

## **BAB III**

### **PERANCANGAN PENELITIAN**

#### **3.1 Peralatan Pendukung**

Peralatan pendukung dalam pembuatan aplikasi berbasis *website* terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk mendukung berjalannya perancangan dan pembuatan.

##### **3.1.1 Perangkat Keras**

Adapun perangkat keras yang dibutuhkan dalam merancang dan membuat *bussines logic* dari *website* perencanaan program penjadwalan acara program studi Teknik Informatika :

1. *Processor* AMD A10-7400P.
2. *Memory* 4 GB.
3. *Hardisk* minimal 250 GB.
4. *Monitor, Keyboard* dan *Mouse*.

##### **3.1.2 Perangkat Lunak**

Adapun perangkat lunak yang dibutuhkan dalam merancang dan membuat dari *website* perencanaan program penjadwalan acara program studi Teknik Informatika :

1. Sistem operasi *windows* 10.
2. *Web Server* : *XAMPP*.

3. *Web Browser.*
4. Bahasa Pemrograman : PHP, *Javascript*, HTML, CSS.
5. *Database Server : MySQL.*

## **3.2 Bahan Penelitian**

### **3.2.1 Studi Literatur**

Proses studi literatur dilakukan dengan mempelajari bahan-bahan literatur tentang penjadwalan acara yang didapat melalui *internet*, jurnal, *blog*, dan buku-buku referensi. Sehingga bahan-bahan literatur yang telah di pelajari dapat di implementasikan untuk membuat sistem informasi penjadwalan acara yang baik. Selain itu penulis juga mempelajari berbagai teknologi yang sesuai dengan kebutuhan sistem dan kebutuhan pengguna yang menjadi acuan pembangunan sistem yang lebih baik.

### **3.2.2 Wawancara**

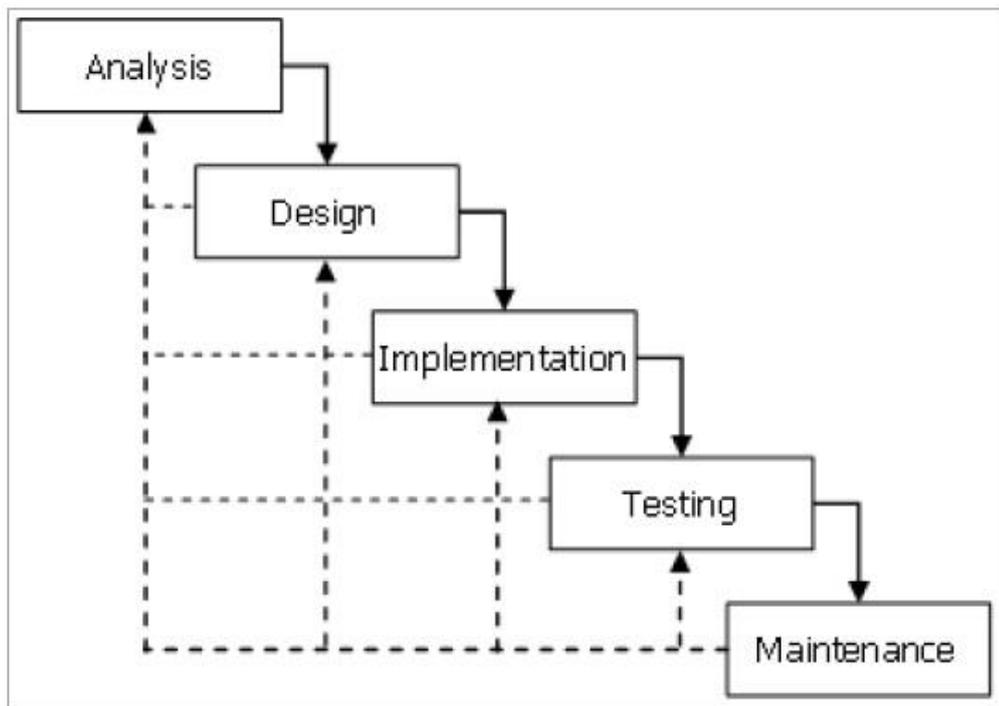
Wawancara dilakukan dengan cara mewawancarai langsung terhadap pihak terkait yang berguna untuk mendapatkan informasi maupun data-data mengenai penjadwalan acara yang dibutuhkan dan yang akan ditampilkan di *website* yang akan dibuat.

### **3.2.3 Observasi**

Metode pengumpulan data melalui pengamatan secara langsung atau peninjauan secara langsung di lokasi penelitian. Dari hasil penelitian secara langsung di UMY fakultas teknik program studi TI belum memiliki *website* untuk sarana informasi penjadwalan acara.

### 3.3 Langkah Penelitian

Dalam Metode pengembangan sistem penulis menggunakan metode SDLC (*Sistem Development Life Cycle*). Fase pengembangan sistem aplikasi ini disebut juga sebagai siklus hidup pengembangan sistem informasi, dengan tahapan-tahapan di antaranya :



**Gambar 3.1** Langkah Penelitian

a. Analisis

Fase analisis dimana akan dilakukan pertanyaan terhadap apa saja yang dibutuhkan oleh sistem seperti siapa pengguna sistem dan pengumpulan data apa saja yang dibutuhkan sistem.

b. Perancangan (*Design*)

Fase perancangan seperti *database* perancangan lain yang digunakan dalam menggambarkan *website* yang akan dibangun seperti antarmuka pengguna, *form*, *display*, laporan dan *file* yang ditetepkan.

c. Implementasi Sistem

Tahapan-tahapan yang dilakukan dalam implementasi sistem yaitu tahap pemrogramman (*coding*).

d. Pengujian (*Testing*)

Fase pengujian berhubungan dengan pemilihan perangkat keras, penyusunan perangkat lunak aplikasi (*coding*), serta pengujian apakah sistem yang dibuat sudah sesuai dengan kebutuhan pengguna atau belum.

e. Perawatan

Fase perawatan adalah tahap melakukan operasi dan pemeliharaan aplikasi yang telah dibuat dengan melakukan perbaikan-perbaikan terhadap masalah yang ada.

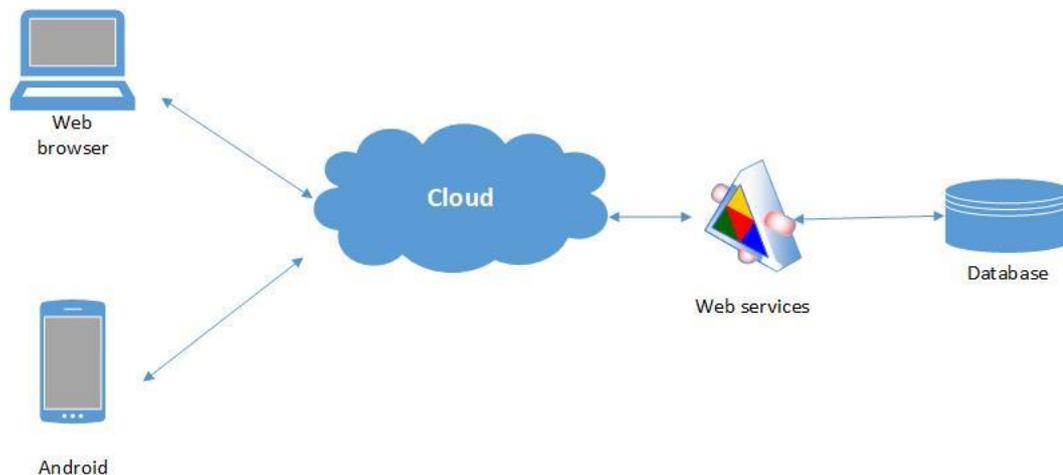
### 3.4 Sistem Yang Berjalan di TI UMY

Di TI UMY terdapat beberapa kendala dalam proses penyampaian informasi kepada mahasiswa dan pihak-pihak yang membutuhkan tentang informasi penjadwalan acara yang ada di TI UMY, bagi mahasiswa yang ingin mendapatkan informasi harus mengakses media sosial yang mempunyai kendala saat pencarian informasi yang diinginkan .

Dengan demikian maka akan menyulitkan mahasiswa dan tak terkecuali pihak TI UMY sendiri yang ingin memberikan informasi penjadwalan secara mudah. Maka dari itu perlu adanya sistem yang dapat menyampaikan informasi penjadwalan acara yang mudah untuk bisa didapatkan mahasiswa.

### 3.5 Arsitektur Web Service

*Web service* merupakan transportasi antara *database server* dan aplikasi. *Web service* juga sebagai penerjemah data sesuai dengan *database server* dan aplikasi. pada gambar dibawah ini merupakan contoh arsitektur *web service* dimana *web browser* yang menjalankan aplikasi akan melakukan request kepada web service lalu di berikan kembali oleh *database* kemudian *web service* melakukan penerjemahan data untuk pengguna android melalui *internet*



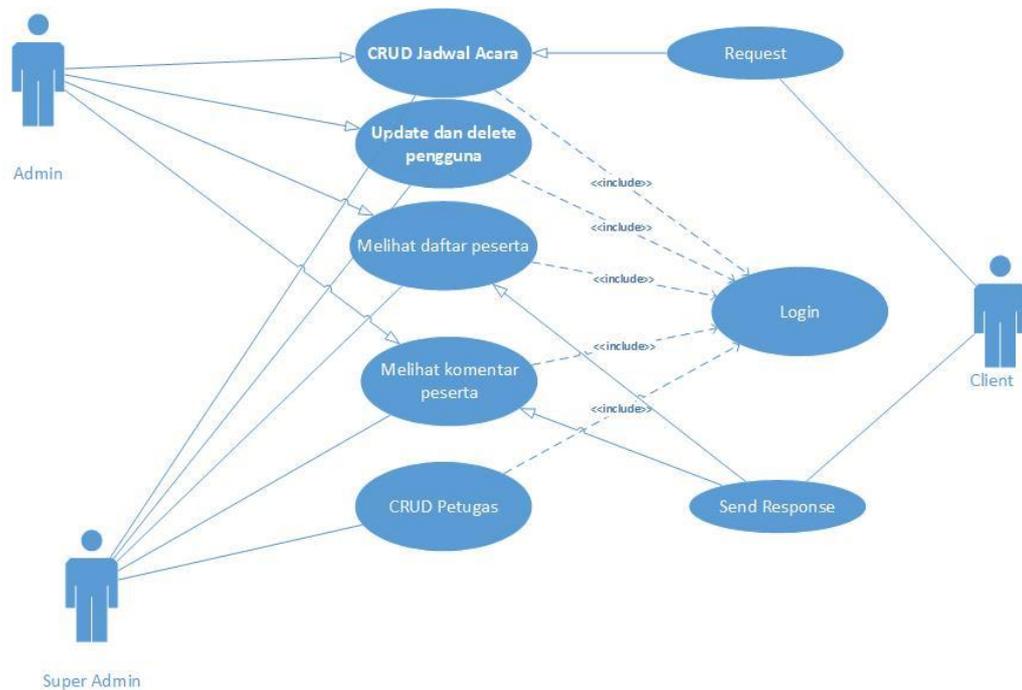
**Gambar 3.2** Arsitektur Sistem

### **3.6 Perancangan Sistem**

Perancangan sistem bertujuan untuk memberikan gambaran secara umum kepada pengguna dalam pembuatan rancangan sistem yang baru untuk mempermudah dalam pengolahan data, sehingga nantinya aplikasi yang dibuat lebih baik dari pengolahan yang masih manual.

#### **3.6.1 Use Case Diagram**

*Use case diagram* merupakan salah satu bagian dari UML, *use case diagram* adalah suatu diagram yang menggambarkan suatu sistem dan bagaimana sistem tersebut bekerja. Perancangan *use case* digunakan untuk memodelkan proses berdasarkan perspektif user. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Use case diagram* menggambarkan orang yang akan mengoperasikan atau berinteraksi dengan sistem. Berikut ini adalah *use case* sistem di *web* penjadwalan acara UMY.

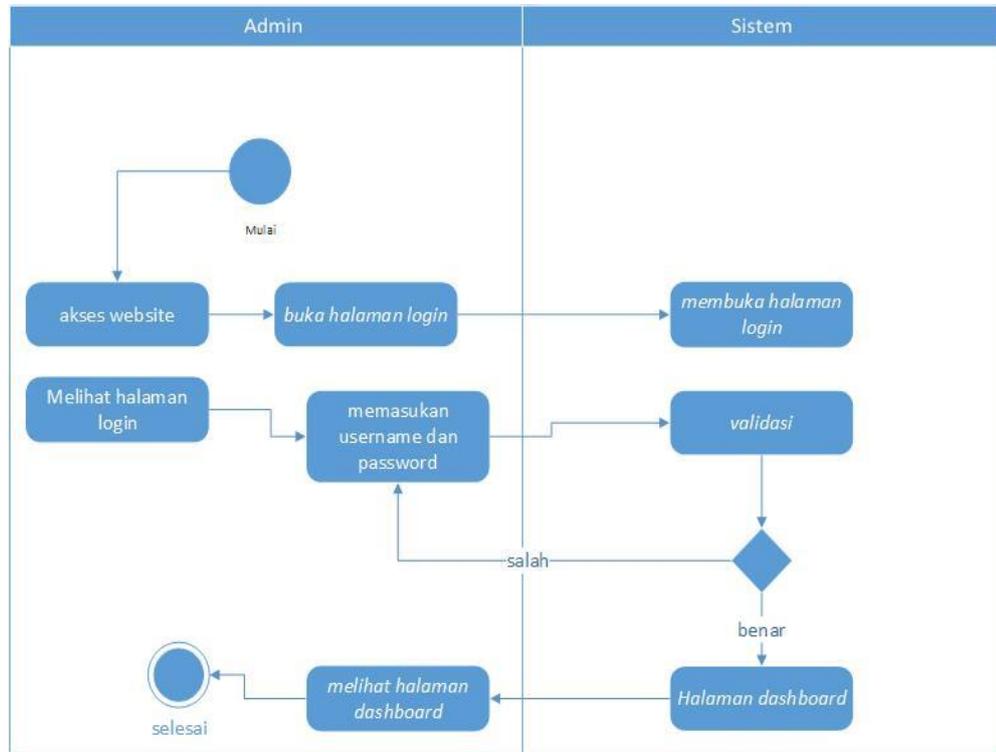


**Gambar 3.3** Use Case Diagram

### 3.6.2 Activity Diagram

*Activity diagram* digunakan untuk menganalisa proses. Sebuah *activity diagram* bukan sebuah *tool* yang sempurna untuk menganalisis masalah dari sistem. Sebagai *tool* untuk menganalisis, pemrogram tidak ingin untuk mulai memecahkan masalah di level teknis dengan membuat class, tetapi dengan menggunakan *activity diagram* untuk mengerti masalah dan menyaring proses yang terdapat dalam sistem. Berikut adalah *activity diagram* yang ada di sistem penjadwalan acara TI UMY berikut ini adalah *activity diagram web* penjadwalan acara :

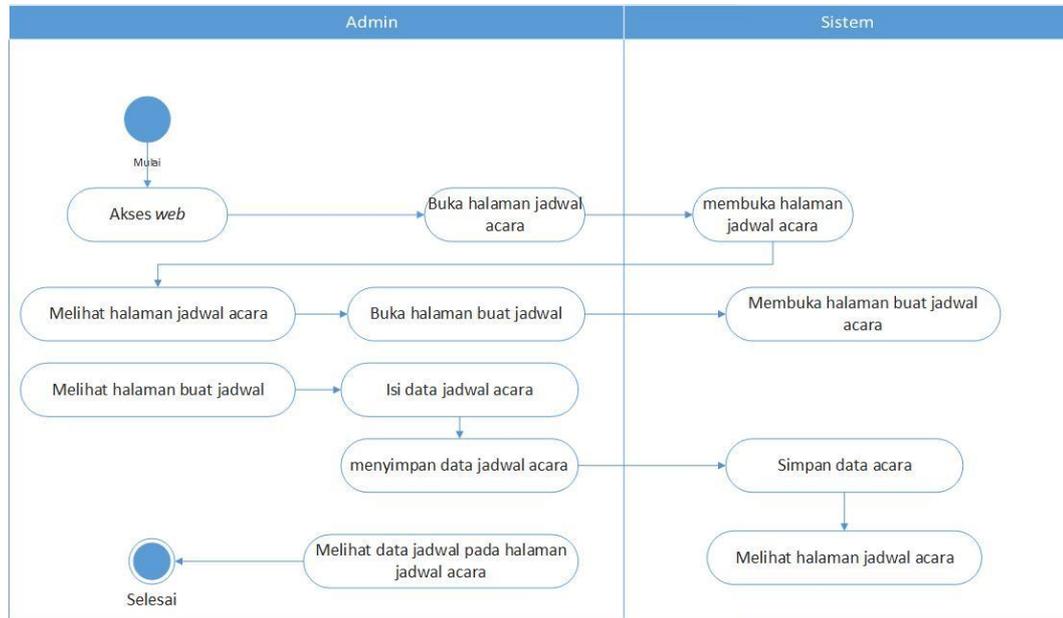
a. *Activity Diagram Login*



**Gambar 3.4** *Activity Diagram Login*

Berdasarkan *activity diagram login* admin harus mengakses web setelah diakses, admin berada di menu *login* di menu *login* admin harus memasukkan *username* dan *password* dengan benar, jika salah maka admin akan kembali ke menu *login* dan jika benar maka admin akan melihat manu *dashboard* halaman utama pada *web*.

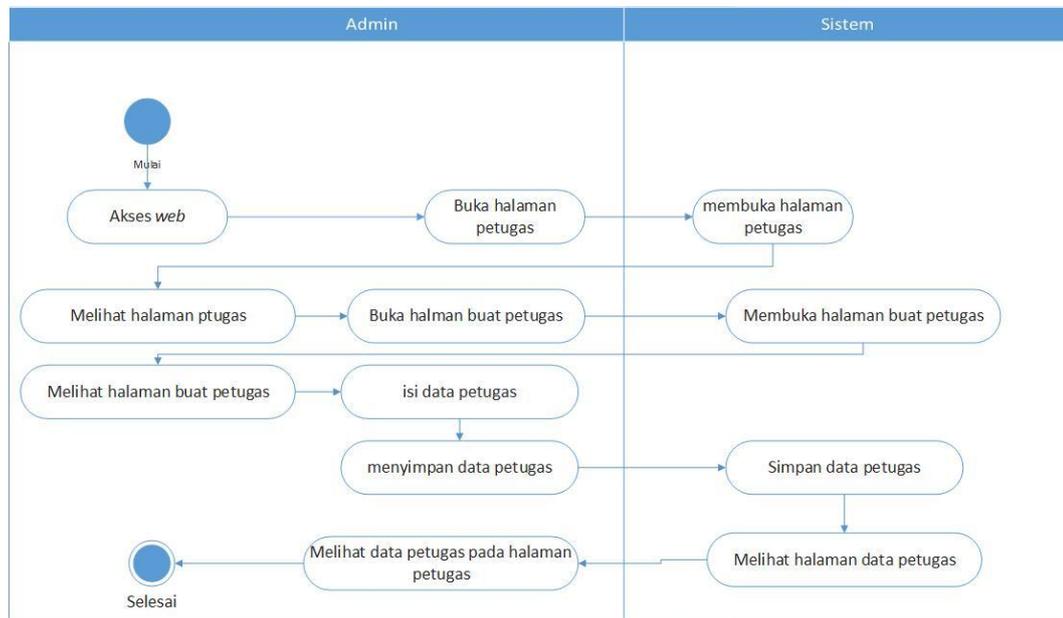
b. *Activity Diagram* Buat Jadwal Acara



**Gambar 3.5** *Activity Diagram* Buat Jadwal Acara

Berdasarkan *activity diagram* buat jadwal acara ketika admin ingin melakukan buat jadwal acara baru maka admin harus masuk kedalam akun admin lalu admin memilih menu buat jadwal acara maka akan tampil menu buat jadwal acara dan admin mengisi data jadwal acara jika jadwal acara diisi dengan benar pilih simpan maka jadwal acara akan tersimpan dan admin kembali melihat jadwal acara.

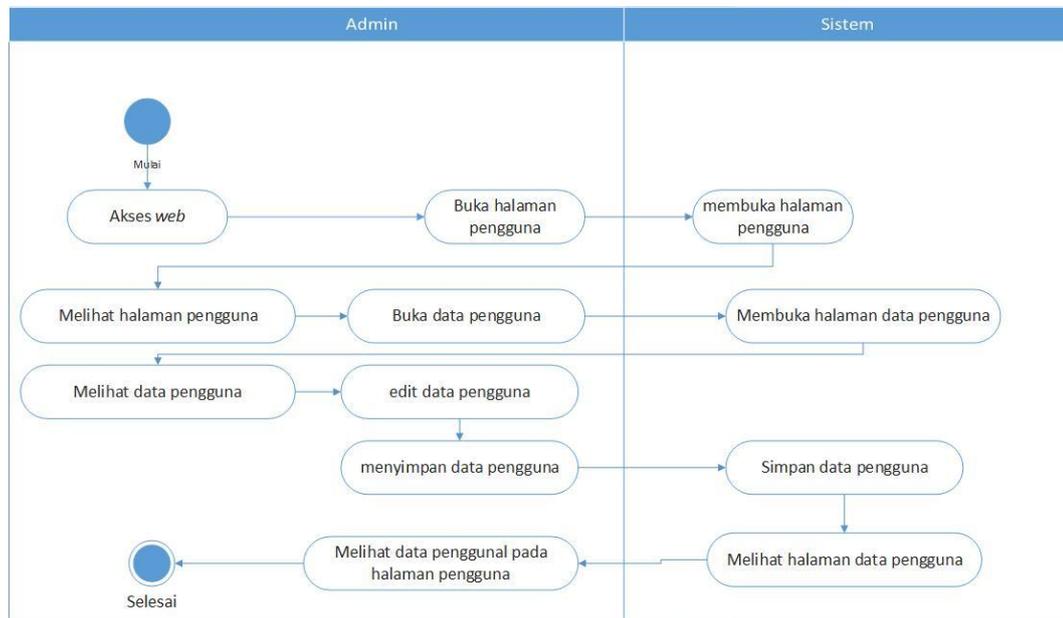
c. *Activity Diagram* Petugas



**Gambar 3.6** *Activity Diagram* Petugas

Berdasarkan *activity diagram* petugas ketika admin ingin melakukan buat petugas maka admin harus login agar dapat mengakses menu petugas setelah itu admin masuk ke menu petugas dan masuk ke halaman buat petugas lalu mengisi data petugas setelah melakukan isi data dengan benar memilih simpan maka data petugas disimpan dan admin akan kembali melihat menu petugas.

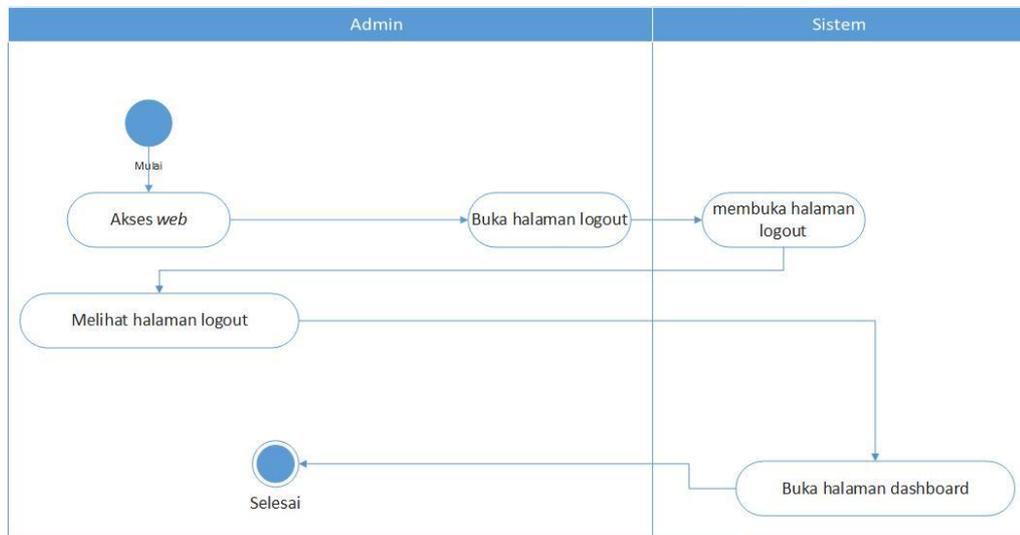
d. *Activity Diagram Pengguna*



**Gambar 3.7** *Activity Diagram Pengguna*

Berdasarkan *activity diagram* pengguna ketika admin ingin melakukan *update* data pengguna maka admin harus login dengan akun admin lalu admin dapat mengakses menu petugas lalu memilih menu data petugas dan *web* akan menampilkan menu petugas lalu admin memilih data petugas yang ingin di *update* setelah *update* data petugas pilih simpan maka data petugas tersimpan dan admin akan melihat ke menu petugas.

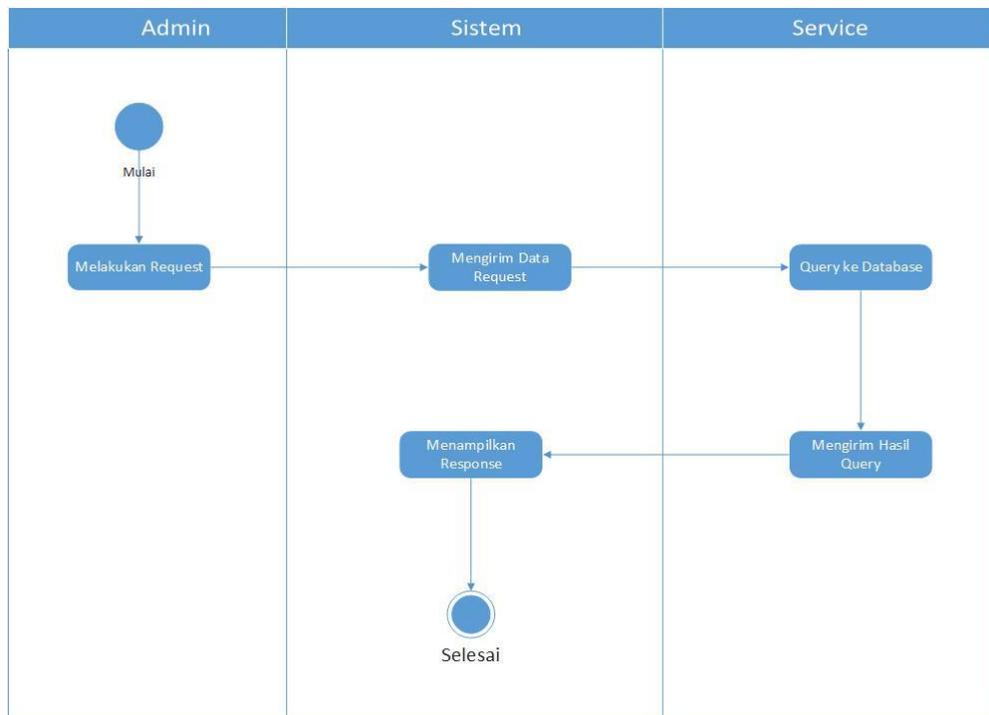
e. *Activity Diagram Logout*



**Gambar 3.8** *Activity Diagram Logout*

Berdasarkan *activity diagram logout* ketika admin *logout* maka admin harus melakukan *login* dengan akun admin lalu admin memilih menu *logout* setelah itu maka admin akan kembali melihat menu *dashboard* tanpa keterangan *login* dari akun petugas.

f. *Activity Diagram Web Service*



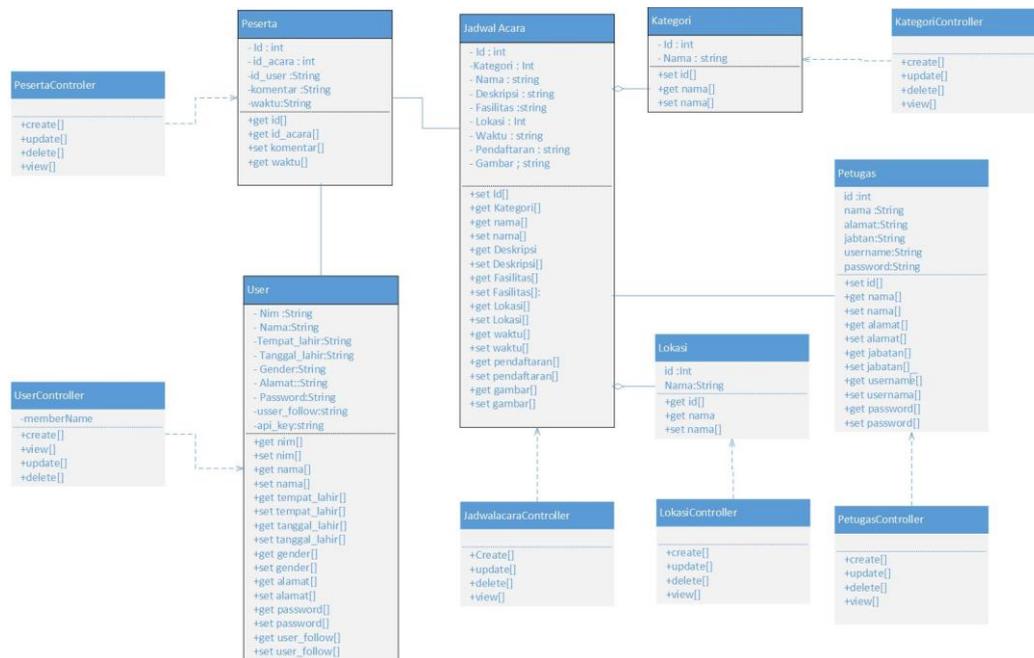
**Gambar 3.9** *Activity Diagram Web Service*

Berdasarkan *activity diagram web service* diatas yang ditunjukkan pada gambar 3.9 dapat dilihat proses dari setiap aktor yang ada di dalam sistem tersebut adalah admin yang memiliki akses *request service* seperti melihat peserta, komentar dan pengguna ke halaman *website*. Kemudian hasil *request* tersebut akan di publikasikan ke *website* tadi kemudian akan menampilkan *response* dari hasil *request* admin tersebut sebagai pengirim pesannya adalag JSON.

### 3.6.3 Class Diagram

*Class Diagram* adalah jenis statis diagram struktur yang menggambarkan struktur dari suatu sistem dengan menunjukkan kelas sistem, atribut, operasi (atau metode), dan hubungan antara objek-objek. Untuk lebih jelasnya ada pada gambar

3.9



**Gambar 3.10** *Class Diagram*

Berikut ini adalah penjelasan fungsi dan relasi *class diagram* pada gambar

3.9 :

1. *Class* Jadwal\_acara berfungsi untuk *edit* data, simpan data, hapus data dan cari data.
2. *Class* Kategori berfungsi untuk menyimpan data dari jenis acara.
3. *Class* Lokasi berfungsi untuk menyimpan data dari tempat acara.

4. *Class* Peserta berfungsi untuk menyimpan data peserta yang hadir pada acara
5. *Class* User berfungsi untuk menyimpan data peserta yang sudah terdaftar
6. *Class* jadwal acara controller berfungsi untuk opration pada jadwal acara
7. *Class* lokasi controller berfungsi untuk opration pada lokasi.
8. *Class* kategori controller berfungsi untuk opration pada kategori
9. *Class* peserta controller berfungsi untuk opration pada peserta
10. *Class* user controller berfungsi untuk opration pada user

Berikut adalah penjelasan realasi *class diagram* pada gambar 3.10

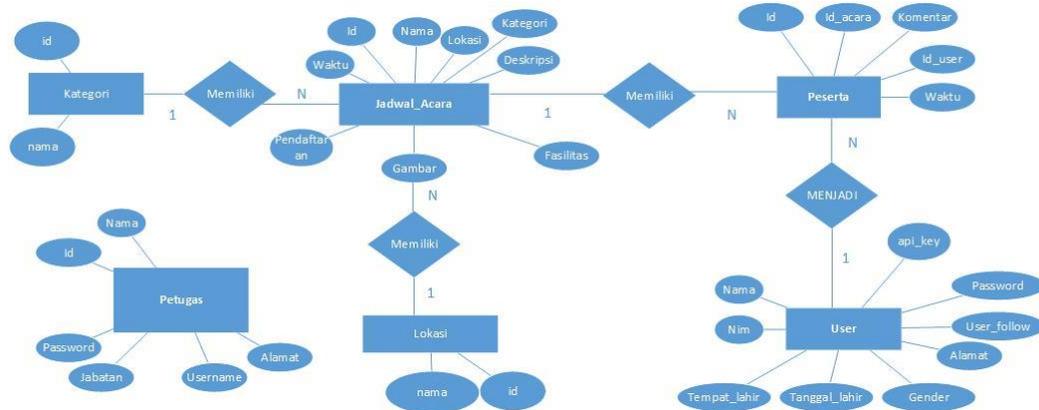
1. *Class* jadwal\_acara memiliki relasi *ccomposition* dengan *class* kategori, lokasi dan peserta karena data yang ditampilkan berdasarkan *method* yang ada pada *class* jadwal\_acara.
2. *Class* petugas memiliki relasi *association* dengan *class* jadwal\_acara karena petugas dapat mengakses data dari *class* jadwal\_acara.
3. *Class* peserta memiliki relasi *association* dengan *class* jadwal\_acara karena peserta dapat mengakses jadwal acara dengan melihat *detail* acara.
4. *Class* user memiliki relasi *association* dengan peserta karena user merupakan bagian dari peserta.
5. *Class* jadwal acara controller memiliki relasi dependency berfungsi untuk opration pada jadwal acara

6. *Class* lokasi controller memiliki relasi dependency berfungsi untuk opration pada lokasi
7. *Class* user controller memiliki relasi dependency berfungsi untuk opration pada user
8. *Class* kategori controller memiliki relasi dependency berfungsi untuk opration pada kategori
9. *Class* petugas controller memiliki relasi dependency berfungsi untuk opration pada petugas
10. *Class* peserta controller memiliki relasi dependency berfungsi untuk opration pada peserta

### **3.7 Perancangan Basis Data**

#### **3.7.1 *Entity Relationship Diagram (ERD)***

Gambar 3.6 menjelaskan hubungan relasi antar data dalam basis data yang terdapat di dalam sistem penjadwalan acara TI UMY berikut gambar *Entity Relationship Diagram*



**Gambar 3.11** Entity Relatinship Diagram (ERD)

Berdasarkan *entity relationship diagram* pada gambar 3.11 basis data yang dirancang memiliki 6 entitas yaitu :

- a. Kategori
- b. Jadwal\_acara
- c. Peserta
- d. User
- e. Lokasi
- f. Petugas

Berikut penjejas tentang relasi antar entitas yang tertera pada Gambar 3.11

:

1. Entitas kategori memiliki relasi *one to many* dengan entitas jadwal\_acara yang merupakan setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih pada tabel ke dua.

2. Entitas `jadwal_acara` memiliki relasi *one to many* dengan entitas peserta yang merupakan setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih pada tabel ke dua.
3. Entitas peserta memiliki relasi *one to many* dengan entitas user yang merupakan setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih pada tabel ke dua.
4. Entitas lokasi memiliki relasi *one to many* dengan entitas `jadwal_acara` yang merupakan setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih pada tabel ke dua.

### 3.7.2 Database (Kamus Data)

#### 1. Jadwal\_acara

Pada tabel `jadwal_acara` berfungsi untuk menyimpan detail acara yang akan dibuat pada penjadwalan acara yang dapat dilihat pada tabel 3.1

**Tabel 3.1** Jadwal\_acara

No	Nama	Tipe	Panjang	Keterangan
1	Id	Int	11	<i>Primary key</i>
2	Kategori	Int	11	
3	Nama	Varchar	50	
4	Deskripsi	Varchar	200	Indeks
5	Waktu	Datetime		
6	Fasilitas	Varchar	30	Indeks
7	Lokasi	Int	11	

No	Nama	Tipe	Panjang	Keterangan
8	Pendaftaran	Varchar	30	
9	Gambar	Varchar	50	

## 2. Kategori

Pada tabel katogori berfungsi untuk menyimpan *detail* kategori acara yang ada pada jadwal acara, dapat dilihat pada tabel dibawah ini

**Tabel 3.2** Kategori

NO	Nama	Tipe	Panjang	Keterangan
1	Id	Int	11	<i>Primary key</i>
2	Nama	Varchar	50	

## 3. Lokasi

Pada tabel lokasi berfungsi untuk menyimpan *detail* lokasi yang ada pada jadwal acara, dapat dilihat pada tabel dibawah ini

**Tabel 3.3** Lokasi

NO	Nama	Tipe	Panjang	Keterangan
1	Id	Int	11	<i>Primary key</i>
2	Nama	varchar	50	

#### 4. Peserta

Pada tabel peserta berfungsi untuk menyimpan detail peserta yang mengikuti acara dapat dilihat pada tabel 3.2

**Tabel 3.4** Tabel Peserta

No	Nama	Tipe	Panjang	Keterangan
1	Id	Int	11	<i>Primary key</i>
2	Id_acara	Int	11	
3	Id_user	Brigint	20	
4	Komentar	Varchar	160	Indeks
5	Waktu	Datetime		

#### 5. Petugas

Pada tabel petugas berfungsi untuk menyimpan data petugas yang bertugas untuk melakukan penjadwalan acara yang dapat dilihat pada tabel 3.3

**Tabel 3.5** Tabel Petugas

No	Nama	Tipe	Panjang	Keterangan
1	Id	Int	11	<i>Primary key</i>
2	Nama	Varchar	50	
3	Alamat	Varchar	160	
4	Jabatan	Varchar	50	
5	Username	Varchar	30	
6	Password	Varchar	32	

## 6. *User*

Pada tabel *user* berfungsi untuk menyimpan data *user* yang terdaftar pada *web* penjadwalan acara dapat dilihat pada tabel 3.4

**Tabel 3.6** Tabel User

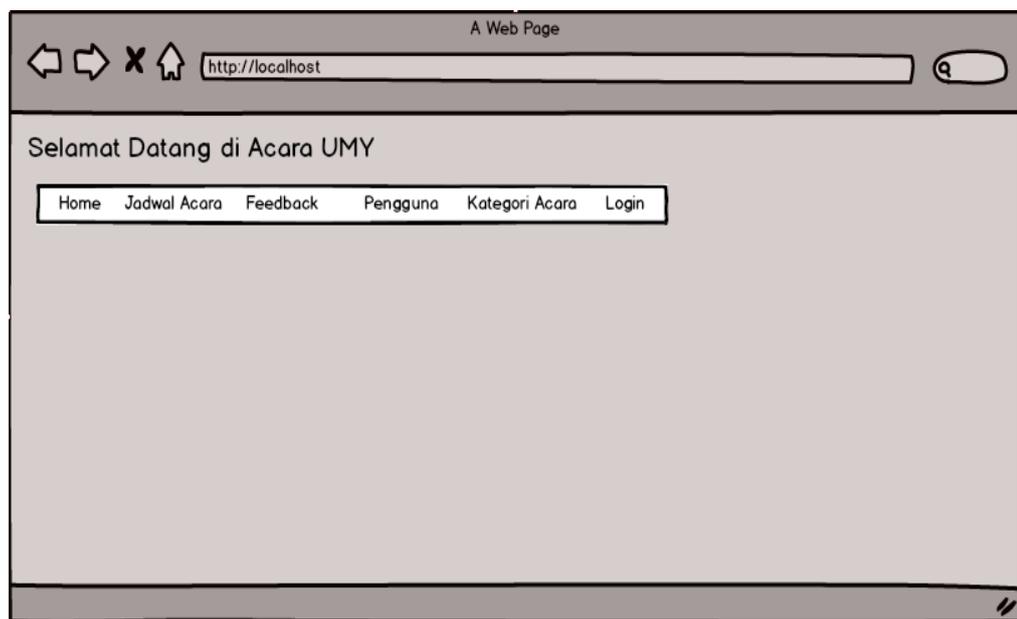
No	Nama	Tipe	Panjang	Keterangan
1	Nim	Brigint	20	<i>Primary key</i>
2	Nama	Varchar	50	
3	Tempat_lahir	Varchar	150	
4	Tanggal_lahir	Date		
5	Gender	Varchar	10	
6	Alamat	varchar	150	Indeks
7	User_follow	Varchar	150	
8	Password	Varchar	32	
9	Api_key	Varchar	32	

### 3.8 Perancangan *Interface*

Perancangan *interface* sistem diperlukan untuk memudahkan *admin* dalam melakukan proses interaksi terhadap sistem. *Interface* menyediakan tampilan halaman sebuah sistem yang digunakan untuk proses *input* hingga menghasilkan *output* yang sesuai dengan kebutuhan. *Interface* untuk sistem yang akan dibuat adalah sebagai berikut.

### 3.8.1 Rancangan *Menu Home*

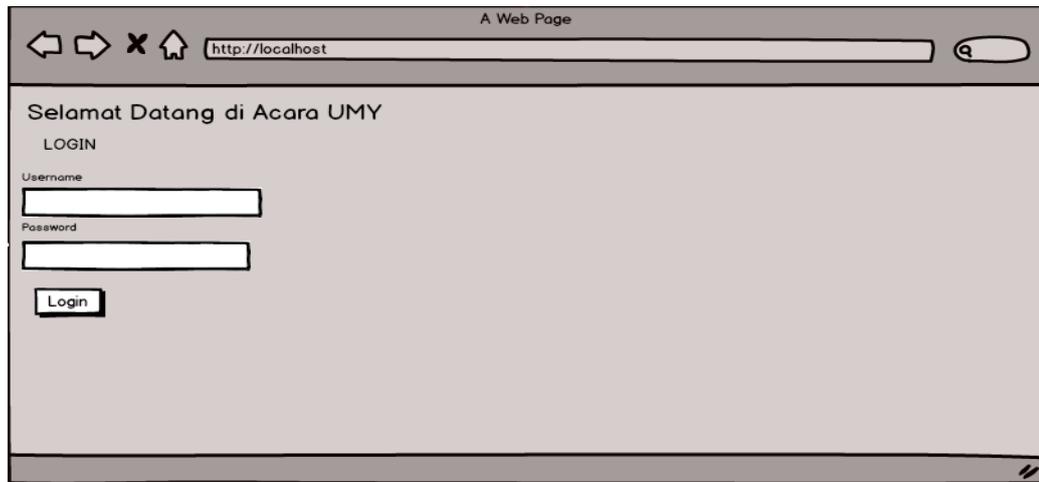
Rancangan antarmuka halaman awal merupakan tampilan awal pada saat aplikasi dijalankan. Gambaran rancangan antarmuka halaman awal dapat dilihat pada gambar 3.0. Pada halaman terdapat beberapa menu dimana pengguna harus melakukan login untuk menjalankan menu yang di aplikasi.



**Gambar 3.12** Rancangan *Menu Home*

### 3.8.2 Rancangan Halaman *Login*

Rancangan awal untuk halaman *login* akan dibuat simpel, hanya akan ada *text box* untuk mengisi *username* dan *password*. Terdapat satu *button* bernama *login* untuk memberikan perintah masuk ketika pengguna sudah dan mengisikan *username* dan *password* dapat dilihat pada gambar 3.12



A Web Page

http://localhost

Selamat Datang di Acara UMY

LOGIN

Username

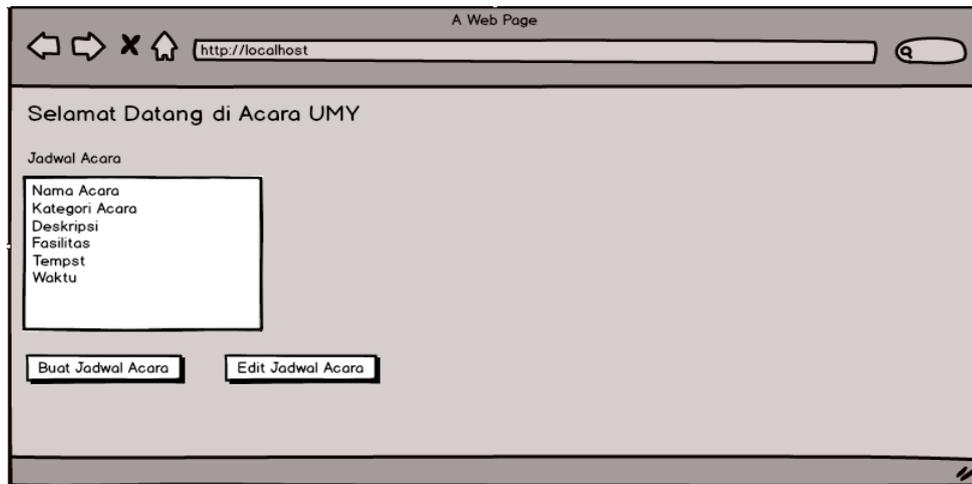
Password

Login

**Gambar 3.13** Rancangan Halaman *Login*

### 3.8.3 Rancangan Halaman Jadwal Acara

Rancangan awal untuk menu halaman jadwal acara merupakan tampilan beberapa jadwal yang sudah dibuat sebelumnya oleh pengguna, pada tampilan ini terdapat penjelasan dari kategori acara, deskripsi, fasilitas, waktu dan tempat dan ada dua *button* yaitu *button* buat jadwal acara dan *button* edit jadwal acara dimana akan dijelaskan pada gambar rancangan awal buat jadwal acara dan rancangan awal *edit*.



**Gambar 3.14** Rancangan Awal Halaman Jadwal Acara

#### **3.8.4 Rancangan Halaman Buat Jadwal Acara**

Rancangan halaman buat jadwal acara merupakan tampilan awal aplikasi ketika pengguna ingin membuat jadwal acara, pada rancangan halaman ini pengguna akan mengisikan beberapa *text box* untuk menjelaskan acara apa yang dibuat dan deskripsi lainnya dari acara tersebut seralah itu pengguna dapat menekan *button* simpan untuk menyimpan jadwal acara yang sudah dibuat.

A Web Page

http://localhost

Selamat Datang di Acara UMY

Buat Jadwal Acara

Kategori Acara ▾

Nama acara

Deskripsi

Fasilitas

Lokasi

Waktu

Simpan

**Gambar 3.15** Rancangan Halaman Buat Jadwal Acara

### 3.8.5 Rancangan Halaman Edit Jadwal

Rancangan halaman *edit* jadwal merupakan halaman dimana pengguna dapat melakukan perubahan terhadap deskripsi acara yang sudah dibuat pada halaman ini pengguna dapat melakukan pencarian dan hapus acara yang diinginkan.

A Web Page

http://localhost

Selamat Datang di Acara UMY

Edit Jadwal Acara

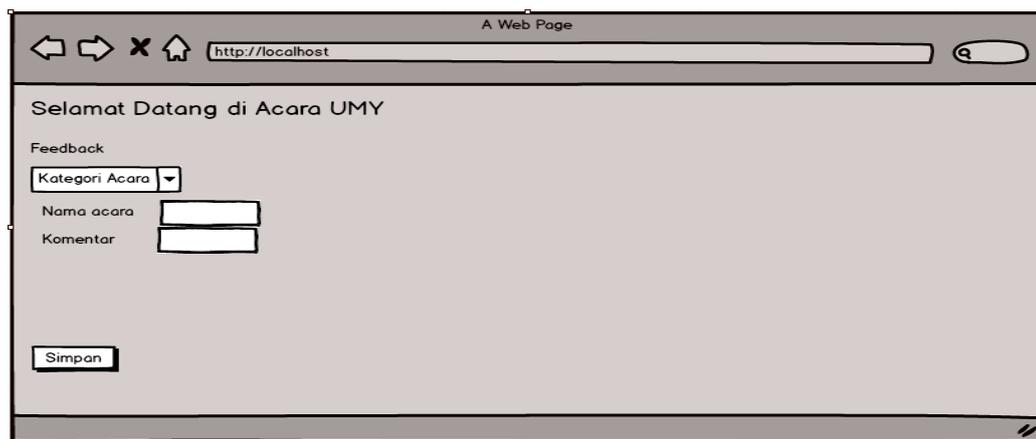
Nama acara Kategori Deskripsi fasilitas tempat waktu 🔍 ✎ 🗑️

Simpan

**Gambar 3.16** Rancangan Halaman *Edit* Jadwal

### 3.8.6 Rancangan Halaman *Feedback*

Rancangan halaman *feedback* merupakan halaman dimana pengguna dapat melihat penilaian acara yang sudah dilaksanakan dengan cara memberikan komentar di halaman ini



The image shows a web browser window titled "A Web Page" with the address bar displaying "http://localhost". The main content area of the browser contains the following elements:

- Greeting: "Selamat Datang di Acara UMY"
- Section Header: "Feedback"
- Form Fields:
  - "Kategori Acara" with a dropdown arrow.
  - "Nama acara" with a text input field.
  - "Komentar" with a text input field.
- Submit Button: "Simpan"

**Gambar 3.17** Rancangan Halaman *Feedback*

### 3.8.7 Halaman Rancangan Pengguna

Rancangan halaman pengguna merupakan halaman dimana pengguna dapat melihat siapa aja pengguna yang sudah terdaftar pada aplikasi *web* penjadwalan acara TI UMY.

A Web Page

http://

Selamat Datang di Acara UMY

Pengguna

Nim Nama    Alamat    Jenis Kelamin

[Large empty text area]

Edit

**Gambar 3.18** Halaman Rancangan Pengguna

### 3.8.8 Halaman Rancangan Kategori Acara

Rancangan halaman Kategori acara merupakan halaman dimana pengguna dapat melihat kategori acara apa saja yang sudah ada dan untuk membuat kategori acara dapat dilakukan dengan cara pengguna menekan tombol *edit* yang dapat dilihat gambar selanjutnya.

A Web Page

http://

Selamat Datang di Acara UMY

Kategori Acara    Q search

Id                    [Input field]

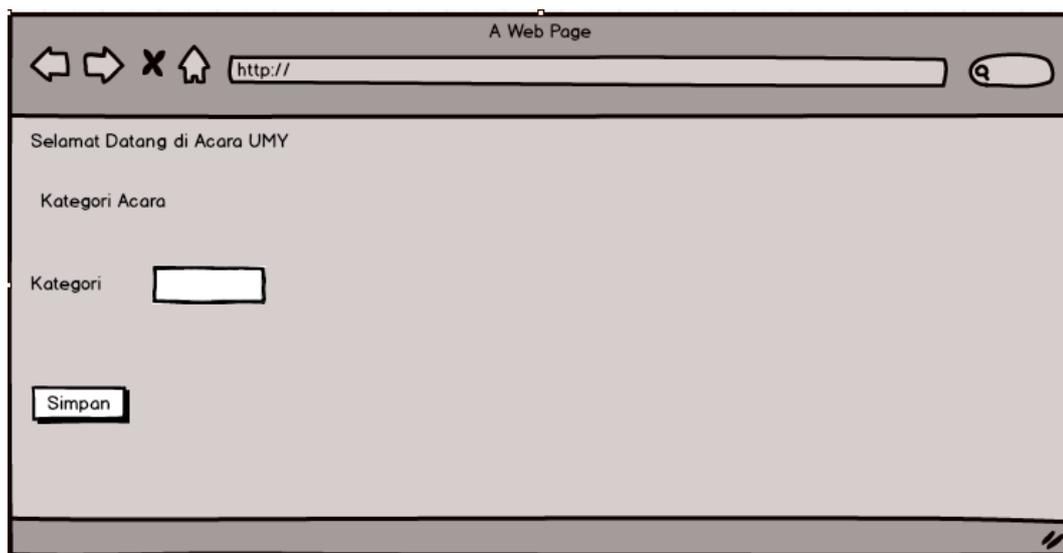
Kategori            [Input field]

Edit

**Gambar 3.19** Halaman Rancangan Kategori Acara

### 3.8.9 Halaman Rancangan Buat Kategori Acara

Rancangan halaman buat kategori acara merupakan dimana pada halaman ini pengguna dapat membuat kategori dimana setelah pengguna membuat kategori acara baru maka kategori akan muncul di dalam menu buat jadwal acara.



The image shows a browser window with the title "A Web Page". The address bar contains "http://". The main content area of the page has a light gray background and contains the following elements:

- Greeting: "Selamat Datang di Acara UMY"
- Section Header: "Kategori Acara"
- Form Label: "Kategori" followed by a text input field.
- Form Button: "Simpan" (Save).

**Gambar 3.20** Halaman Rancangan Buat Kategori Acara

### 3.9 Metode Pengujian

Pengujian perangkat lunak merupakan suatu kegiatan yang dilakukan untuk mendapatkan informasi serta mengevaluasi kualitas dari produk atau layanan yang sedang diuji. Metode pengujian yang dipakai dalam pengembangan aplikasi adalah *black box testing* yaitu dimana yang akan dilakukan hanya mengamati hasil eksekusi dan memeriksa fungsional dari aplikasi yang dikembangkan.

Beberapa hal yang diujikan dalam sistem aplikasi yaitu apakah sesuai dengan tujuan awal dikembangkan sistem, tujuan pengembangan sistem adalah:

1. Aplikasi dapat membuat jadwal acara dengan keinginan dan kebutuhan *user*.
2. Aplikasi dapat melakukan *update* dengan keinginan *user*.
3. Aplikasi dapat menampilkan daftar komentar dari jadwal acara yang sudah terlaksana.
4. Aplikasi dapat memberikan informasi kepada pengguna yang sudah terdaftar.
5. Aplikasi dapat membuat kategori acara dengan kebutuhan *user*.
6. Aplikasi dapat memberikan informasi peserta acara yang mengikuti kegiatan acara.