melengkapi semua data dari objek wisata yang diperlukan dimulai dari jenis objek wisata, nama, jarak, harga, fasilitas, transportasi, deskripsi, gambar, dan penilaian dari tiap kriteria yang tersedia untuk objek wisata tersebut. Jika data yang diisi belum lengkap, maka akan diminta untuk melakukan pengisian data kembali hingga semua data lengkap dan apabila sudah lengkap maka akan dilakukan penyimpanan ke dalam *database*.



Gambar 3.5 *Flowchart* Tambah Objek Wisata

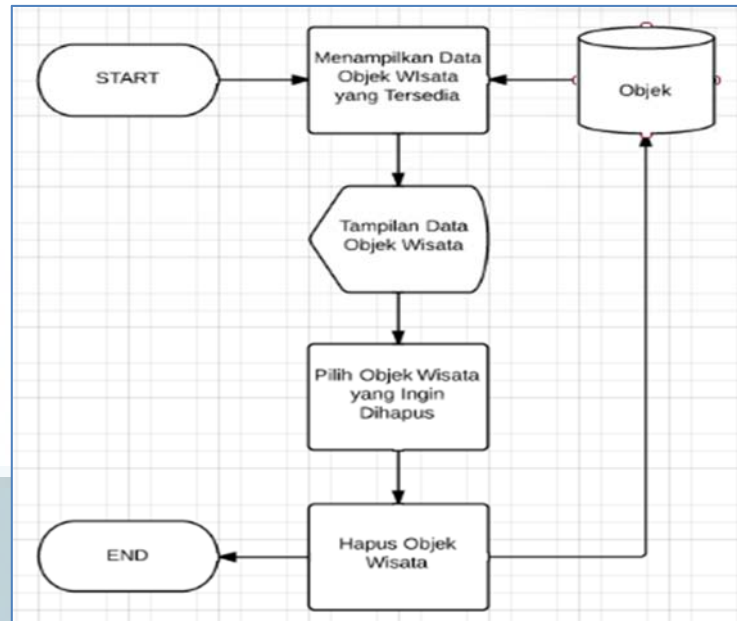
Untuk melakukan *update*, pertama-tama akan ditampilkan seluruh data objek wisata yang ada pada *database*. *Admin* diharuskan untuk memilih terlebih dahulu objek wisata yang ingin di-*update*. Kemudian *admin* dapat melakukan *update* sesuai yang diinginkan. Jika data yang diisi belum lengkap, maka akan diminta untuk melakukan pengisian data ulang dan apabila data yang dimasukkan sudah lengkap, maka akan dilakukan penyimpanan ke dalam *database* untuk dilakukan *update*.



Gambar 3.6 *Flowchart Update Objek Wisata*

Untuk melakukan penghapusan data, pertama-tama akan ditampilkan seluruh data objek wisata yang ada pada *database* dan *admin* dapat langsung memilih objek wisata yang ingin dihapus yang kemudian akan dilanjutkan dengan peng-update-an *database*.

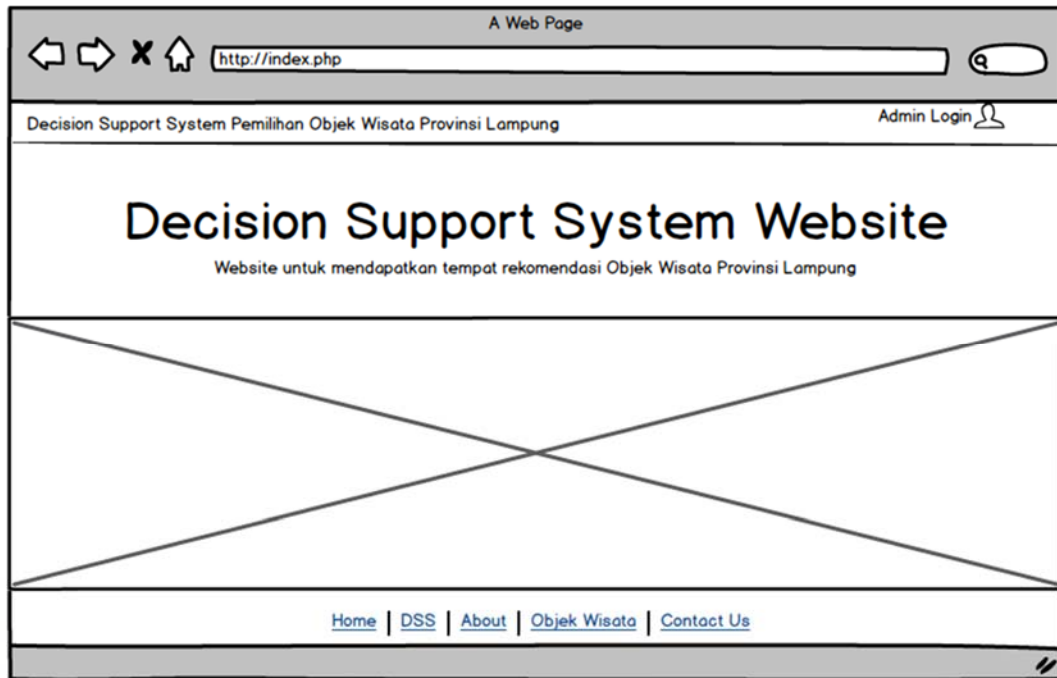




Gambar 3.7 Flowchart Hapus Objek Wisata

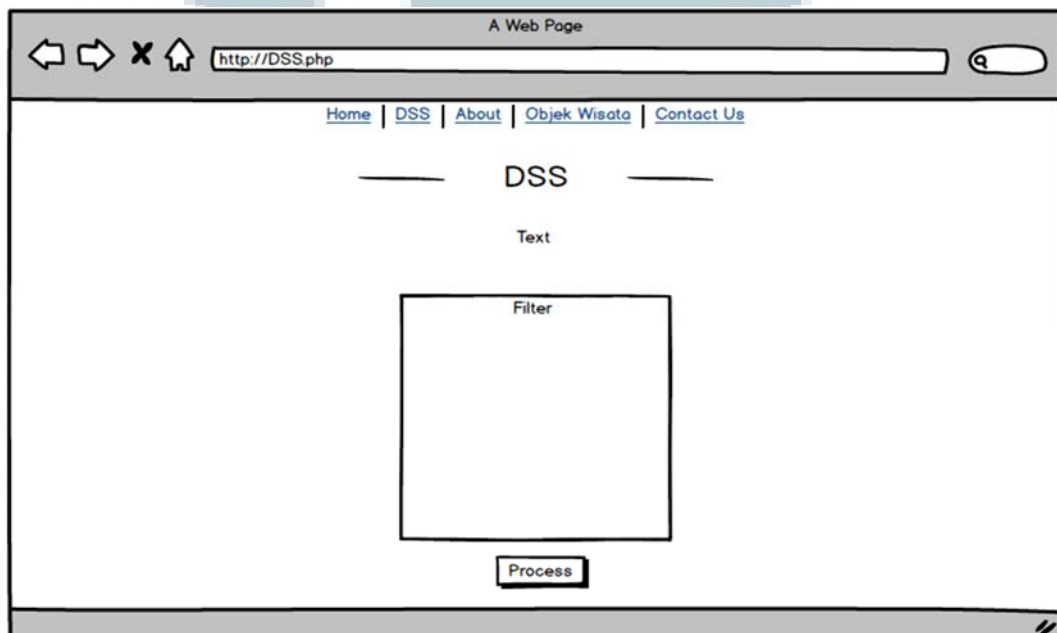
### 3.2.5 Rancangan Antarmuka

Sistem ini memiliki beberapa rancangan antarmuka yang terdiri dari rancangan antarmuka bagian *front-end* dan rancangan antarmuka bagian *back-end*. Bagian *front-end* merupakan bagian yang digunakan oleh *user*, sedangkan bagian *back-end* merupakan bagian yang digunakan oleh *admin*. Gambar 3.12 menunjukkan halaman awal yang ditampilkan untuk *user*. Terdapat *slide show* yang menampilkan tiga objek wisata yang berbeda dari masing-masing kategori objek wisata. Pada halaman ini juga terdapat *link* untuk *login admin* serta menu-menu halaman *front-end* lain yang dapat dipilih oleh *user*.



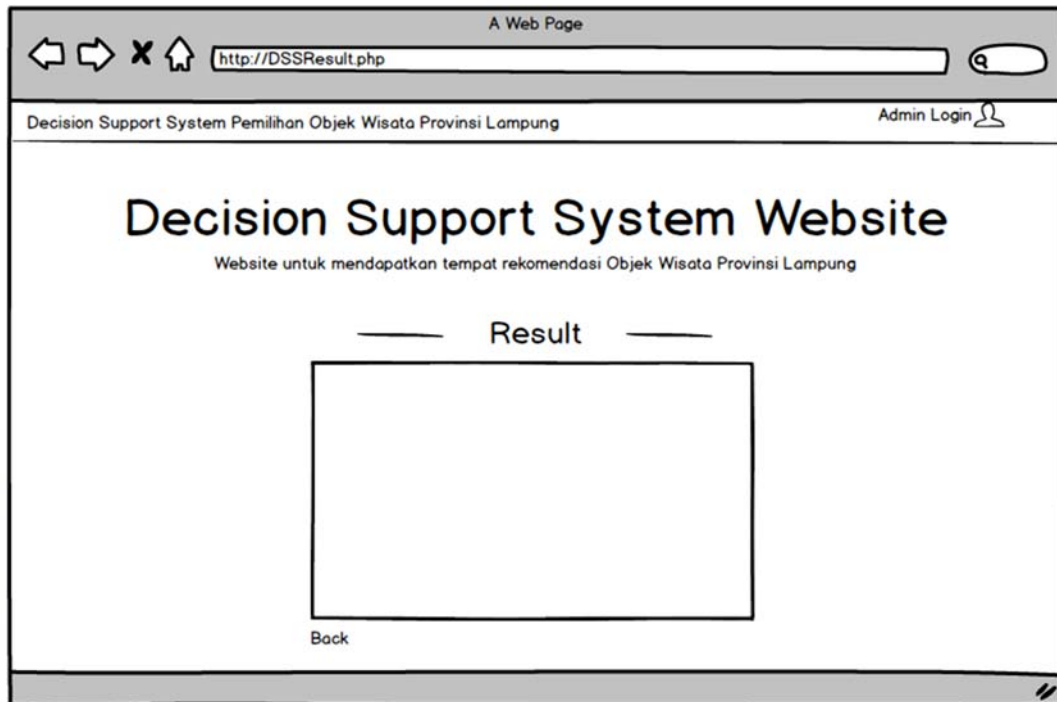
Gambar 3.12 Rancangan Antarmuka halaman *Home*

Pada Halaman DSS, sistem pendukung keputusan akan melakukan pemrosesan setelah *user* melakukan *filter* dan menekan tombol proses yang kemudian menampilkan pilihan di bagian *result*.



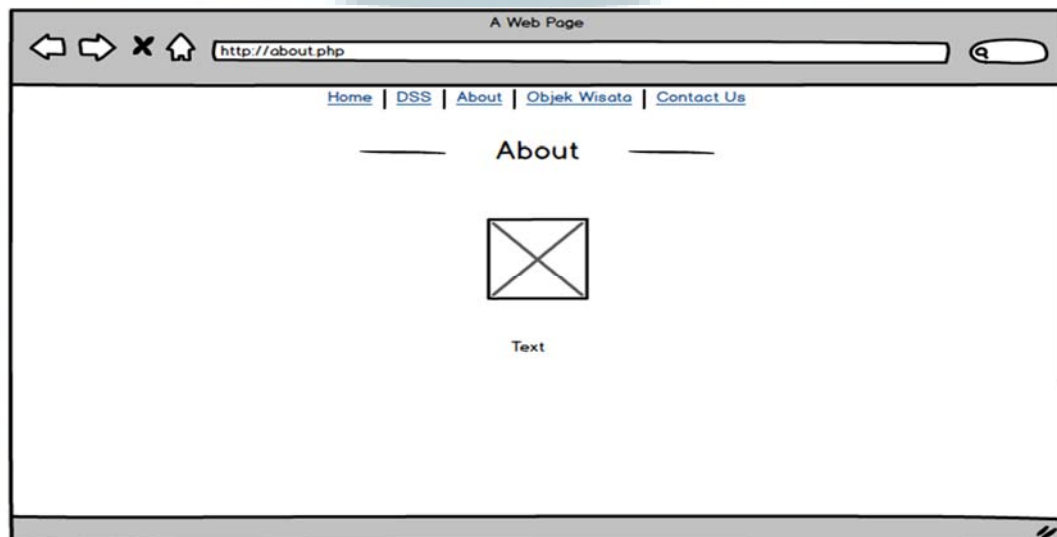
Gambar 3.13 Rancangan Antarmuka halaman DSS

Pada halaman DSSResult ini, ditampilkan hasil rekomendasi dari *filter* yang telah dilakukan oleh *user*.



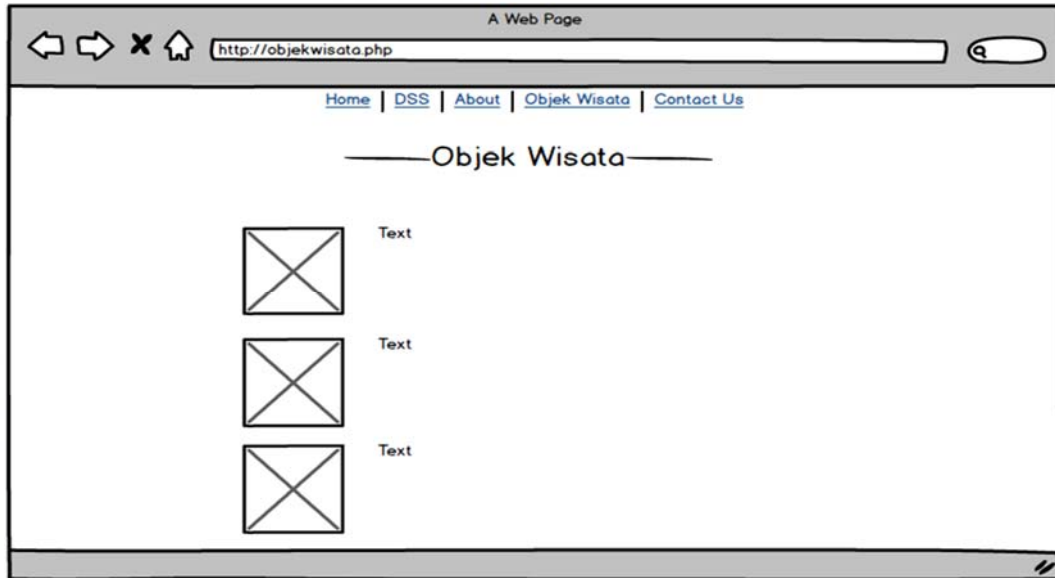
Gambar 3.14 Rancangan Antarmuka Halaman DSSResult

Halaman *about* ini berisikan sebuah logo dari algoritma yang dipakai yaitu *Analytical Hierarchy Process* dan sedikit tentang pembuatan website ini.



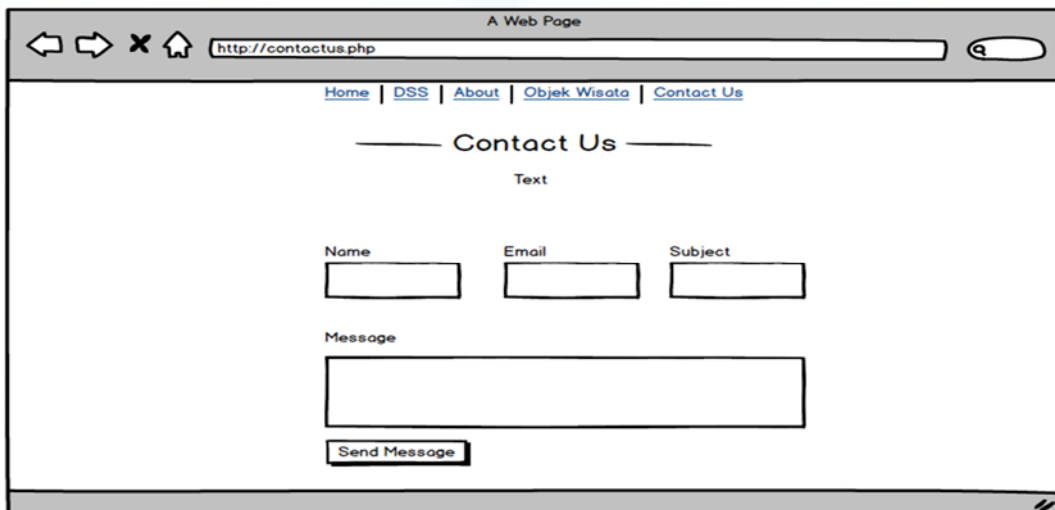
Gambar 3.15 Rancangan Antarmuka Halaman *About*

Pada bagian halaman objek wisata, terdapat 3 buah gambar yang mewakili masing-masing dari jenis objek wisata dan disertakan juga nama-nama dari objek wisata dari masing-masing jenis tersebut.



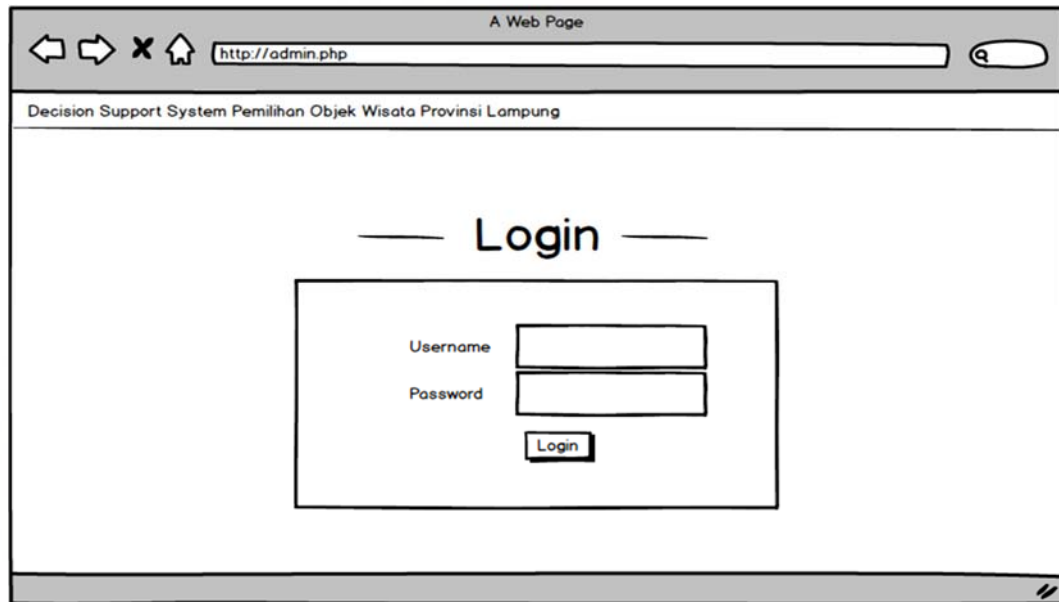
Gambar 3.16 Rancangan Antarmuka Halaman Objek Wisata

Halaman *contact us* merupakan halaman *front-end* yang terakhir untuk *user*. User dapat mengirimkan pesan melalui *email* dengan mengisi data-data seperti yang ditunjukkan pada gambar 3.16.



Gambar 3.17 Rancangan Antarmuka Halaman *Contact Us*

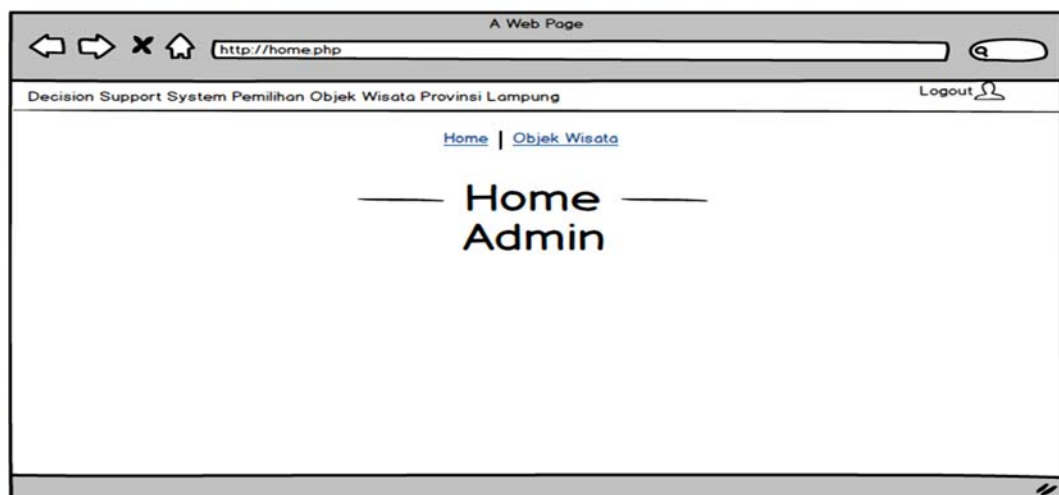
Halaman berikut merupakan halaman *login* yang dapat digunakan oleh *admin* untuk melakukan pengolahan data yang ada pada sistem.



The screenshot shows a web browser window with the title "A Web Page" and the address bar containing "http://admin.php". The page content is titled "Decision Support System Pemilihan Objek Wisata Provinsi Lampung". In the center, there is a "Login" form with two input fields labeled "Username" and "Password", and a "Login" button below them.

Gambar 3.18 Rancangan Antarmuka Halaman *Admin Login*

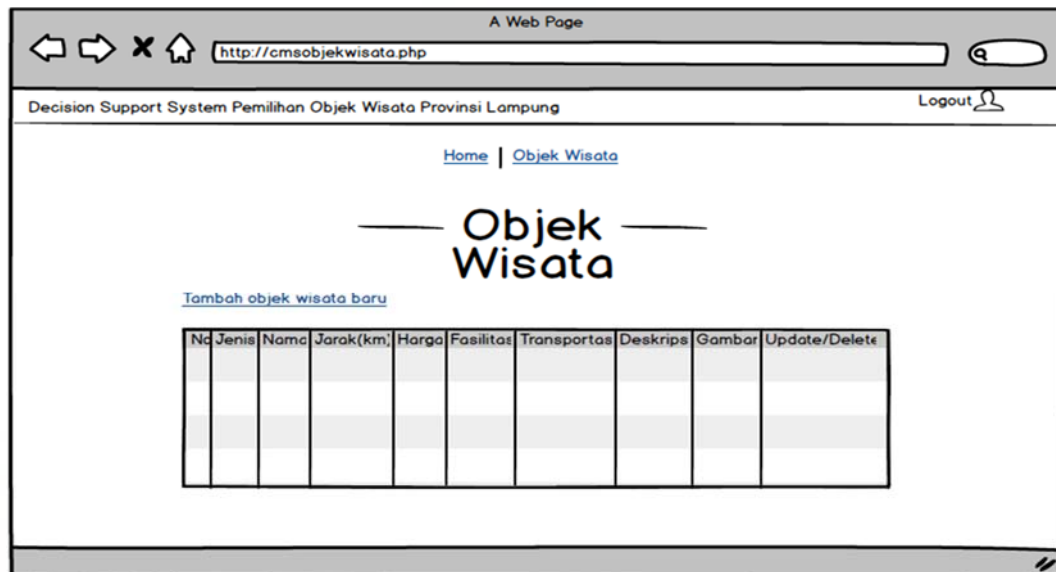
Setelah *admin login* dengan memasukkan *username* dan *password* dan divalidasi, *admin* akan dihadapkan pada halaman *home* dari *back-end* sistem. Pada bagian *back-end* ini terdapat menu untuk melakukan pengolahan data objek wisata.



The screenshot shows a web browser window with the title "A Web Page" and the address bar containing "http://home.php". The page content is titled "Decision Support System Pemilihan Objek Wisata Provinsi Lampung". In the top right corner, there is a "Logout" link with a user icon. In the center, there is a "Home Admin" form with a "Home" link and an "Objek Wisata" link above it.

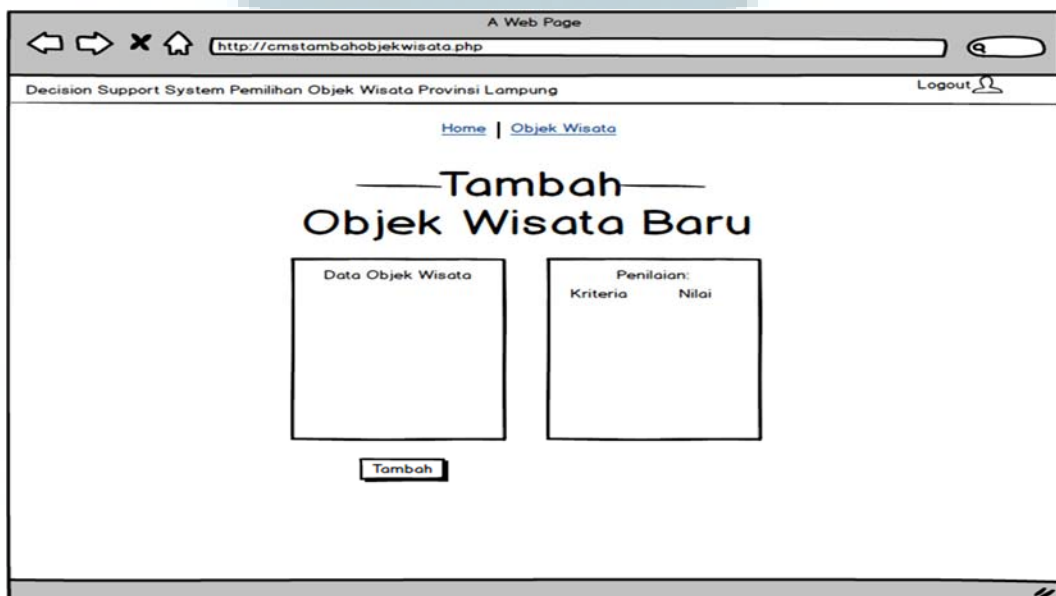
Gambar 3.19 Rancangan Antarmuka Halaman *Home Admin*

Pada halaman pengolahan objek wisata tersebut akan ditampilkan keseluruhan dari objek wisata yang telah tersimpan di *database*.



Gambar 3.20 Rancangan Antarmuka Halaman CMS Objek Wisata

Halaman berikut merupakan halaman yang muncul jika *admin* mau melakukan penambahan dari objek wisata. *Admin* diharuskan memasukkan seluruh data yang diperlukan secara lengkap.



Gambar 3.21 Rancangan Antarmuka Halaman Tambah Objek Wisata

Halaman berikut merupakan halaman untuk melakukan *update* objek wisata. *Admin* juga diharuskan mengisi lengkap seluruh data yang diperlukan sebelum melakukan *update* data.

A Web Page

http://cmsupdateobjekwisata.php

Decision Support System Pemilihan Objek Wisata Provinsi Lampung Logout

Home | Objek Wisata

## Update Objek Wisata

|                   |                                 |
|-------------------|---------------------------------|
| Data Objek Wisata | Penilaian:<br>Kriteria    Nilai |
|-------------------|---------------------------------|

Update

Gambar 3.22 Rancangan Antarmuka *Update* Objek Wisata

UMMN