

## **BAB III**

### **CARA DAN METODOLOGI PENELITIAN**

#### **3.1. Alat dan Bahan Penelitian**

Dalam melakukan penelitian ini, peneliti menggunakan beberapa perangkat-perangkat yang mendukung dalam melakukan pengembangan dan implementasi *Website* yaitu antara lain:

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

*Personal Computer (PC)/ Laptop :*

- a. *64 bit architecture processor*
- b. *8 GB Random Access Memmory (RAM)*

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

###### **a. Sistem Operasi Windows 10**

Windows 10 adalah sistem operasi komputer pribadi yang dikembangkan oleh Microsoft sebagai bagian dari keluarga sistem operasi Windows NT. Windows 10 berjalan optimal pada prosesor Intel Core dengan keamanan dan interaksi pribadi yang lebih tinggi.

###### **b. XAMPP**

XAMPP berfungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari beberapa program yaitu : *Apache HTTP Server, MySql Database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License and free*, yang merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis.

**c. *VSCode***

VSCode sebagai editor kode efisien yang didukung untuk operasi pengembangan seperti *debugging*, *task running* dan *version control*. VSCode bertujuan untuk menyediakan alat yang dibutuhkan pengembang untuk siklus cepat dalam membangun kode-debug dan memberikan alur kerja yang lebih kompleks ke IDE berfitur lengkap.

**d. *MySql***

MySql digunakan sebagai pengelola *Database* beserta isinya dengan menggunakan beberapa perintah-perintah yang ada ada di MySql.

**e. *Bootstrap***

Bootstrap digunakan sebagai alat untuk membuat sebuah tampilan halaman website yang dapat mempercepat pekerjaan seorang pengembang website ataupun pendesain website.

**3.2. Tempat dan Waktu Penelitian**

Dalam melakukan penelitian ini, tempat yang dijadikan sebagai objek penelitian peneliti yaitu Fakultas Teknik Universitas Muhammadiyah Yogyakarta. Penelitian ini dilaksanakan dari bulan april sampai agustus 2018.

**3.3. Subject Penelitian**

Berdasarkan dengan informasi yang dihasilkan dalam penelitian ini, maka yang ditetapkan sebagai subjek penelitian ini adalah *Website* yang ada di Universitas Muhammadiyah Yogyakarta serta user yang akan terlibat dengan *Website* yang akan dikembangkan nantinya.

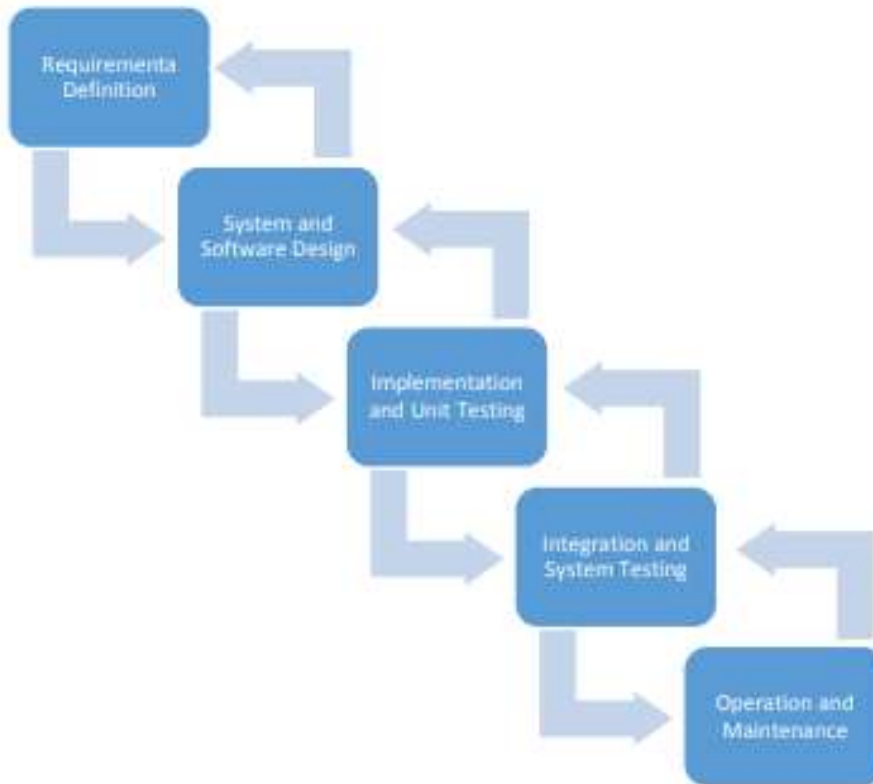
### 3.4. Metodologi Penelitian

Penelitian pembuatan sistem Penanganan Keluhan Berbasis Web dalam studi kasus Fakultas Teknik Universitas Muhammadiyah Yogyakarta menggunakan metode *Software Development Life Cycle (SDLC)* model *Waterfall*. Metode SDLC dengan model *Waterfall* dipilih dengan alasan untuk memudahkan peneliti dalam menentukan pencapaian suatu sistem.

Model *Waterfall* dengan pendekatan SDLC merupakan awal dari peneliti dalam mengembangkan perangkat lunak yang dibuat. Hal ini juga disebut sebagai model SDLC *linear-sekuensial*. Hal ini juga mempermudah peneliti dalam memahami dan mengimplementasikannya dalam sebuah sistem.

Dalam model *Waterfall*, setiap tahapan harus berurutan dan tidak dapat melompat ke tahap berikutnya yang artinya bahwa peneliti tidak dapat lanjut ke tahap kedua ataupun seterusnya jika belum menyelesaikan proses pada tahap pertama. Hal ini juga dijadikan sebuah pertimbangan oleh peneliti dalam membuat sistem Penanganan Keluhan Berbasis Web ini, karena sistem ini hanya dikerjakan oleh satu orang dan untuk mempermudah pembuatan sistem maka peneliti harus mengembangkan sistem dengan tahap yang berurutan juga.

Tahapan pengembangan sistem dengan menggunakan model *Waterfall* dengan alur SDLC bisa dilihat pada Gambar 2.4.



**Gambar 2.4** Model *Waterfall* dengan alur SDLC

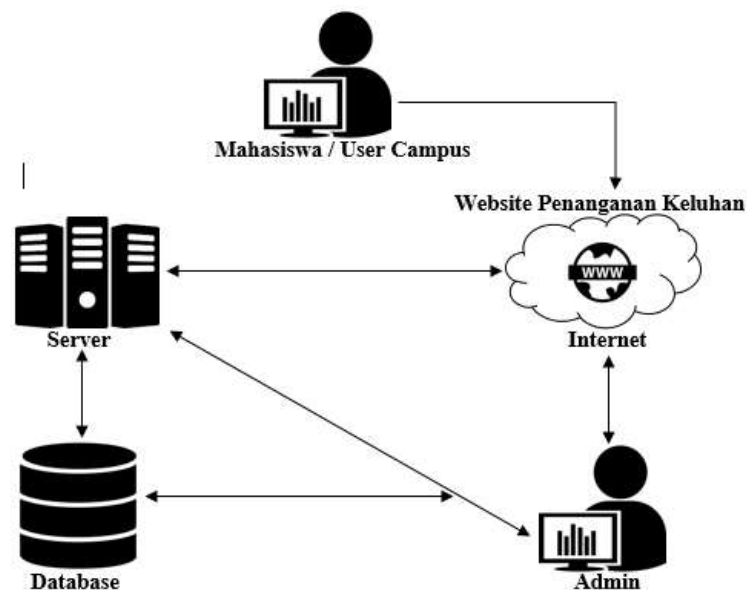
1. *Requirements Definition* – merupakan tahap perencanaan sistem yang menyangkut kebutuhan pengguna, kelayakan baik secara teknik maupun secara teknologi. Setelah segala kebutuhan pengguna telah terkumpul secara lengkap lalu dilakukan proses analisis, dari proses analisis inilah dapat didefinisikan kebutuhan mana yang harus dipenuhi oleh sistem yang akan dibangun. Tahap ini harus dikerjakan secara menyeluruh dan lengkap untuk dapat menghasilkan keluaran (*output*) yang lengkap yang diinginkan pengguna.
2. *System and Software Design* – merupakan proses perancangan sistem mulai dari pembagian persyaratan dalam sistem perangkat keras maupun perangkat lunak. Pada tahap ini ditentukan arsitektur sistem



secara keseluruhan. Proses perancangan melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar. Tahap ini dikerjakan setelah proses tahap sebelumnya yaitu kebutuhan pengguna telah selesai dikumpulkan secara lengkap.

3. *Implementation and Unit Testing* – merupakan proses perancangan perangkat lunak direalisasikan dengan program atau unit program. Proses pengujian ini melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.
4. *Integration and System Testing* – merupakan proses pengujian unit program atau program individual yang diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah terpenuhi.
5. *Operation and Maintenance* – merupakan proses pengoperasian sistem pada lingkungannya dan melakukan pemeliharaan sistem. Tahap ini merupakan tahap siklus hidup yang lama. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan pada tahap-tahap sebelumnya yang kemudian dilakukan perbaikan kembali atas implementasi unit sistem dan pengembangan layanan sistem serta persyaratan-persyaratan baru yang ditambahkan.

### 3.5. Arsitektur


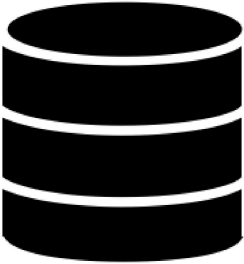



**Gambar 3.1** Arsitektur Sistem Penanganan Keluhan

*Database server* yang digunakan adalah MySQL sebagai pengelola data yang terdapat dalam Website Penganan Keluhan dan *Web Server* yang digunakan adalah Xampp. Komunikasi antar *pengguna* dan *web server* menggunakan *internet* dan *web browser* pada perangkat pengguna. Saat pengguna mengakses aplikasi, *web server* akan memuat antarmuka dan melakukan pengambilan data yang diperlukan dari *database server*. Melalui antarmuka yang dimuat *web server*, pengguna bisa menyimpan data ke *database server*.

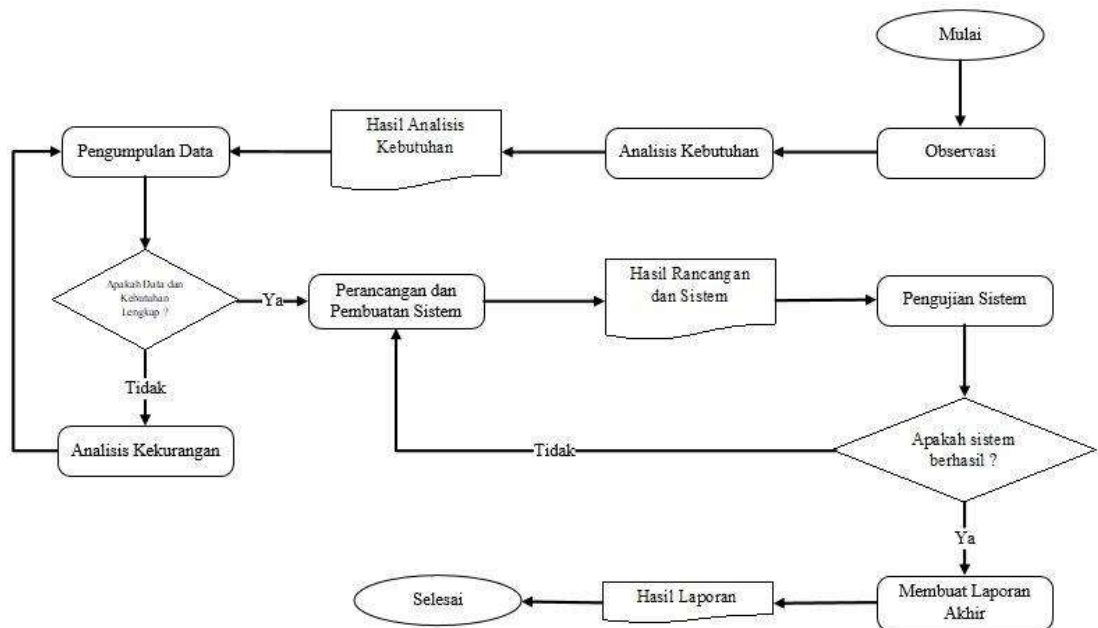
Dari arsitektur yang telah dibuat sebelumnya, terdapat beberapa komponen yang digunakan dalam arsitektur Website Penanganan Keluhan. Komponen arsitektur tersebut yang akan digunakan dalam pengembangan sistem Penanganan Keluhan nantinya, yang akan disajikan dalam bentuk data dalam database sistem. Komponen Arsitektur Sistem Penanganan Keluhan dapat dilihat pada Tabel 3.1.

**Tabel 3.1** Komponen Arsitektur Sistem Penanganan Keluhan

Nama	Gambar	Keterangan
	<b>USER</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa / User Campus</li> <li>2. Admin</li> </ol>
	<b>DATABASE</b>	<p>Berisi data-data sebagai berikut :</p> <ol style="list-style-type: none"> <li>1. Data Petugas</li> <li>2. Data Group Akses</li> <li>3. Data Mahasiswa</li> <li>4. Data Prodi</li> <li>5. Data Fakultas</li> <li>6. Data Kategori</li> <li>7. Data Pengaduan</li> <li>8. Data tahun</li> <li>9. Data setting</li> </ol>
	<b>SERVER</b>	<p><i>Server</i> digunakan untuk mengelola semua data dalam sistem Website Penanganan Keluhan</p>

### 3.6. Teknik Pengumpulan Data

Dalam penelitian untuk sistem Penanganan Keluhan Berbasis *web* studi kasus Fakultas Teknik Universitas Muhammadiyah Yogyakarta ini, penulis menggunakan metode SDLC. Hal ini bertujuan untuk mengatasi kesalahan, kerusakan ataupun error yang terjadi dalam sistem dan cara mengatasinya adalah dengan menganalisis kembali kebutuhan dari awal untuk memperbaiki sistem. Tahap alur pengumpulan data yang dilakukan dalam penelitian Website Penanganan Keluhan ini terdapat dalam Gambar 3.2.



**Gambar 3.2** Tahap Teknik Pengumpulan Data

Selain itu, metode ini juga diambil karena sistem Penanganan Keluhan Berbasis *Web* ini dilakukan oleh satu peneliti, yang berarti bahwa metode SDLC sangat cocok dalam melakukan pengembangan sistem Penanganan Keluhan Berbasis *Web* ini. Tahap alur penelitian dilakukan penulis melalui beberapa tahap yaitu sebagai berikut:

### 3.6.1. Menganalisis Kebutuhan

Analisis kebutuhan merupakan tahap dasar dalam membangun sebuah sistem informasi. Analisis kebutuhan antara lain berupa analisis kebutuhan *Software* dan *Hardware*, analisis kebutuhan isi dan interaksi menu pada aplikasi. Sebelum pembuatan sistem Penangan Keluhan Berbasis *Web* ini, penulis terlebih dahulu melakukan sesi wawancara terhadap beberapa orang terkait penggunaan Website yang akan dibuat nantinya. Sesi wawancara yang dilakukan bertujuan untuk melengkapi data yang sebelumnya telah didapatkan penulis. Dalam sesi wawancara, partisipan akan diberikan beberapa pertanyaan mengenai fitur-fitur tambahan yang

diharapkan oleh pengguna yang ada didalam Website Penanganan Keluhan tersebut. Pertanyaan tersebut diajukan guna menghasilkan sistem yang diharapkan oleh para pengguna.

### **3.6.2. Pengumpulan data**

Pengumpulan data dilakukan berdasarkan data yang telah dianalisis pada tahap sebelumnya berdasarkan penggabungan data primer dan sekunder.

### **3.6.3. Menganalisis Kekurangan**

Analisis kekurangan dilakukan setelah semua data telah terkumpul dengan lengkap. Hal ini dilakukan guna meminimalisir ketidaksesuaian antara sistem dan kebutuhan pengguna nantinya. Analisis kekurangan juga dilakukan guna mengurangi terjadinya kesalahan dalam proses jalannya sistem nantinya.

### **3.6.4. Pengumpulan data kembali dan menentukan kebutuhan**

Pengumpulan data kembali dilakukan setelah beberapa proses sebelumnya telah dilakukan. Pengumpulan data kembali berguna untuk melengkapi data yang kurang setelah melalui tahap analisis kekurangan atas kebutuhan pengguna terhadap sistem Penanganan Keluhan ini. Setelah itu, dilakukanlah penentuan kebutuhan yang terakhir yang akan dimuat dalam sistem Penanganan Keluhan ini. Penentuan Kebutuhan terakhir ini dilakukan guna membuat sistem lebih terarah dan terstruktur. Selain itu, kebutuhan terakhir ini juga akan mempermudah dalam melakukan *testing* ketika sistem telah selesai dikembangkan.

### **3.6.5. Perancangan dan Pembuatan Sistem**

Setelah beberapa tahap sebelumnya telah dilakukan, maka selanjutnya penulis merancang desain atas antarmuka sistem yang akan dibuat dalam bentuk Website nantinya. Perancangan ini dilakukan guna menyesuaikan harapan pengguna atas apa yang

dibutuhkan dalam sistem Penanganan Keluhan tersebut baik dalam segi interface, fungsi, dan fitur-fitur yang terdapat dalam sistem.

### **3.6.6. Pengujian Sistem**

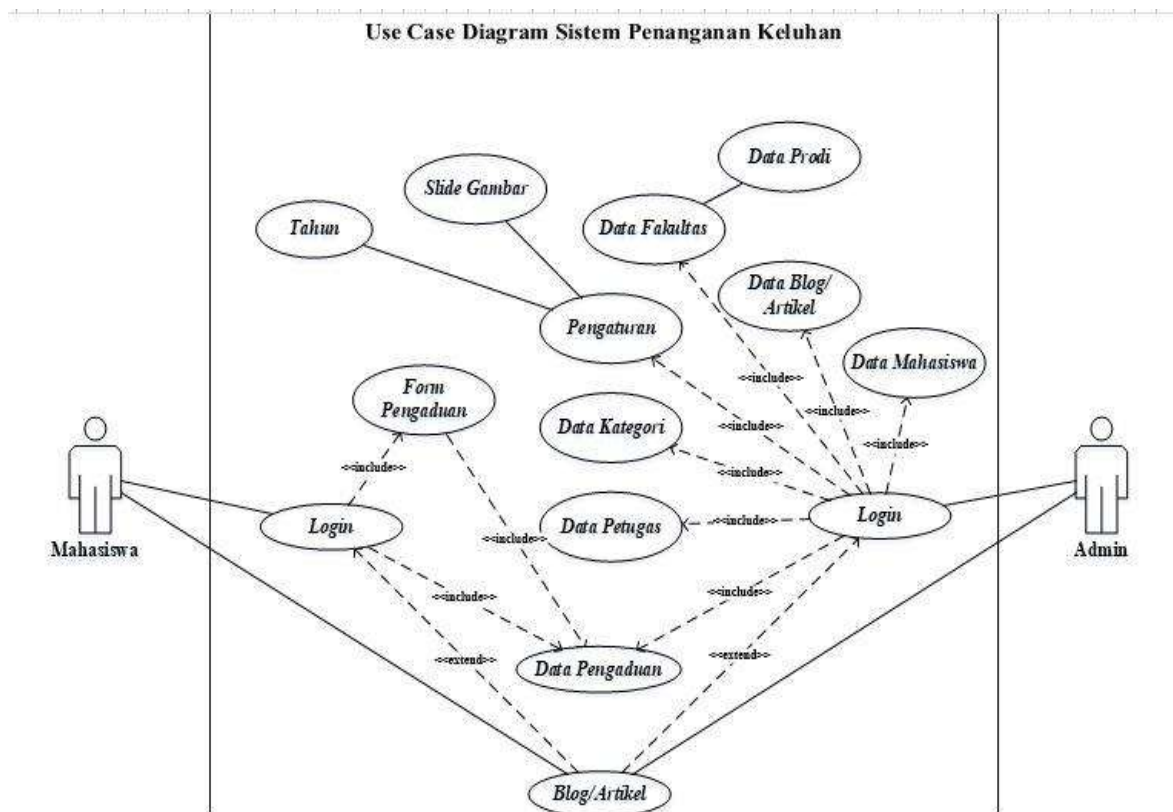
Pengujian sistem dilakukan setelah sistem yang telah dibuat sebelumnya tadi telah siap untuk dijalankan. Pengujian bertujuan untuk mengetahui sampai mana sistem yang telah dibuat tadi berfungsi dengan baik tanpa ada error yang tidak diharapkan terjadi saat sistem tersebut telah dipakai perdana oleh pengguna. Sistem akan diuji sesuai dengan kebutuhan yang telah ditentukan sebelumnya. Seluruh kebutuhan harus terpenuhi dalam sistem dan tidak ada yang tertinggal ataupun tidak sesuai dengan kebutuhan pengguna. Pengujian akan dilakukan dengan menggunakan metode *blackbox*.

## **3.7. Rancangan**

Dalam pembuatan sistem Website Penanganan Keluhan, dilakukan perancangan *database* menggunakan alat bantu Diagram ER. Selain itu, perancangan juga menggunakan metode UML (*Unified Markup Language*). Model UML yang digunakan dalam pengembangan sistem Website Penanganan Keluhan ini yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *ER Diagram*.

### **3.7.1. Use Case Diagram**

Gambaran Use Case Diagram yang dibuat untuk sistem *Penangan Keluhan Berbasis Web* dapat dilihat pada Gambar 3.3.



**Gambar 3.3** Use Case Diagram

Berikut ini penjelasan mengenai *Use Case Diagram* pada Gambar 3.3:

1. *Actor Mahasiswa* :

- Bisa mengakses *Blog* atau *Artikel* tanpa melakukan *Login* ataupun dengan melakukan *Login*.
- Tidak bisa mengakses *Form Pengaduan* ataupun melakukan proses *Pengaduan* tanpa melakukan *Login* terlebih dahulu.
- Tidak bisa melihat daftar *Data Pengaduan* tanpa melakukan *Login* terlebih dahulu.
- Bisa melihat langsung daftar *Data Pengaduan* setelah melakukan proses *Pengaduan* pada *Form Pengaduan* ataupun sebelum atau tanpa melakukan proses *Pengaduan* terlebih dahulu.

- Tidak bisa melihat *Jumlah Batasan Pengaduan* tanpa melakukan *Login* terlebih dahulu.

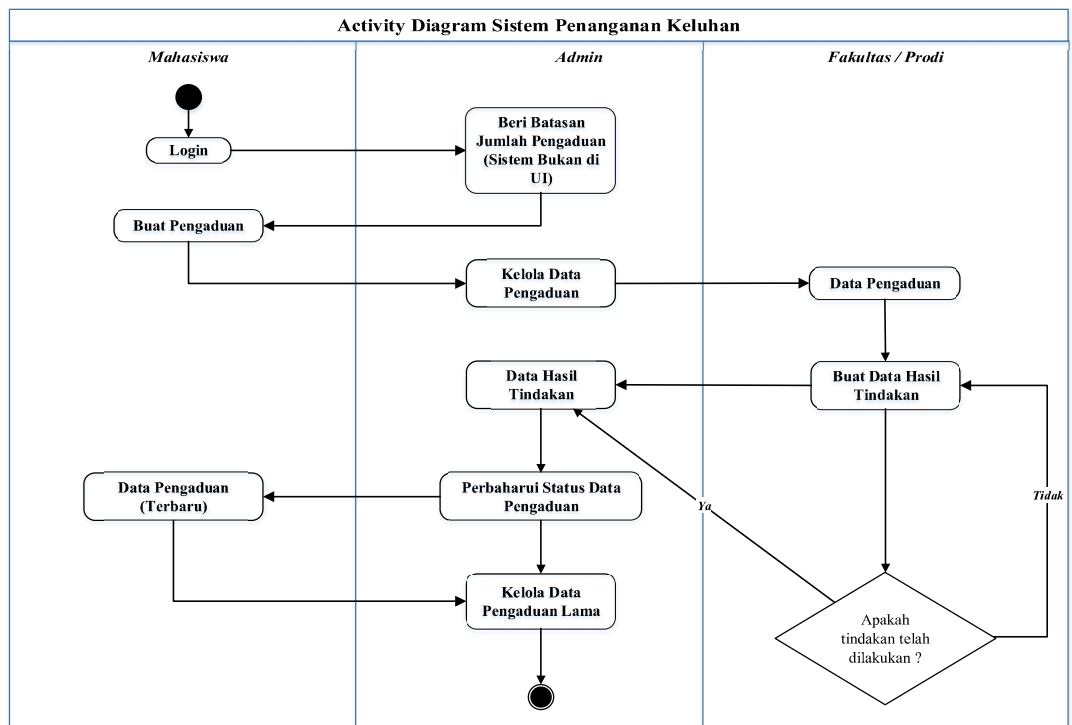
## 2. Actor Admin :

- Bisa mengakses *Blog* atau *Artikel* tanpa melakukan *Login* ataupun dengan melakukan *Login*.
- Tidak bisa mengakses ataupun mengelola *Data Pengaduan* tanpa melakukan *Login* terlebih dahulu.
- Tidak bisa mengakses ataupun mengelola *Data Petugas* tanpa melakukan *Login* terlebih dahulu.
- Tidak bisa mengakses ataupun mengelola *Data Mahasiswa* tanpa melakukan *Login* terlebih dahulu.
- Tidak bisa mengakses ataupun mengelola *Data Kategori* tanpa melakukan *Login* terlebih dahulu.
- Tidak bisa mengakses ataupun mengelola *Data Fakultas* tanpa melakukan *Login* terlebih dahulu.
- *Data Prodi* bisa diakses jika *Admin* telah mengakses *Data Fakultas* terlebih dahulu.
- Tidak bisa mengakses ataupun mengelola *Setting (pengaturan)* tanpa melakukan *Login* terlebih dahulu.
- *Setting (pengaturan)* terdiri *Setting Tahun* dan *Setting Slide Gambar*.

### 3.7.2. Activity Diagram

Gambaran Use Case Diagram yang dibuat untuk sistem *Penangan Keluhan Berbasis Web* dapat dilihat pada Gambar 3.4.





**Gambar 3.4** Activity Diagram

Berikut ini penjelasan *Activity Diagram* mengenai proses sistem *Penanganan Keluhan Berbasis Web* pada Gambar 3.4:

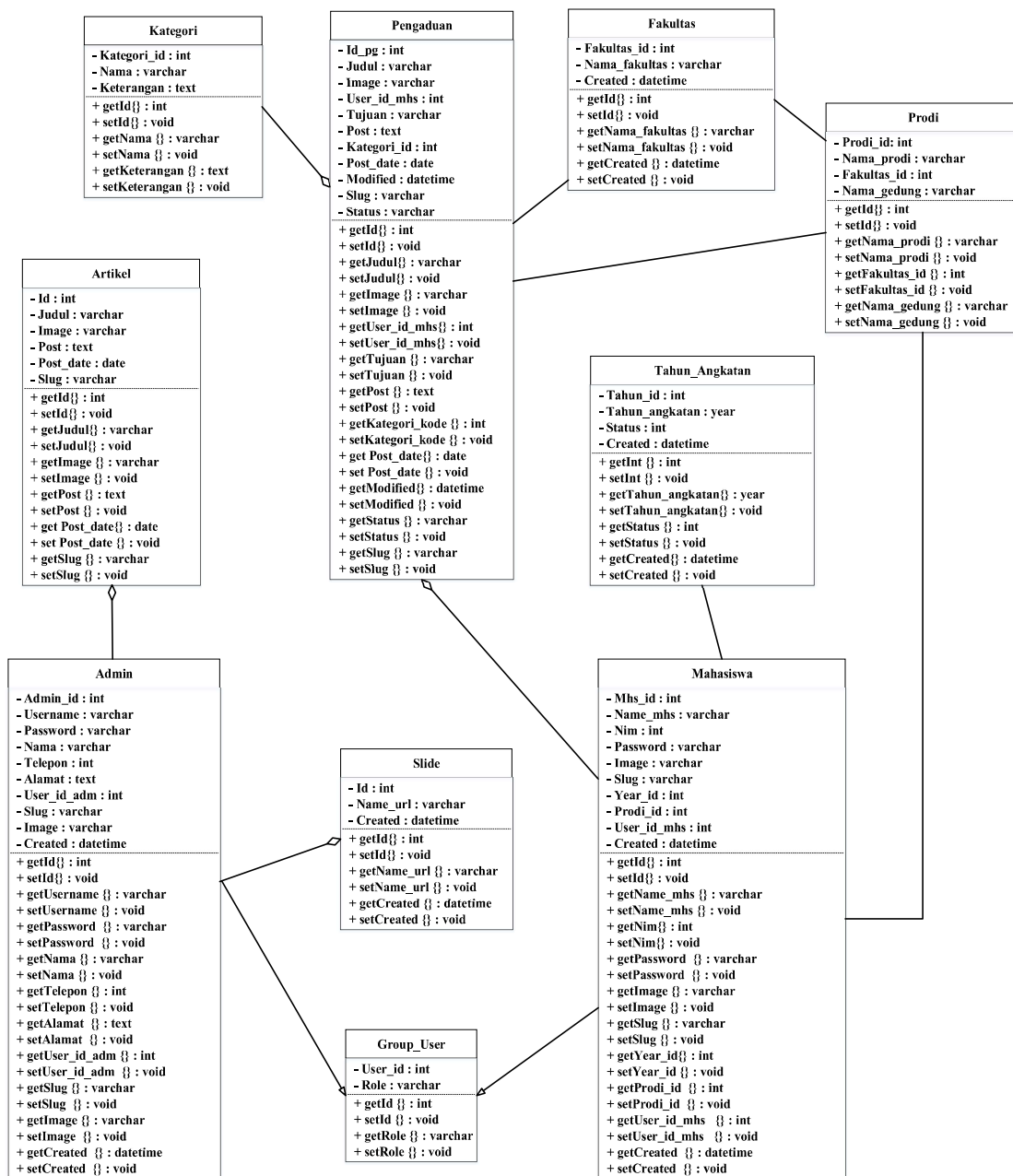
1. *Mahasiswa* melakukan *Login*.
2. Pengaduan yang dibuat mahasiswa dibatasi oleh admin melalui sistem. Pembatasannya yaitu, mahasiswa hanya boleh membuat pengaduan dengan kategori yg berbeda dihari yang sama, jika ada data pengaduan dengan kategori yang sama pada hari yang sama juga, maka mahasiswa tersebut tidak dapat memberikan pengaduan yang ingin dibuat oleh mahasiswa bersangkutan.
3. Setelah *mahasiswa* melakukan *Login*, mahasiswa lalu membuat *Pengaduan* pada *Form Pengaduan* yang telah disediakan dalam web.
4. Setelah data *Pengaduan* telah dibuat *mahasiswa*, lalu *admin* mengelola data tersebut sesuai prosedur yang telah diatur diluar sistem.

5. *Data Pengaduan* yang telah dikelola sebelumnya oleh *admin*, lalu dikirim ke *fakultas atau prodi* sesuai tujuan yang dibuat oleh *mahasiswa* pada saat membuat *Pengaduan* pada website terkait.
6. Setelah *Data Pengaduan* tersebut telah diterima oleh *fakultas atau prodi*, maka *fakultas atau prodi* langsung membuat tindakan atas laporan yang telah dibuat mahasiswa sebelumnya. Lalu tindakan tersebut dirangkum dalam sebuah data lagi dengan rapi agar mudah dibaca.
7. Setelah rangkuman data hasil tindakan tersebut dibuat oleh *fakultas atau prodi* sebelumnya, lalu *fakultas atau prodi* melakukan konfirmasi untuk setiap tindakan yang telah dilakukan oleh pihak-pihak terkait dari fakultas atau prodi.
8. Setelah melakukan konfirmasi atas tindakan *Pengaduan* tersebut, maka *fakultas atau prodi* mengirim *Hasil Tindakan* terakhir tersebut ke *admin*.
9. *Fakultas atau prodi* juga bisa langsung mengirim *Data Hasil Tindakan* tersebut ke *Admin* tanpa harus mengkonfirmasi tindakan tersebut terlebih dahulu.
10. Setelah *admin* menerima *Data Hasil Tindakan*, lalu *admin* membuat *Hasil Konfirmasi Tindakan* lagi berupa membuat perubahan status pada setiap pengaduan berdasarkan data pengaduan dari pihak *fakultas* ataupun *prodi* yang bersangkutan.
11. *Mahasiswa* dapat melihat perubahan status data pengaduan yang dibuat sebelumnya, dan jika data terakhir pengaduan tersebut belum sesuai dengan harapan *mahasiswa*, maka *mahasiswa* bisa membuat pengaduan yang sama lagi dengan memberikan penegasan pada isi pengaduannya.
12. Setelah itu *admin* dapat mengelola juga data pengaduan yang lama dengan menghapus data-data pengaduan yang lama tersebut dengan menggunakan tombol yang telah disediakan pada laman pengaduan admin. Tombol ini dapat menghapus

data-data pengaduan dengan status 'Telah Selesai atau Done' yang batas waktu pembuatan pengaduan tersebut dihitung 2 tahun sebelumnya dari tahun dimana *admin* ingin menghapus data tersebut.

### 3.7.3. Class Diagram

Gambaran Class Diagram yang dibuat untuk sistem *Penangan Keluhan Berbasis Web* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Class Diagram

Berikut ini penjelasan *Class Diagram* mengenai proses sistem *Penanganan Keluhan Berbasis Web* pada Gambar 3.5:

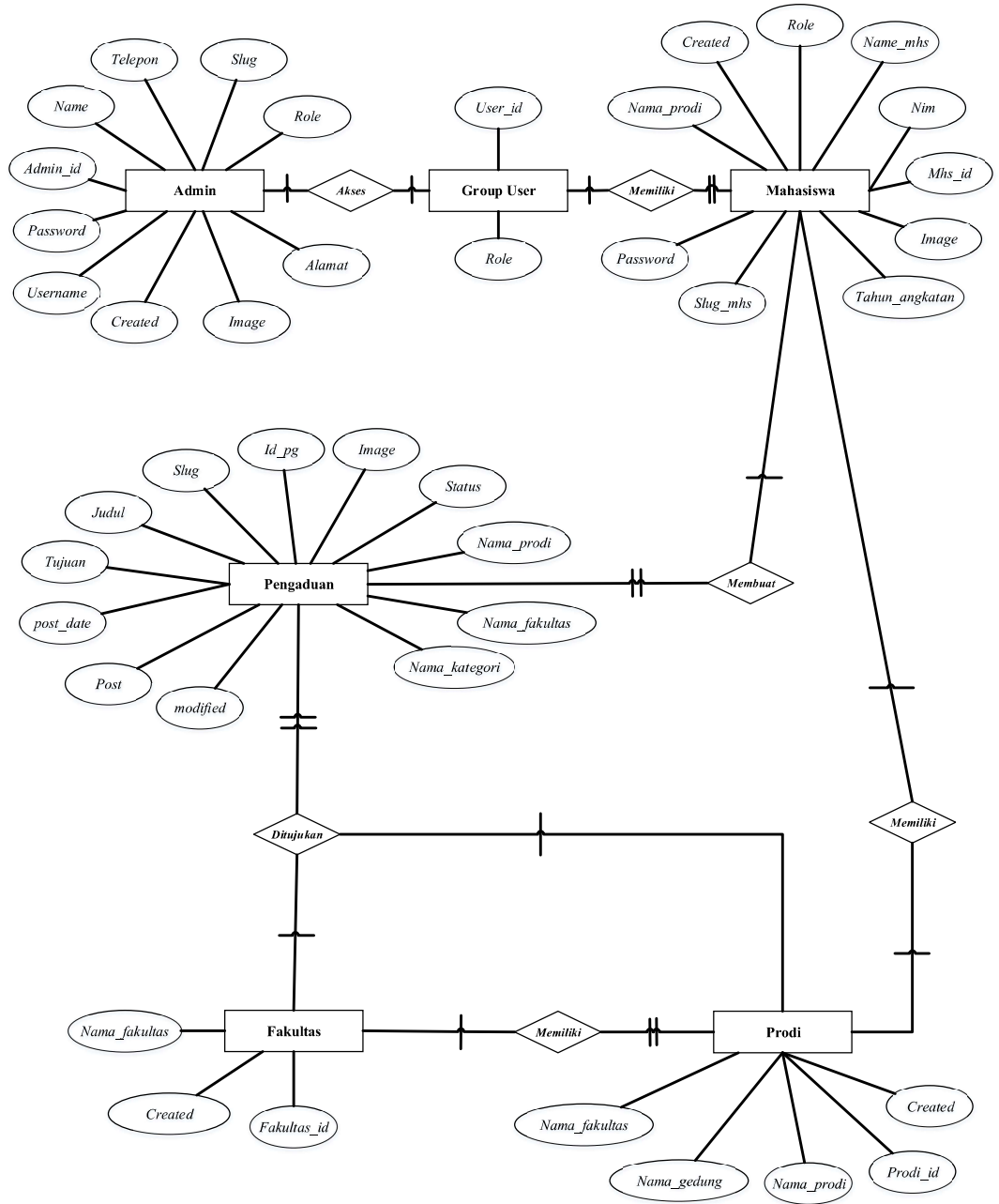
1. *Class Admin*, berfungsi untuk menyimpan data *admin* ataupun pengelola website lainnya yang telah diberi akses dan sebagai sumber data untuk memberikan akses pada website *penanganan keluhan*. Pada *Class Admin* ini, aksi yang bisa dilakukan *admin* yaitu menambahkan *admin* atau pengelola website lainnya, mengedit, ataupun menghapus data *admin*.
2. *Class Mahasiswa*, berfungsi untuk menyimpan data *mahasiswa* secara lengkap dan sebagai sumber data untuk memberikan akses pada website *penanganan keluhan*. Pada *Class Mahasiswa* ini, aksi yang bisa dilakukan *admin* yaitu menambahkan data *mahasiswa*, mengedit, ataupun menghapus data *mahasiswa*.
3. *Class Group\_User*, berfungsi memberikan hak akses kepada *admin* dan *mahasiswa* yang telah terdaftar untuk melakukan akses pada website *Penanganan Keluhan*.
4. *Class Fakultas*, berfungsi sebagai tujuan pengaduan yang dibuat oleh *mahasiswa*. Pada *Class Fakultas* ini, aksi yang bisa dilakukan oleh *admin* yaitu menambahkan data *fakultas*, mengedit, ataupun menghapus data *fakultas*.
5. *Class Prodi*, berfungsi sebagai tujuan pengaduan yang dibuat oleh *mahasiswa*. Pada *Class Prodi* ini, aksi yang bisa dilakukan oleh *admin* yaitu menambahkan data *prodi*, mengedit, ataupun menghapus data *prodi*.
6. *Class Kategori*, berfungsi sebagai tujuan pengaduan yang dibuat oleh *mahasiswa* berdasarkan kategori yang tersedia yaitu *fasilitas*, *pelayanan* dan *akademik*. Pada *Class Kategori* ini, aksi yang bisa dilakukan oleh *admin* yaitu menambahkan data *kategori*, mengedit, ataupun menghapus data *kategori*.
7. *Class Pengaduan*, berfungsi sebagai tempat penyimpanan semua *pengaduan* yang telah dibuat mahasiswa. Pada *Class Pengaduan*

ini, aksi yang bisa dilakukan oleh *admin* yaitu mengelola data *pengaduan* berdasarkan tujuannya yaitu *fakultas*, *prodi* dan *kategori*. Selain itu, aksi yang bisa dilakukan oleh *admin* adalah mengkonfirmasi data *pengaduan* yang telah selesai ditangani oleh pihak terkait dan bisa menghapus data *pengaduan*. Sedangkan aksi yang bisa dilakukan oleh *mahasiswa* dalam *Class Diagram* ini yaitu melakukan input data *pengaduan* berupa judul, tujuan berdasarkan *fakultas*, *prodi* dan *kategori*, serta pesan *pengaduan* yang diajukan.

8. *Class Tahun\_Angkatan*, berisi daftar tahun angkatan alumni mahasiswa serta tahun angkatan yang aktif. Pada *Class Tahun* ini, aksi yang bisa dilakukan oleh *admin* yaitu menambahkan tahun angkatan baru dan mengaktifkan tahun angkatan berdasarkan tahun angkatan yang aktif pada tahun tersebut.
9. *Class Artikel*, berfungsi sebagai tempat penyimpanan semua *artikel* yang telah dibuat admin.
10. *Class Slide*, berfungsi sebagai tempat penyimpanan semua gambar *slide* yang telah dimasukkan oleh admin yang akan berfungsi untuk mengubah slide tampilan halaman awal setelah login.

### 3.7.4. ER Diagram

Gambaran ER Diagram yang dibuat untuk sistem *Penangan Keluhan Berbasis Web* dapat dilihat pada Gambar 3.6.



Gambar 3.6 ER Diagram

Berikut ini penjelasan *ER Diagram* mengenai proses sistem *Penanganan Keluhan Berbasis Web* pada Gambar 3.6:

1. ER Diagram sistem *Penanganan Keluhan* memiliki 6 buah entitas yaitu:
  - a. Admin
  - b. Mahasiswa
  - c. Group\_akses
  - d. Pengaduan
  - e. Fakultas
  - f. Prodi
2. Relasi yang digunakan pada setiap entitas yaitu:
  - a. Entitas *Admin* dengan entitas *Group\_akses* menggunakan relasi *one-to-one*.
  - b. Entitas *Group\_akses* dengan entitas *Mahasiswa* menggunakan relasi *one-to-many*.
  - c. Entitas *Mahasiswa* dengan entitas *Pengaduan* menggunakan relasi *one-to-many*.
  - d. Entitas *Mahasiswa* dengan entitas *Prodi* menggunakan relasi *one-to-one*.
  - e. Entitas *Prodi* dengan entitas *Pengaduan* menggunakan relasi *one-to-many*.
  - f. Entitas *Fakultas* dengan entitas *Pengaduan* menggunakan relasi *one-to-many*.
  - g. Entitas *Fakultas* dengan entitas *Prodi* menggunakan relasi *one-to-many*.
3. Relasi *one-to-one* memiliki arti "Setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua". Salah satu contohnya dapat dilihat pada poin nomor 2 bagian 'a' yang artinya satu baris 'admin' hanya berhubungan dengan satu data 'group\_akses' yang dapat dimilikinya dan begitu juga sebaliknya.



4. Relasi *one-to-many* memiliki arti "Setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua". Salah satu contohnya dapat dilihat pada poin nomor 2 bagian 'e' yang artinya satu baris 'mahasiswa' bisa berhubungan dengan satu atau lebih data 'pengaduan yang dapat dibuatnya. Tabel-tabel tersebut antara lain:

**Tabel 3.2** Struktur Tabel Admin

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Admin_id	PK	NN	int	11
2	Name		NN	varchar	100
3	Image		NN	varchar	255
4	Username		NN	varchar	100
5	Password		NN	varchar	100
6	Telepon		NN	varchar	12
7	Alamat		NN	text	
8	Slug		NN	varchar	255
9	User_id_adm	FK	NN	int	11
10	Created		NN	Datetime	

**Tabel 3.3** Struktur Tabel Mahasiswa

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Mhs_id	PK	NN	int	11
2	Name_mhs		NN	varchar	100
3	Nim		NN	varchar	11
4	Password		NN	varchar	100
5	Image		NN	varchar	255

6	Slug_mhs		NN	varchar	255
7	Year_id	FK	NN	int	11
8	Prodi_id	FK	NN	int	11
9	User_id_mhs	FK	NN	int	11
10	Created		NN	Datetime	

**Tabel 3.4** Struktur Tabel Group\_user

No	Field Name	Key Type	Null	Data Type	Max. Length
1	User_id	PK	NN	int	11
2	Role		NN	varchar	9

**Tabel 3.5** Struktur Tabel Pengaduan

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_pg	PK	NN	int	11
2	Judul		NN	varchar	255
3	Tujuan		NN	varchar	255
4	Kategori_id	FK	NN	int	11
5	Image		NN	varchar	255
6	Post		NN	text	
7	Slug		NN	varchar	255
8	User_id_mhs	FK	NN	int	11
9	Post_date		NN	date	
10	Modified		NN	datetime	
11	Status		NN	tinyint	2

**Tabel 3.6** Struktur Tabel Fakultas

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Fakultas_id	PK	NN	int	11
2	Nama_fakultas		NN	varchar	255
3	Created		NN	datetime	

**Tabel 3.7** Struktur Tabel Prodi

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Prodi_id	PK	NN	int	11
2	Nama_prodi		NN	varchar	255
3	Fakultas_id	FK	NN	int	11
4	Nama_gedung		NN	varchar	255
5	Created		NN	datetime	

**Tabel 3.8** Struktur Tabel Tahun\_Angkatan

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Tahun_id	PK	NN	int	11
2	Tahun_angkatan		NN	year	4
3	Status		NN	Enum	'1', '2'
4	Created		NN	datetime	

**Tabel 3.9** Struktur Tabel Kategori

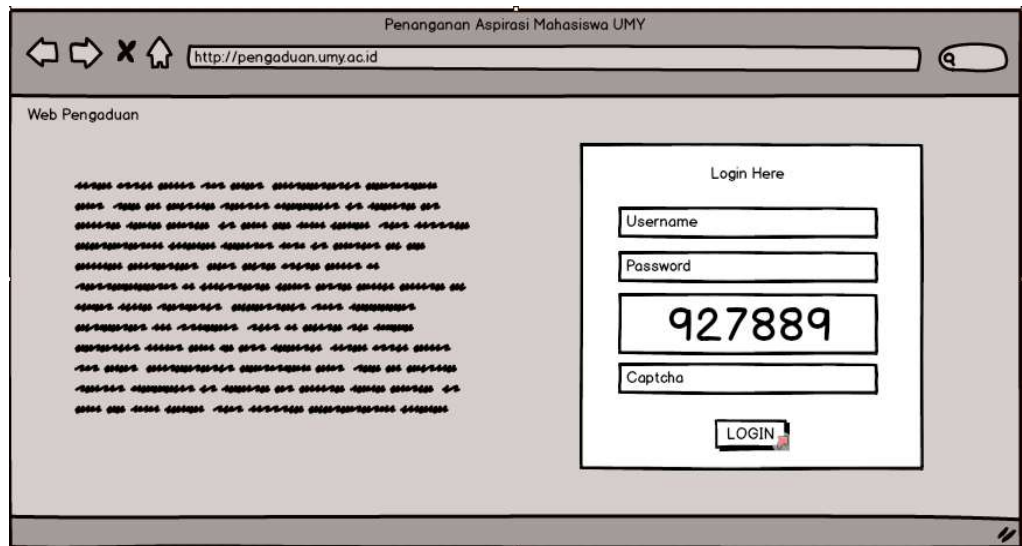
No	Field Name	Key Type	Null	Data Type	Max. Length
1	Kategori_id	PK	NN	int	11
2	Name		NN	varchar	255
3	Keterangan		NN	text	
4	Created		NN	datetime	

### 3.7.5. Rancangan Antarmuka (*Design User Interface*)

*User Interface* merupakan bagian terkemuka dalam sebuah perangkat lunak sebagai sarana komunikasi antara pengguna dengan sistem perangkat lunak itu sendiri. Hal ini karena mempermudah interaksi antara sistem dengan pengguna dalam melakukan aktivitasnya.

#### 1. Rancangan Antarmuka Halaman Login (Admin & Mahasiswa)

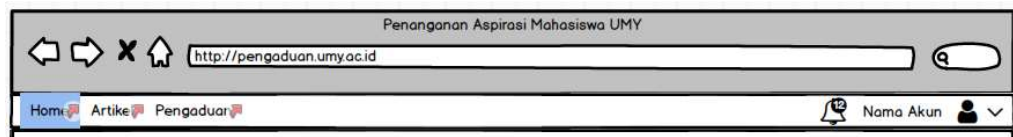
Rancangan antarmuka halaman login merupakan tampilan login yang digunakan oleh mahasiswa maupun admin dengan tampilan yang sama. Gambar rancangan antarmuka halaman login dapat dilihat pada Gambar 3.7.



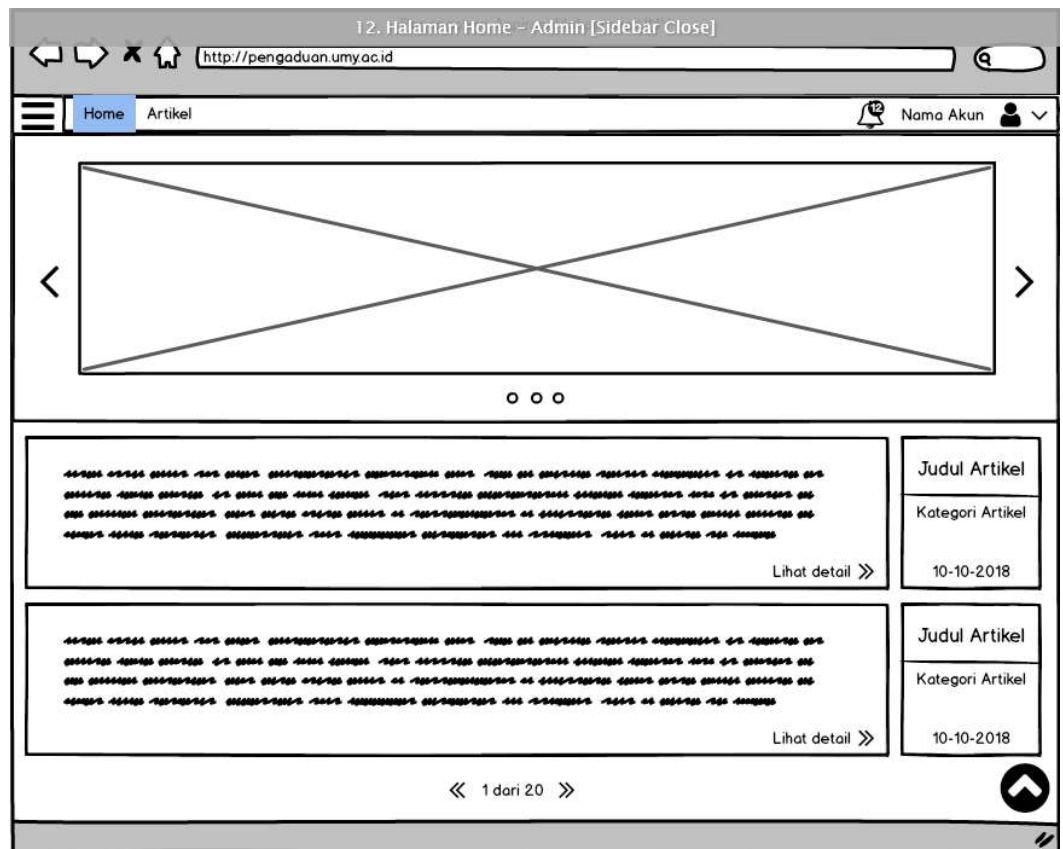
**Gambar 3.7** Halaman Login (Admin & Mahasiswa)

## 2. Rancangan Antarmuka Halaman Home (Admin & Mahasiswa)

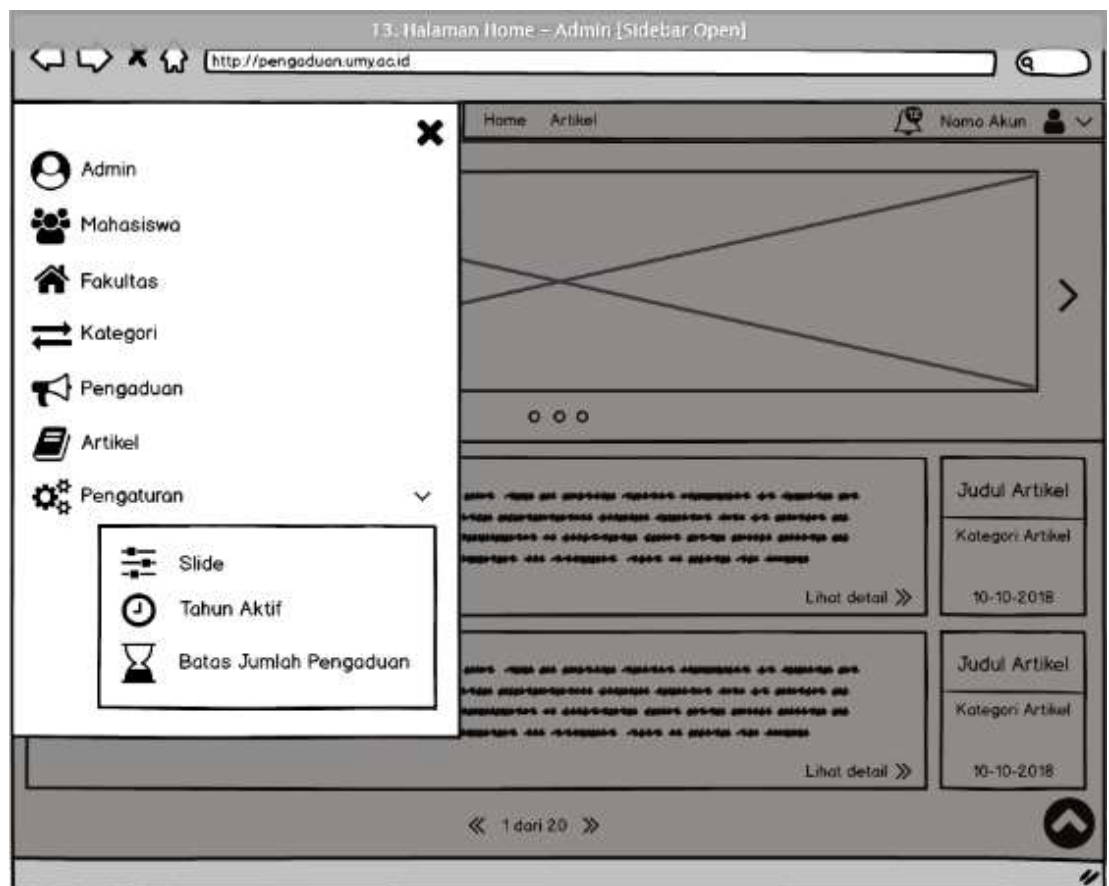
Rancangan antarmuka halaman home merupakan tampilan pertama yang dilihat setelah melakukan login. Tampilan antarmuka home pada admin dan mahasiswa memiliki kesamaan desain, namun ada sedikit perbedaan yaitu pada menu bar. Untuk perbedaan menu bar antara kedua user ini, dapat dilihat pada Gambar 3.8, Gambar 3.9 dan Gambar 3.10.



**Gambar 3.8** Menu Bar (Mahasiswa)



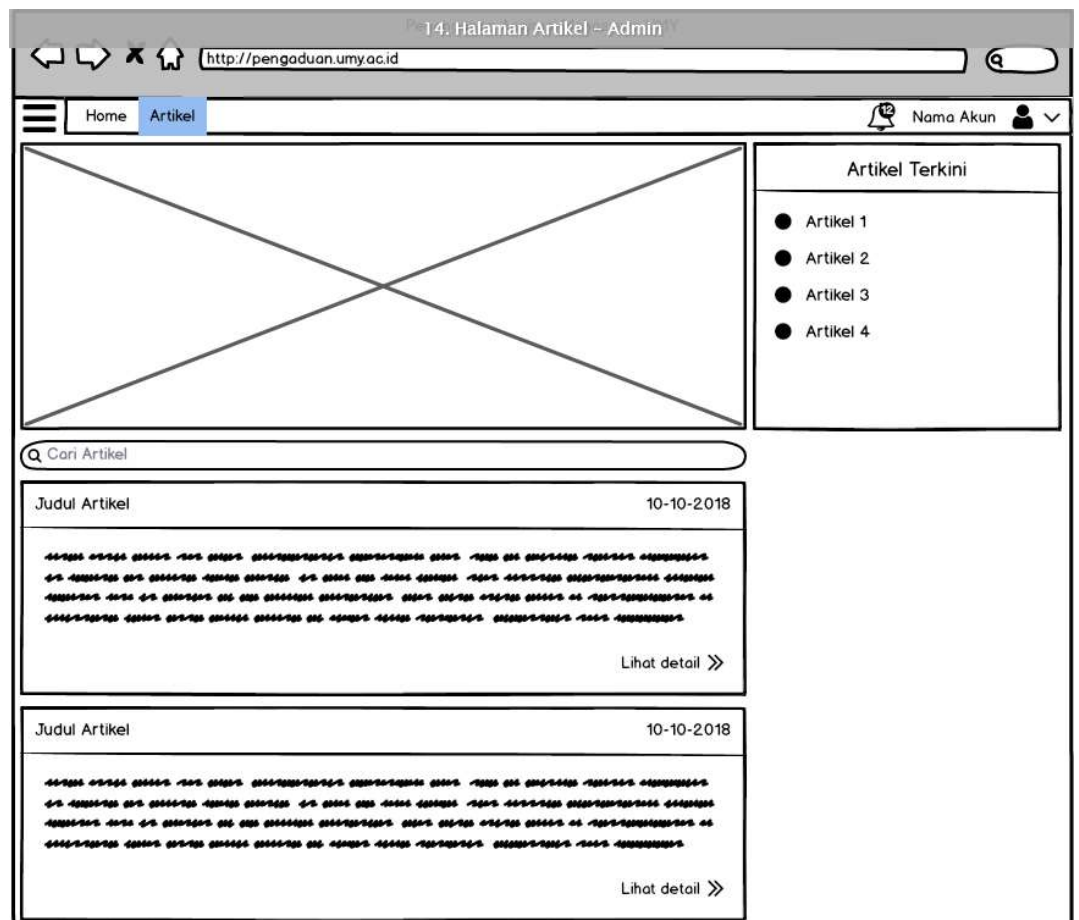
Gambar 3.9 Halaman Home (Admin – Open & Close Sidebar Menu)



**Gambar 3.10** Halaman Home (Admin – Open & Close Sidebar Menu)

### 3. Rancangan Antarmuka Halaman Artikel (Admin & Mahasiswa)

Rancangan antarmuka halaman artikel merupakan tampilan halaman yang memuat sejumlah artikel yang dapat dilihat oleh admin ataupun mahasiswa. Tampilan antarmuka halaman artikel untuk admin dan mahasiswa memiliki kesamaan desain, namun ada sedikit perbedaan pada menu bar untuk admin dan mahasiswa. Untuk tampilan menu bar pada mahasiswa dapat dilihat pada gambar sebelumnya yaitu Gambar 3.8, sedangkan tampilan rancangan antarmuka halaman artikel keduanya dapat dilihat pada Gambar 3.11.



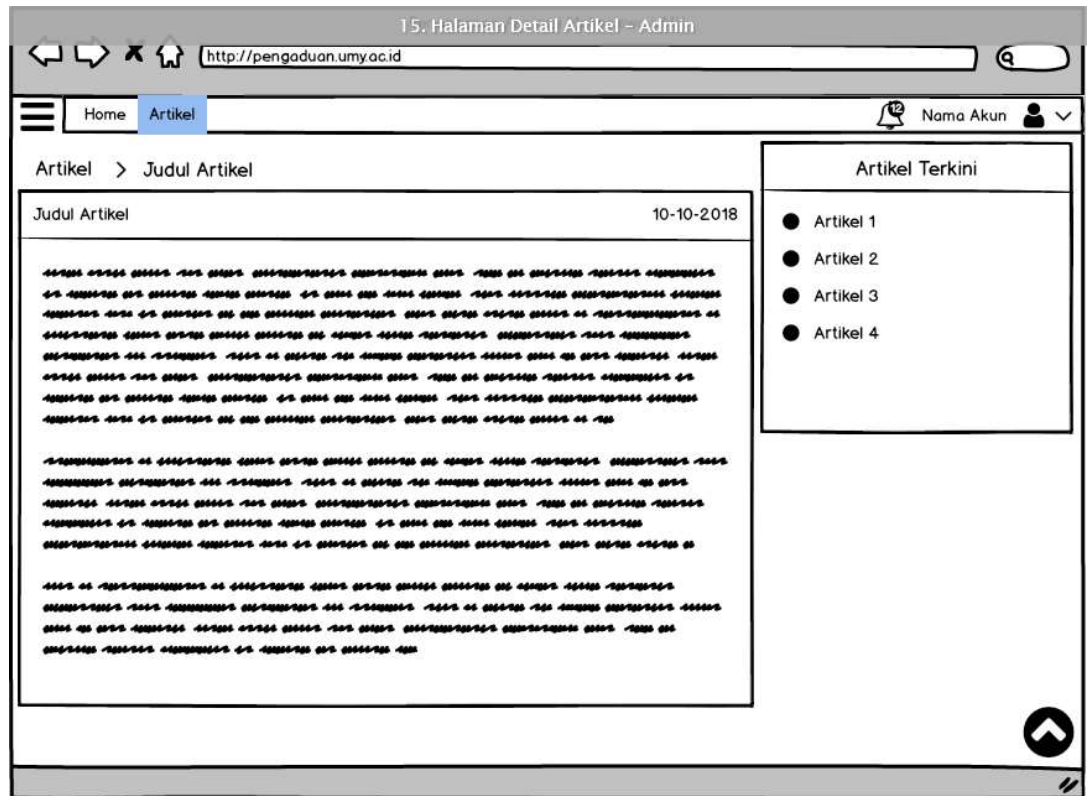
Gambar 3.11 Halaman Artikel (Admin & Mahasiswa)

#### 4. Rancangan Antarmuka Halaman Detail Artikel (Admin & Mahasiswa)

Rancangan antarmuka halaman detail artikel merupakan tampilan halaman yang memuat detail dari artikel yang terdapat pada halaman utama artikel yang dapat dilihat oleh admin dan mahasiswa. Tampilan antarmuka halaman detail artikel untuk admin dan mahasiswa memiliki kesamaan desain, namun ada sedikit perbedaan pada menu bar untuk admin dan mahasiswa. Untuk tampilan menu bar pada mahasiswa dapat dilihat pada gambar sebelumnya yaitu Gambar 3.8, sedangkan tampilan



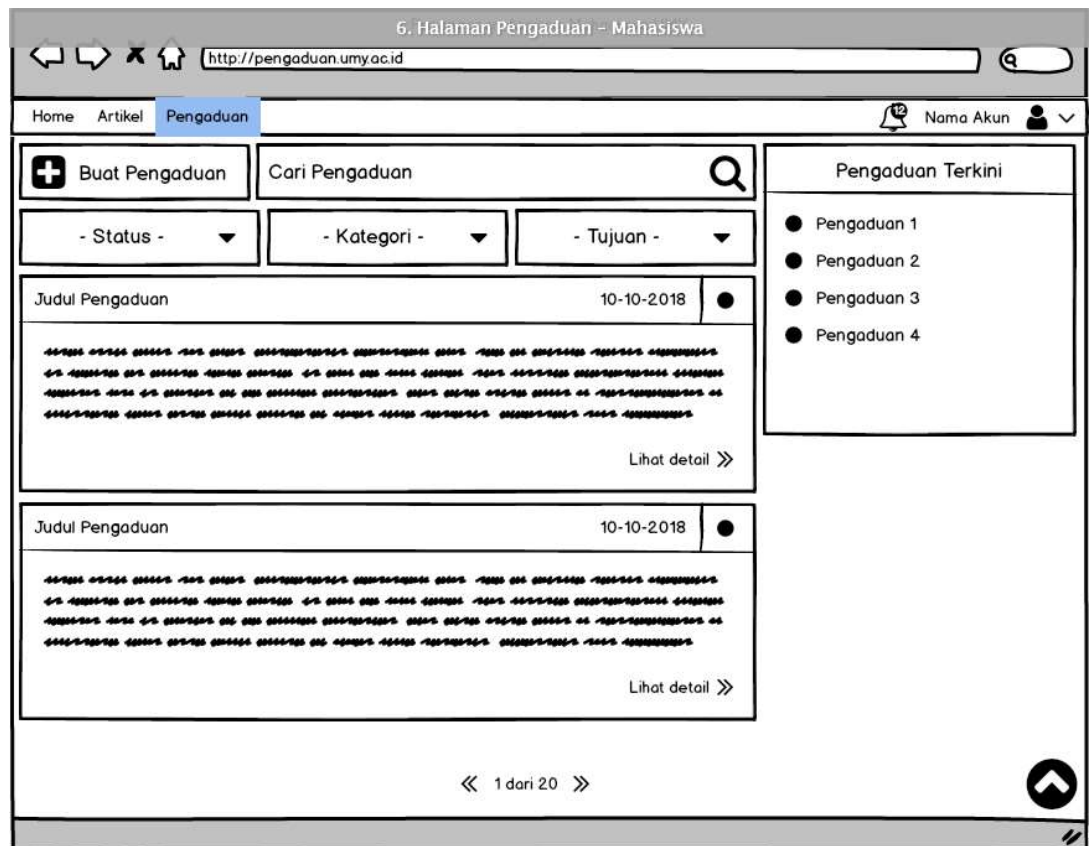
rancangan antarmuka halaman detail artikel keduanya dapat dilihat pada Gambar 3.12.



Gambar 3.12 Halaman Detail Artikel (Admin & Mahasiswa)

##### 5. Rancangan Antarmuka Halaman Pengaduan (Mahasiswa)

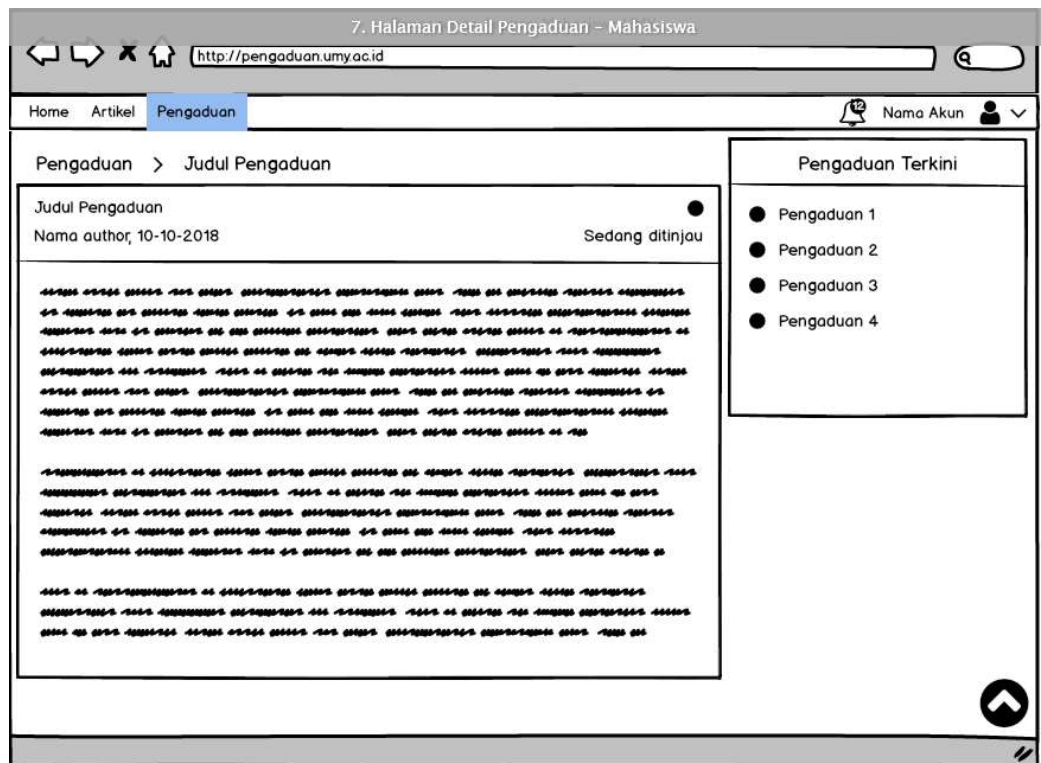
Rancangan antarmuka halaman pengaduan merupakan tampilan halaman yang memuat semua pengaduan yang sebelumnya telah dibuat oleh mahasiswa. Tampilan ini menyediakan tombol navigasi yang memuat halaman yang berisi form pengisian pengaduan yang akan dilakukan oleh mahasiswa. Gambaran rancangan antarmuka halaman pengaduan dapat dilihat pada Gambar 3.13.



Gambar 3.13 Halaman Pengaduan (Mahasiswa)

## 6. Rancangan Antarmuka Halaman Detail Pengaduan (Mahasiswa)

Rancangan antarmuka halaman detail pengaduan merupakan tampilan halaman yang memuat lebih lengkap info mengenai pengaduan yang sebelumnya telah dibuat oleh mahasiswa. Gambaran rancangan antarmuka halaman detail pengaduan dapat dilihat pada Gambar 3.14.



Gambar 3.14 Halaman Detail Pengaduan (Mahasiswa)

## 7. Rancangan Antarmuka Halaman Form Pengaduan (Mahasiswa)

Rancangan antarmuka halaman form pengaduan merupakan tampilan halaman yang memuat form pengaduan yang akan diisi oleh mahasiswa sebagai aspirasi mahasiswa. Selain rancangan antarmuka halaman form pengaduan, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol kirim setelah selesai mengisi semua field pada form pengaduan ini. Gambaran rancangan antarmuka halaman form pengaduan dan modal tombol *kirim* dapat dilihat pada Gambar 3.15 dan 3.16.

8. Halaman Buat Pengaduan - Mahasiswa

http://pengaduan.umy.ac.id

Home Artikel Pengaduan Nama Akun

Pengaduan > Buat Pengaduan

Nama Author \*  
Nim \*  
Judul \*  
Tujuan Pengaduan \*  Fakultas  Prodi  
Kategori Pengaduan \*  
- Pilih Kategori -  
Pelayanan  
Akademik  
Fasilitas

Nama File Pilih File

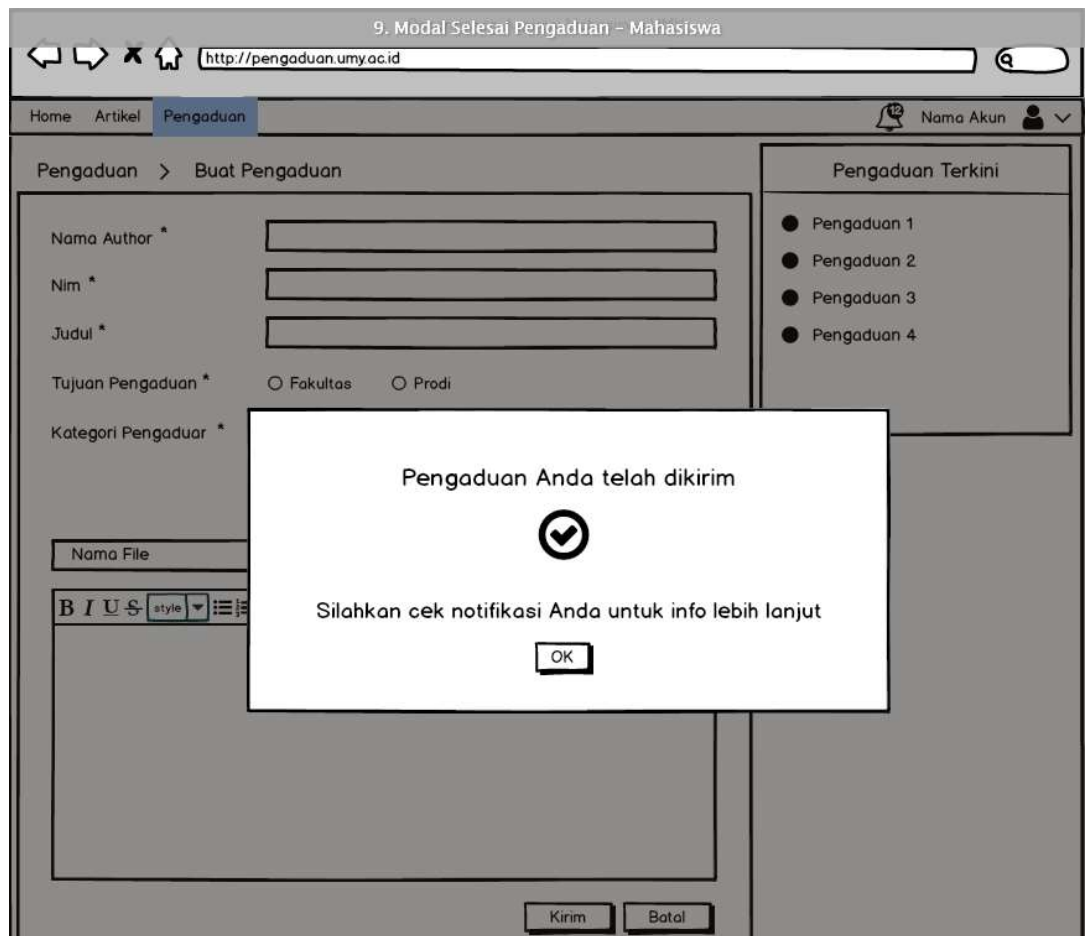
B I U S style [icons]

Kirim Batal

Pengaduan Terkini

- Pengaduan 1
- Pengaduan 2
- Pengaduan 3
- Pengaduan 4

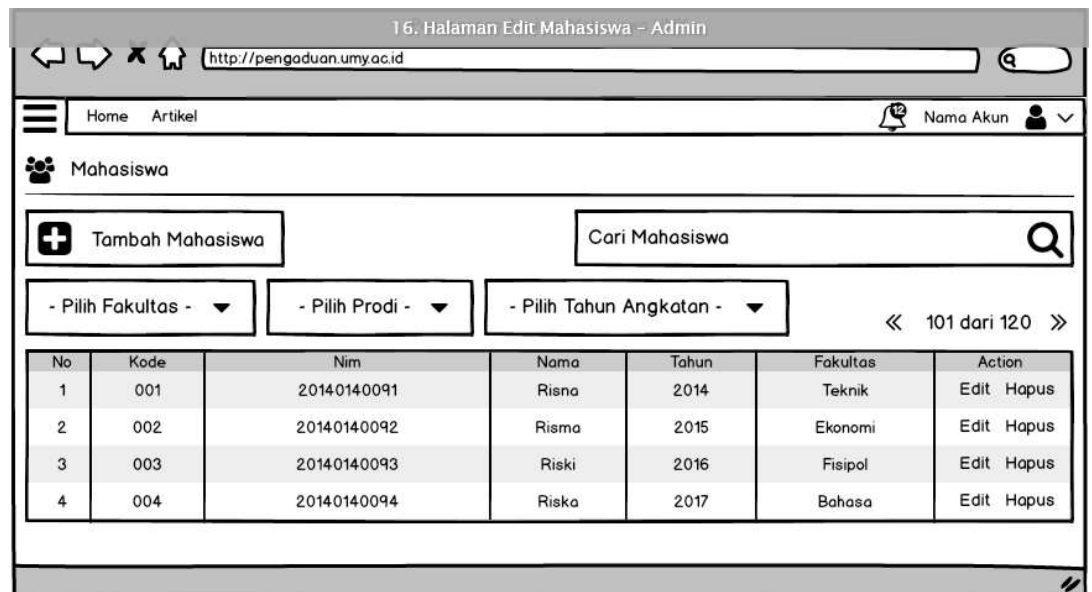
Gambar 3.15 Halaman Form Pengaduan (Mahasiswa)



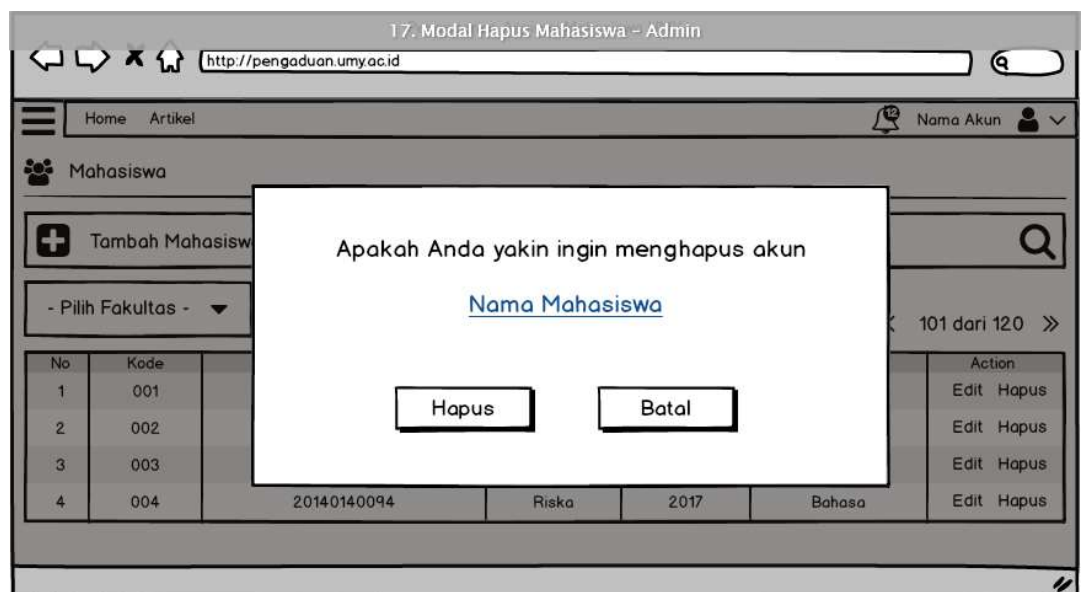
**Gambar 3.16** Modal Selesai Pengaduan (Mahasiswa)

#### 8. Rancangan Antarmuka Halaman Data Mahasiswa (Admin)

Rancangan antarmuka halaman data mahasiswa merupakan tampilan halaman yang memuat semua data mahasiswa dan terdapat tombol navigasi untuk menambahkan data mahasiswa baru. Selain rancangan antarmuka halaman data mahasiswa, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka halaman data mahasiswa dan modal tombol *hapus* dapat dilihat pada Gambar 3.17 dan 3.18.



Gambar 3.17 Halaman Data Mahasiswa (Admin)



Gambar 3.18 Modal Hapus Data Mahasiswa (Admin)

## 9. Rancangan Antarmuka Halaman Tambah dan Edit Data Mahasiswa (Admin)

Rancangan antarmuka halaman tambah dan edit data mahasiswa merupakan tampilan halaman yang memuat semua data mahasiswa yang akan ditambahkan ataupun diupdate datanya.

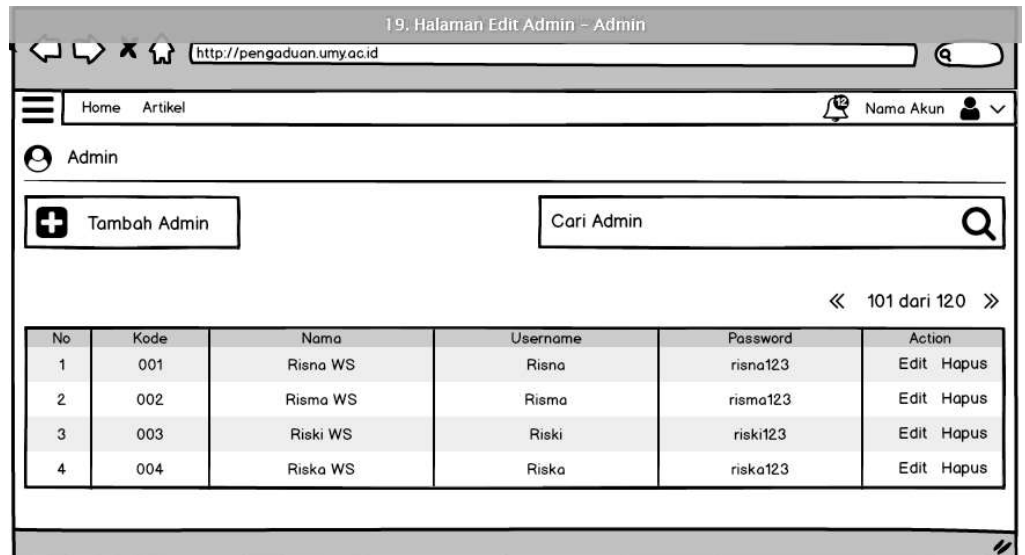
Kedua rancangan antarmuka ini memiliki desain yang sama. Gambaran rancangan antarmuka halaman tambah dan edit data mahasiswa dapat dilihat pada Gambar 3.19.

**Gambar 3.19** Halaman Tambah dan Edit Data Mahasiswa (Admin)

### 10. Rancangan Antarmuka Halaman Data Admin (Admin)

Rancangan antarmuka halaman data admin merupakan tampilan halaman yang memuat semua data admin dan terdapat tombol navigasi untuk menambahkan data admin baru. Selain rancangan antarmuka halaman data admin, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka modal tombol *hapus* ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18.

Sedangkan untuk gambaran rancangan antarmuka halaman data admin dapat dilihat pada Gambar 3.20.



**Gambar 3.20** Halaman Data Admin (Admin)

## 11. Rancangan Antarmuka Halaman Tambah dan Edit Data Admin (Admin)

Rancangan antarmuka halaman tambah dan edit data admin merupakan tampilan halaman yang memuat semua data admin yang akan ditambahkan ataupun diupdate datanya. Kedua rancangan antarmuka ini memiliki desain yang sama. Gambaran rancangan antarmuka halaman tambah dan edit data admin dapat dilihat pada Gambar 3.21.



**Gambar 3.21** Halaman Tambah dan Edit Data Admin  
(Admin)

## 12. Rancangan Antarmuka Halaman Data Fakultas (Admin)

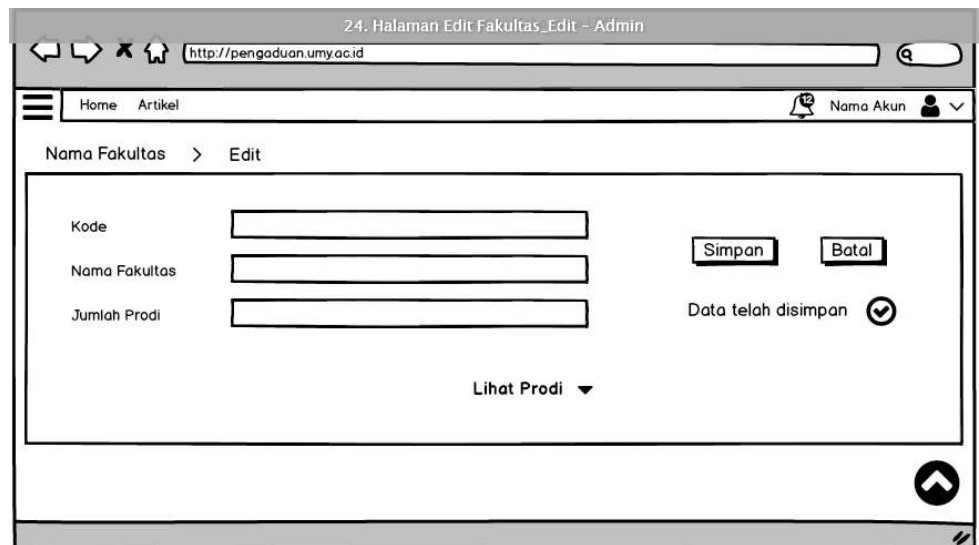
Rancangan antarmuka halaman data fakultas merupakan tampilan halaman yang memuat semua data fakultas dan terdapat tombol navigasi untuk menambahkan data fakultas baru. Selain rancangan antarmuka halaman data fakultas, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka modal tombol *hapus* ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18. Sedangkan untuk gambaran rancangan antarmuka halaman data fakultas dapat dilihat pada Gambar 3.22.

No	Kode	Nama Fakultas	Jumlah Prodi	Action
1	001	Teknik	4	Edit Hapus
2	002	Ekonomi	2	Edit Hapus
3	003	Fisipol	8	Edit Hapus
4	004	Bahasa	6	Edit Hapus

**Gambar 3.22** Halaman Data Fakultas (Admin)

### 13. Rancangan Antarmuka Halaman Tambah dan Edit Data Fakultas (Admin)

Rancangan antarmuka halaman tambah dan edit data fakultas merupakan tampilan halaman yang memuat semua data fakultas yang akan ditambahkan ataupun diupdate datanya. Kedua rancangan antarmuka ini memiliki desain yang sama. Gambaran rancangan antarmuka halaman tambah dan edit data fakultas dapat dilihat pada Gambar 3.23.

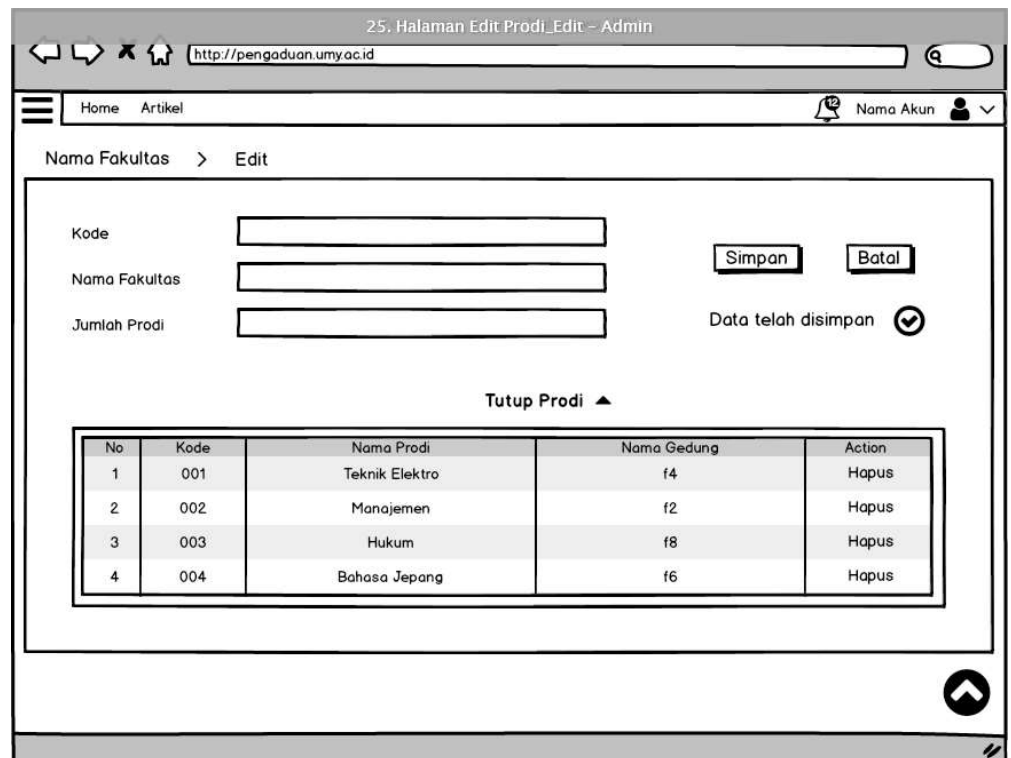


**Gambar 3.23** Halaman Tambah dan Edit Data Fakultas  
(Admin)

#### **14. Rancangan Antarmuka Tampilan Maximize Data Prodi**

##### **(Admin)**

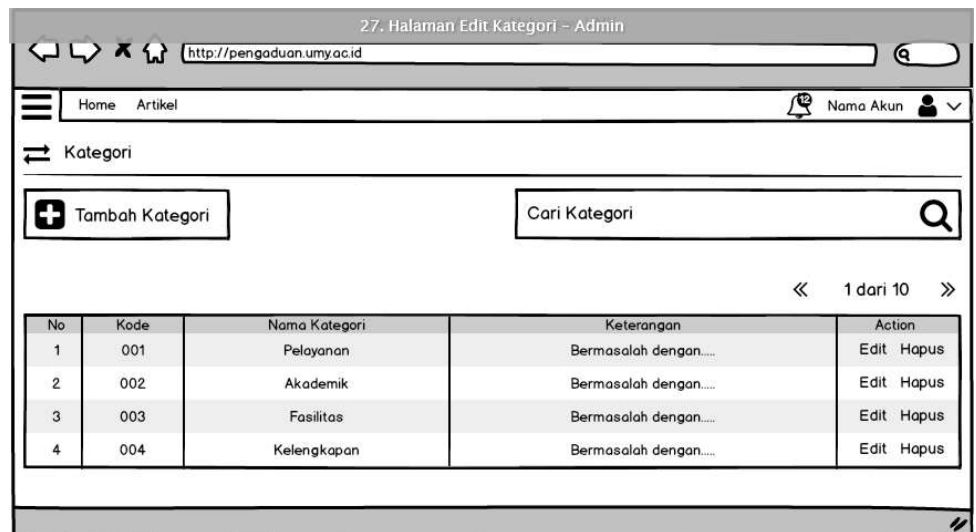
Rancangan antarmuka tampilan maximize data prodi dapat dilihat jika pengguna mengklik tombol *lihat prodi* yang terdapat pada halaman *Tambah dan Edit Data Fakultas* pada Gambar 3.23 diatas. Rancangan antarmuka tampilan ini memuat semua data prodi sesuai dengan data fakultas yang dipilih (*pada mode edit*). Selain rancangan antarmuka tampilan maximize data prodi, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada tampilan ini. Gambaran rancangan antarmuka modal tombol hapus ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18. Sedangkan gambaran rancangan antarmuka tampilan maximize data prodi dapat dilihat pada Gambar 3.24.



**Gambar 3.24** Tampilan Maximize Data Prodi (Admin)

### 15. Rancangan Antarmuka Halaman Data Kategori (Admin)

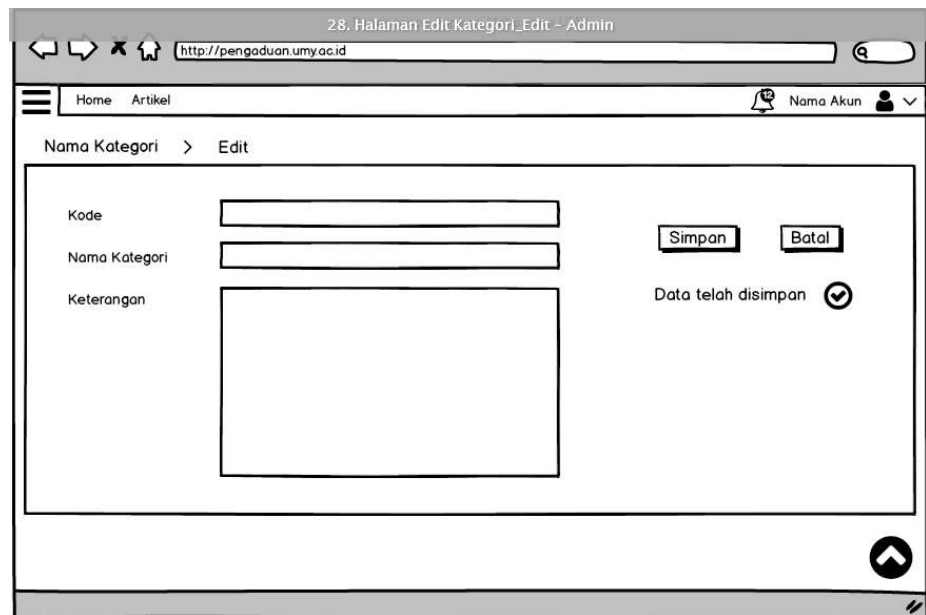
Rancangan antarmuka halaman data kategori merupakan tampilan halaman yang memuat semua data kategori dan terdapat tombol navigasi untuk menambahkan data kategori baru. Selain rancangan antarmuka halaman data admin, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka modal tombol hapus ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18. Sedangkan untuk gambaran rancangan antarmuka halaman data kategori dapat dilihat pada Gambar 3.25.



**Gambar 3.25** Halaman Data Kategori (Admin)

## **16. Rancangan Antarmuka Halaman Tambah dan Edit Data Kategori (Admin)**

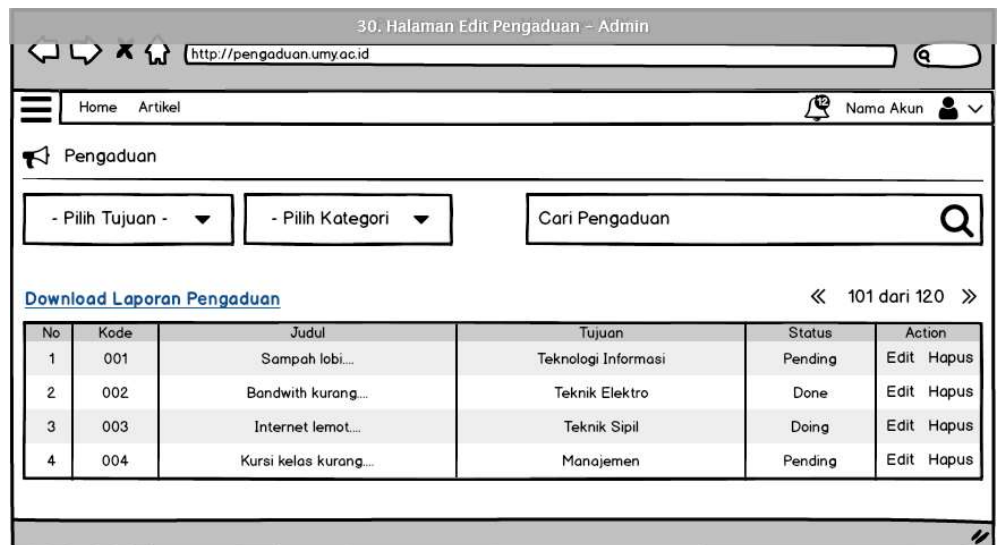
Rancangan antarmuka halaman tambah dan edit data kategori merupakan tampilan halaman yang memuat semua data kategori yang akan ditambahkan ataupun diupdate datanya. Kedua rancangan antarmuka ini memiliki desain yang sama. Gambaran rancangan antarmuka halaman tambah dan edit data kategori dapat dilihat pada Gambar 3.26.



**Gambar 3.26** Halaman Tambah dan Edit Data Kategori (Admin)

### 17. Rancangan Antarmuka Halaman Data Pengaduan (Admin)

Rancangan antarmuka halaman data pengaduan merupakan tampilan halaman yang memuat semua data pengaduan yang sebelumnya telah dikirim oleh mahasiswa. Selain rancangan antarmuka halaman data pengaduan, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka modal tombol hapus ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18. Sedangkan untuk gambaran rancangan antarmuka halaman data pengaduan dapat dilihat pada Gambar 3.27.



**Gambar 3.27** Halaman Data Pengaduan (Admin)

### 18. Rancangan Antarmuka Halaman Edit Data Pengaduan (Admin)

Rancangan antarmuka halaman edit data pengaduan merupakan tampilan halaman yang memuat semua data pengaduan yang akan diupdate status datanya oleh admin (*tidak bisa mengedit data lainnya*). Gambaran rancangan antarmuka halaman edit data pengaduan dapat dilihat pada Gambar 3.28.

33. Halaman Edit Pengaduan\_Edit - Admin

http://pengaduan.umy.ac.id

Home Artikel Nama Akun

Judul Pengaduan > Edit

Judul Pengaduan

Nama Author - Nim Author, 03:15 - 10/10/2018

Pesan

- Ubah Status -

001  
Kode Pengaduan

- Pilih Kategori -

- Pilih Tujuan -

Simpan Batal

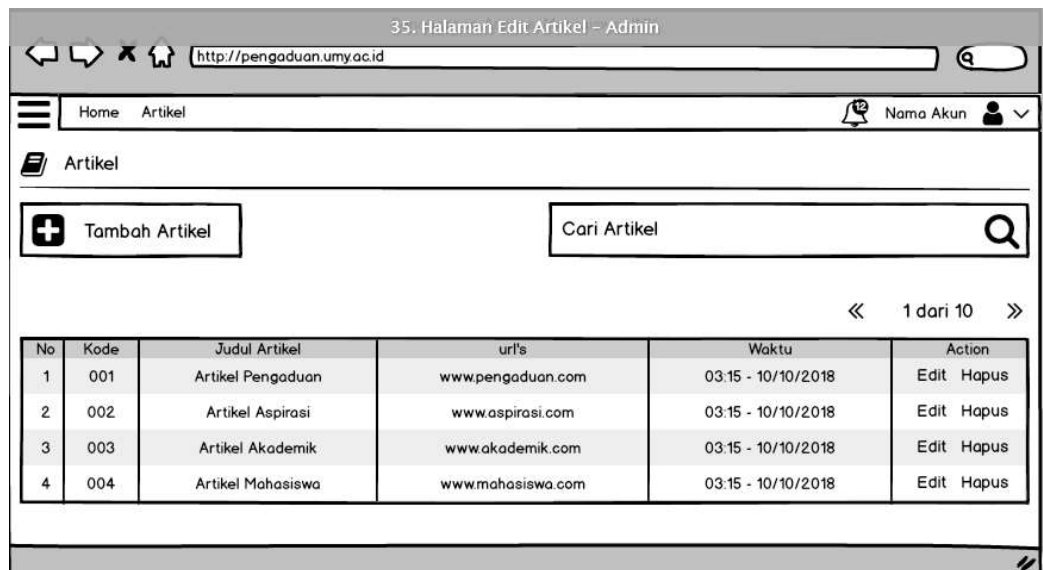
Data telah disimpan ✓

**Gambar 3.28** Halaman Edit Data Pengaduan (Admin)

### 19. Rancangan Antarmuka Halaman Data Artikel (Admin)

Rancangan antarmuka halaman data artikel merupakan tampilan halaman yang memuat semua data artikel dan terdapat tombol navigasi untuk menambahkan data artikel baru. Selain rancangan antarmuka halaman data artikel, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka modal tombol hapus ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18. Sedangkan untuk gambaran rancangan antarmuka halaman data artikel dapat dilihat pada Gambar 3.29.

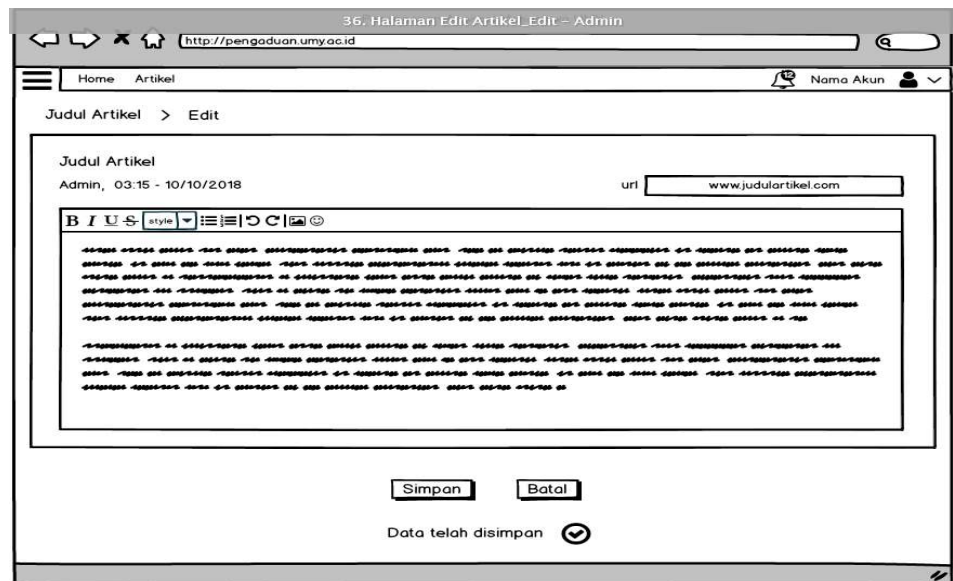




**Gambar 3.29** Halaman Data Artikel (Admin)

## **20. Rancangan Antarmuka Halaman Tambah dan Edit Data Artikel (Admin)**

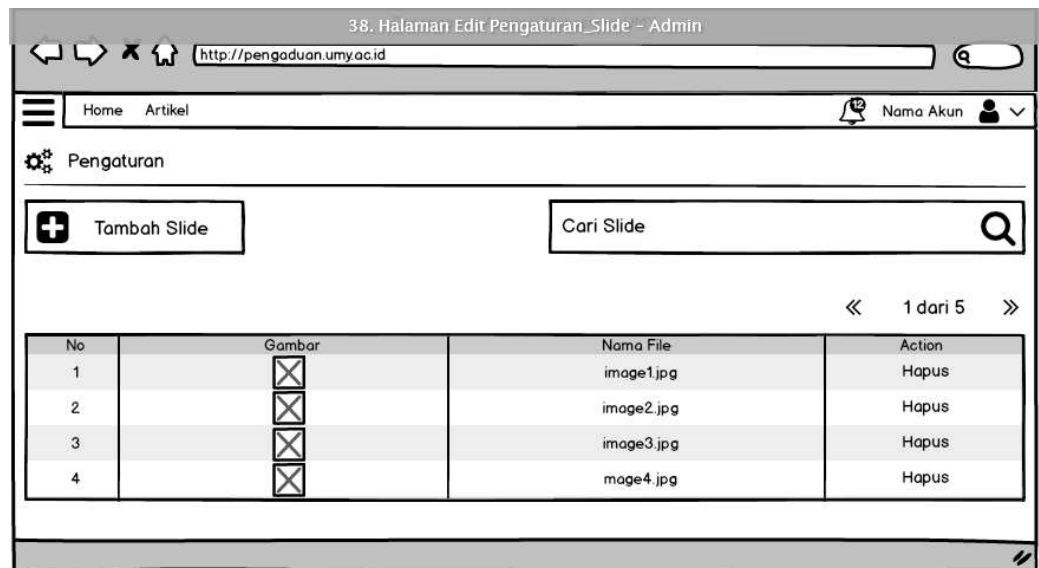
Rancangan antarmuka halaman tambah dan edit data artikel merupakan tampilan halaman yang memuat semua data artikel yang akan ditambahkan ataupun diupdate datanya. Kedua rancangan antarmuka ini memiliki desain yang sama. Gambaran rancangan antarmuka halaman tambah dan edit data artikel dapat dilihat pada Gambar 3.30.



**Gambar 3.30** Halaman Tambah dan Edit Data Pengaduan (Admin)

## 21. Rancangan Antarmuka Halaman Pengaturan - Edit Slide (Admin)

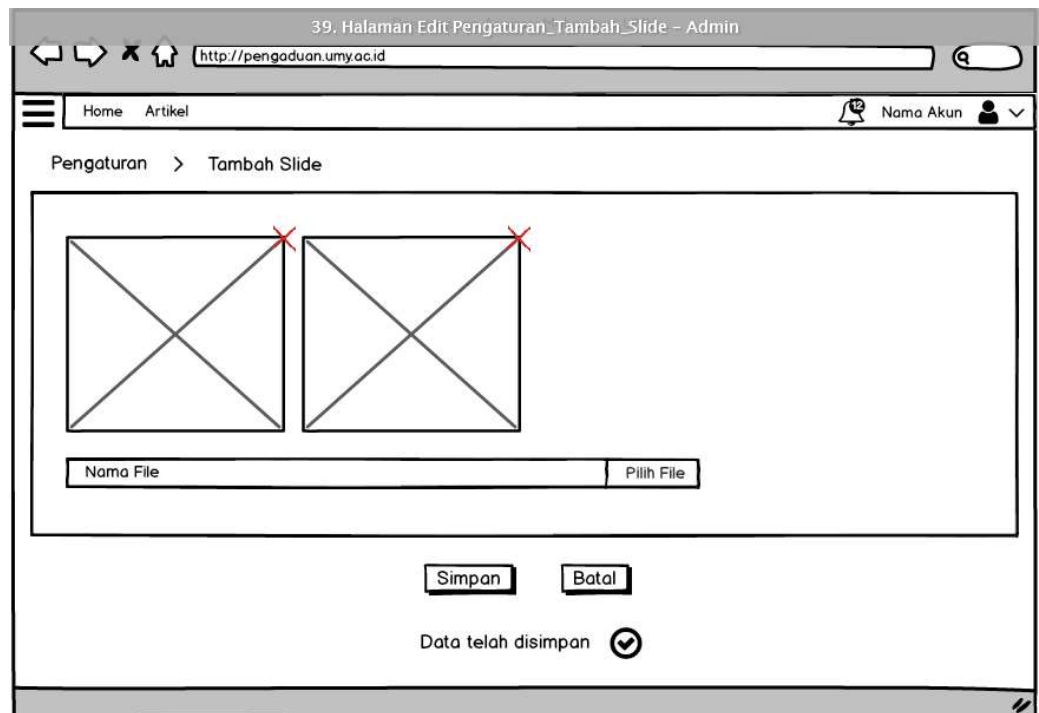
Rancangan antarmuka halaman pengaturan - edit slide merupakan tampilan halaman yang memuat semua data slide yang akan ditampilkan pada halaman utama dan halaman home. Selain rancangan antarmuka halaman pengaturan - edit slide, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka modal tombol hapus ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18. Sedangkan gambaran rancangan antarmuka halaman pengaturan - edit slide dapat dilihat pada Gambar 3.31.



**Gambar 3.31** Halaman Pengaturan – Edit Slide (Admin)

## 22. Rancangan Antarmuka Halaman Pengaturan – Tambah Slide (Admin)

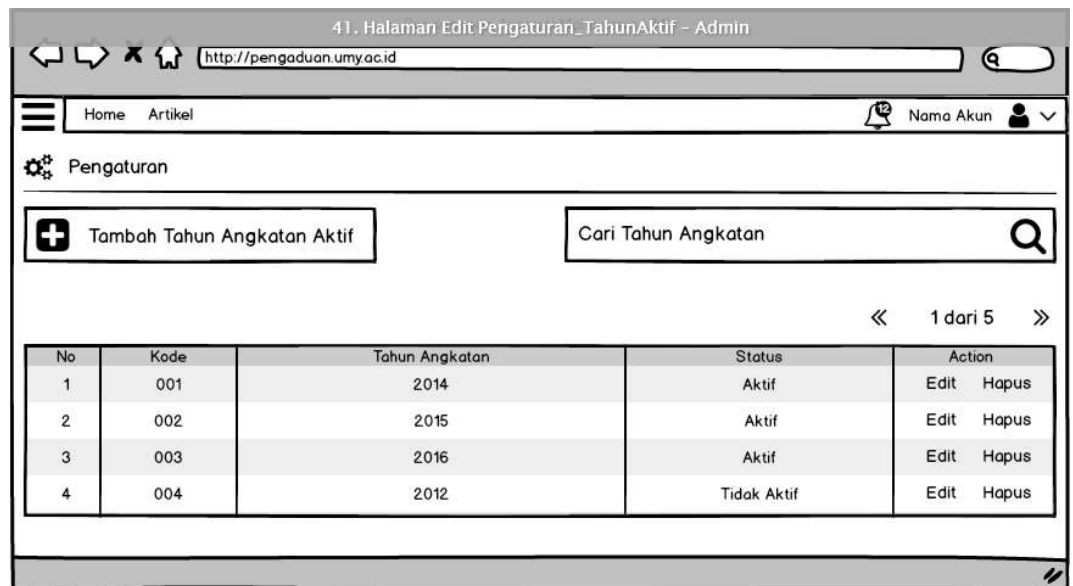
Rancangan antarmuka halaman pengaturan – tambah slide merupakan tampilan yang memungkinkan admin untuk menambahkan slide yang ingin ditampilkan pada halaman utama. Gambaran rancangan antarmuka halaman pengaturan – tambah slide dapat dilihat pada Gambar 3.32.



**Gambar 3.32** Halaman Pengaturan – Tambah Slide  
(Admin)

### 23. Rancangan Antarmuka Halaman Pengaturan – Edit Tahun Aktif (Admin)

Rancangan antarmuka halaman pengaturan – edit tahun aktif merupakan tampilan halaman yang memuat semua data tahun aktif mahasiswa. Selain rancangan antarmuka halaman pengaturan - edit tahun aktif, juga ada rancangan antarmuka modal untuk halaman ini yang akan muncul jika mengklik tombol *hapus* jika ingin menghapus salah satu data pada halaman ini. Gambaran rancangan antarmuka modal tombol hapus ini sama dengan rancangan sebelumnya yang terdapat pada Gambar 3.18. Sedangkan gambaran rancangan antarmuka halaman pengaturan – edit tahun aktif dapat dilihat pada Gambar 3.33.

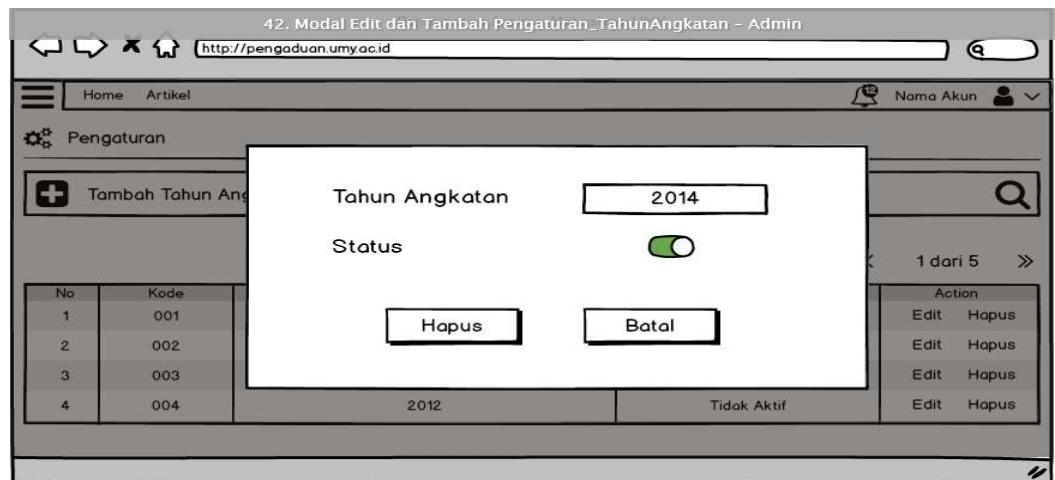


**Gambar 3.33** Halaman Pengaturan – Edit Tahun Aktif  
(Admin)

## 24. Rancangan Antarmuka Tampilan Modal Pengaturan

### Tambah dan Edit Tahun Aktif (Admin)

Rancangan antarmuka tampilan modal tambah dan edit pengaturan – tahun aktif merupakan tampilan modal yang memuat data tahun untuk diubah status datanya sesuai data yang dipilih untuk diedit (*mode edit*) statusnya. Tampilan modal untuk mode tambah atau edit memiliki rancangan desain yang sama. Gambaran rancangan antarmuka tampilan modal tambah dan edit pengaturan – tahun aktif dapat dilihat pada Gambar 3.34.

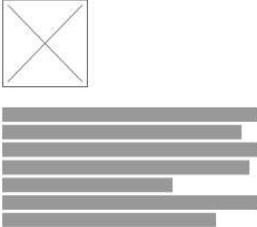
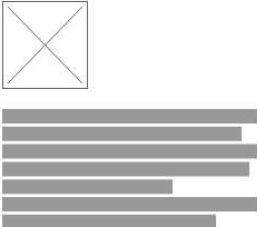


**Gambar 3.34** Modal Pengaturan – Tambah dan Edit Tahun Aktif (Admin)

## 25. Rancangan Antarmuka Tampilan Pdf Laporan Pengaduan (Admin)

Rancangan antarmuka tampilan pdf laporan pengaduan merupakan tampilan halaman yang memuat semua data pengaduan yang dengan dengan semua status lengkap. Gambaran rancangan antarmuka tampilan pdf laporan pengaduan dapat dilihat pada Gambar 3.35.

32. Bentuk PDF Laporan Pengaduan

No	Pengaduan	Tujuan dan Isi	Kategori	Status Tindakan
1.	Judul Pengaduan Kode : 001 Waktu : 03:15 - 10/10/2018	Fakultas Teknik 	Pelayanan	Done
2.	Judul Pengaduan Kode : 002 Waktu : 03:15 - 10/10/2018	Fakultas Ekonomi 	Akademik	Doing

**Gambar 3.35** Tampilan Pdf Laporan Pengaduan (Admin)