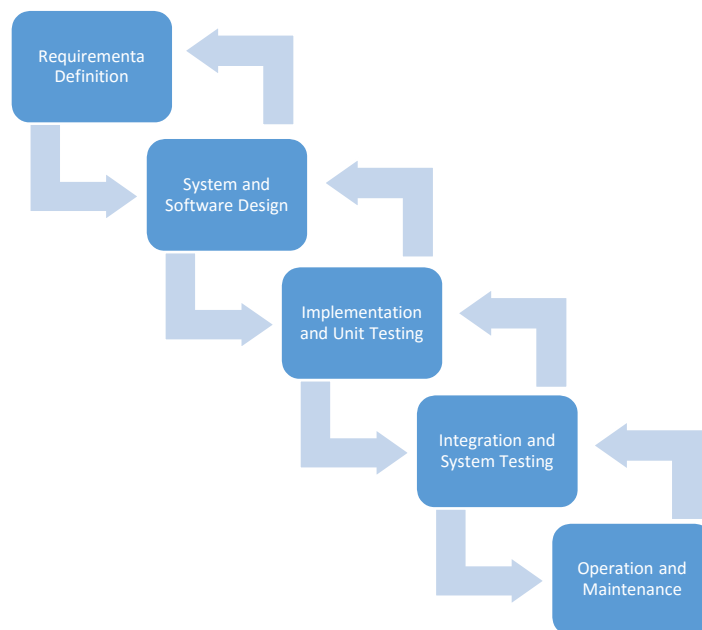


BAB III CARA DAN METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Penelitian tentang sistem penanganan keluhan berbasis *web* studi kasus Fakultas Teknik Universitas Muhammadiyah Yogyakarta menggunakan metode *Software Development Life Cycle (SDLC)* model *waterfall*. Model *waterfall* digambarkan seperti pada Gambar 3.1.



Gambar 3.1. *SDLC* Alur Penelitian

- a. Tahap perencanaan menyangkut studi tentang kebutuhan pengguna, kelayakan baik secara teknik maupun secara teknologi. Tahap (*SDLC*) model *waterfall* dilakukan perencanaan tentang sistem yang akan dibangun. Dalam hal ini *website* Penanganan Keluhan Fakultas Teknik.
- b. Tahap analisis, merupakan proses pendalaman mengenai segala permasalahan dan resiko pada pengguna.
- c. Tahap perancangan, menyangkut perancangan sistem dimana akan memberikan rencana solusi dari masalah yang muncul pada tahap analisis.

- d. Tahap implementasi, adalah tahapan dimana sistem diimplementasikan pada situasi nyata dengan pemilihan perangkat keras dan penyusunan desain (*coding*). Untuk implementasi yaitu dengan memberitahu *user*, melatih *user*, memasang sistem (*install* sistem).
- e. Tahap pengujian, tahap untuk menguji sistem sudah berjalan sesuai rencana yang sudah disepakati sebelumnya, termasuk pengujian masing-masing menu apa masih ada *error* atau tidak. Tujuan dari pengujian ini adalah untuk meminimalisir cacat desain *web* sehingga sistem yang dikembangkan benar-benar dapat berjalan dengan sebaik mungkin. Pengujian ini akan dilakukan *interview* dengan mewawancarai beberapa orang yang akan berkaitan dengan *web* Penanganan Keluhan Fakultas Teknik.
- f. Tahap pemeliharaan, adalah tahap dimana dilakukan perawatan dan pemeliharaan *web*. Jika diperlukan akan dilakukan perbaikan kecil kemudian jika periode sistem sudah habis akan masuk lagi pada tahap perencanaan.

3.2 Tempat dan Waktu Penelitian

Tempat yang digunakan penulis dalam melakukan penelitian ini yaitu Fakultas Teknik Universitas Muhammadiyah Yogyakarta. Adapun waktu penelitian tentang Penanganan Keluhan Fakultas Teknik. Ini dilaksanakan dari bulan April sampai Juni 2016.

3.3 Subject Penelitian

Sesuai dengan informasi yang dibutuhkan dalam penelitian, maka yang ditetapkan sebagai subyek penelitian adalah *website* yang ada di Universitas Muhammadiyah Yogyakarta serta *user* yang akan terlibat dengan *website* yang akan dibuat nantinya.

3.4 Alat dan Bahan Penelitian

Penelitian membutuhkan piranti-piranti untuk mendukung berjalannya perancangan dan implementasi *website*, antara lain:

3.4.1 Perangkat Keras

Personal Computer (PC)/Laptop

- a. *32/64 bit architecture processor*
- b. *4 GB Random Access Memmory (RAM)*

3.4.2 Perangkat Lunak

a. *phpDesigner*

Perangkat lunak *phpDesigner* merupakan alat utama dalam melaksanakan pembuatan *web* Penanganan Keluhan Fakultas Teknik.

b. *MySql*

Sebagai alat yang digunakan untuk membuat dan mengelola *database* beserta isinya.

c. *Xampp*

Sebagai alat yang digunakan untuk menjadi sebuah *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server, MySQL database*, dan penerjemah Bahasa yang ditulis dengan Bahasa pemrograman *PHP* dan *Perl*.

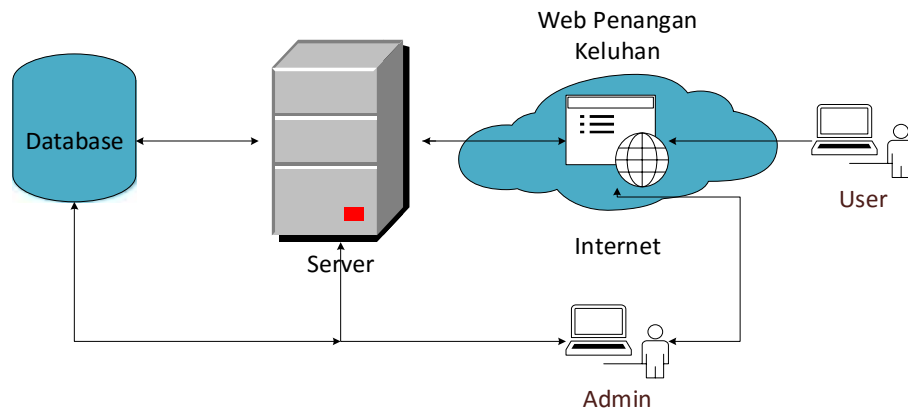
d. *Bootstrap*

Sebagai alat yang digunakan untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain *website*

e. *Sistem Operasi Windows 8*

Windows 8 adalah nama dari versi terbaru *Microsoft Windows*, serangkaian sistem operasi yang diproduksi oleh *Microsoft* untuk digunakan pada komputer pribadi, termasuk komputer rumah dan bisnis, laptop, *netbook, tablet PC, server*, dan PC pusat media. Sistem operasi ini menggunakan mikroprocessor ARM selain mikroprosesor x86 tradisional buatan intel dan AMD.

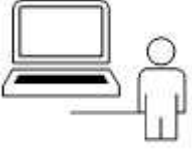

3.5 Arsitektur

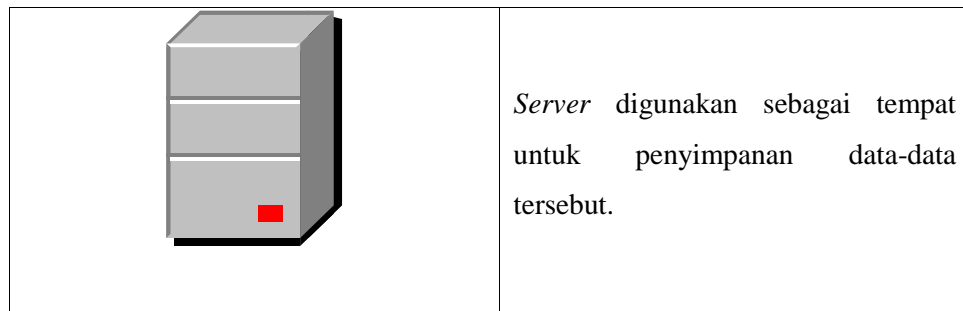


Gambar 3.2. Arsitektur Penanganan Keluhan

Database server yang digunakan pada aplikasi *MySQL* dan menggunakan *Xampp* sebagai *web server*. Komunikasi antar pengguna dan *web server* menggunakan internet dan *web browser* pada perangkat pengguna. Saat pengguna mengakses aplikasi, *web server* memuat antarmuka dan melakukan pengambilan data yang diperlukan dari *database server*. Melalui antarmuka yang dimuat *web server* pengguna bisa menyimpan data ke *database server*.

Tabel 3.1. Keterangan komponen arsitektur penanganan keluhan

	<p><i>User</i> <i>Admin</i></p>
	<p>Berisi data-data sebagai berikut:</p> <ol style="list-style-type: none"> 1. Data Petugas 2. Data <i>Group</i> Akses 3. Data Mahasiswa 4. Data Prodi 5. Data Fakultas 6. Data Kategori 7. Data Pengaduan 8. Data tahun 9. Data <i>setting</i>



3.6 Teknik Pengumpulan Data

Tahap pada alur teknik pengumpulan data digambarkan dalam *flow chart* pada gambar 3.3. Penelitian sistem penanganan keluhan berbasis *web* studi kasus Fakultas Teknik Universitas Muhammadiyah Yogyakarta menggunakan metode SDLC, hal ini bertujuan apabila dalam perjalanan sistem tersebut terdapat kesalahan, kerusakan ataupun *error* maka dilakukan analisis kebutuhan kembali dari awal memperbaiki sistem. Alur penelitian penulis dilakukan dalam beberapa tahap sebagai berikut:

1. Menganalisis kebutuhan

Analisis kebutuhan pada dasarnya merupakan tahap merancang dan membangun sebuah sistem informasi. Analisis kebutuhan mencakup kebutuhan *software* dan kebutuhan *hardware*, analisis kebutuhan isi dan interaksi menu pada aplikasi. Sebelum membuat program aplikasi berbasis *web*, terlebih dahulu melakukan wawancara terhadap beberapa orang terkait pengguna. Wawancara yang dilakukan akan melengkapi data dari kuisisioner yang sangat terbatas jumlah data yang dapat diterima. Wawancara akan memberikan data terkait bagaimana fitur-fitur yang diharapkan oleh pengguna berupa *web* agar *web* yang dibuat sesuai dengan harapan pengguna.



Gambar 3.3. Alur Teknik Pengumpulan Data

2. Pengumpulan data dan menentukan kebutuhan

Pengumpulan data berasal dari *requirement* yang telah ditentukan berdasarkan penggabungan data primer dan sekunder. *requirement* merupakan daftar kebutuhan dan persyaratan dari aplikasi. Dengan adanya *requirement*, pembuatan aplikasi akan dapat terarah dan terstruktur. Selain itu, *requirement* juga dapat membantu dalam melakukan *testing* ketika aplikasi telah selesai dibuat.

3. Perancangan dan pembuatan sistem

Sebelum sistem atau aplikasi dibuat, penulis membuat rancangan dari aplikasi *web*. Pembuatan rancangan tersebut bertujuan agar *web* yang dibuat dapat sesuai dengan yang diharapkan dan tidak akan ada fitur yang dihilangkan atau tertinggal.

4. Pengujian sistem

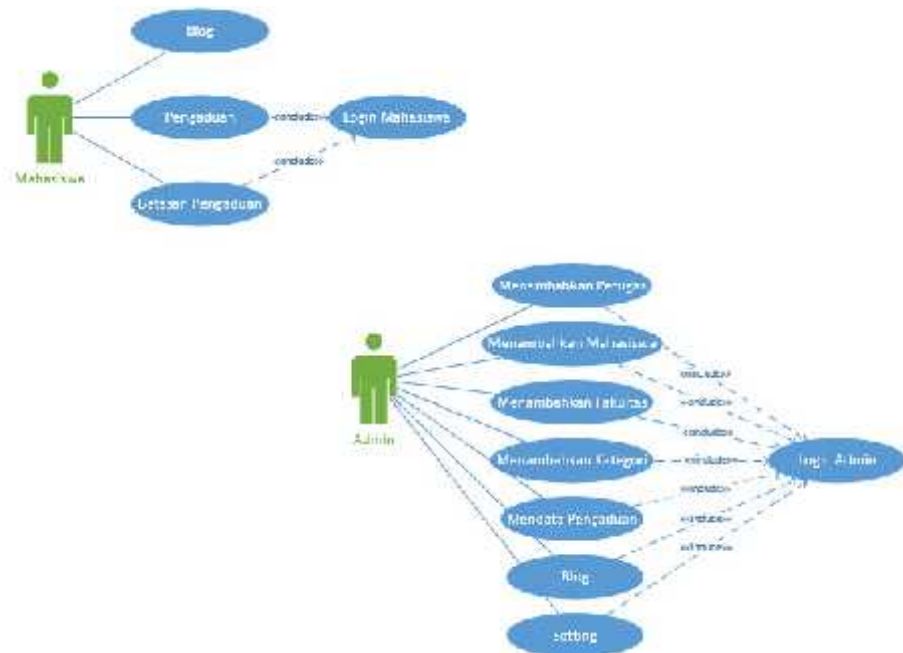
Sistem akan diuji sesuai dengan *requirement* yang telah ditentukan sebelumnya. Seluruh *requirement* harus terpenuhi dan tidak ada yang tertinggal ataupun tidak sesuai dengan *requirement*. Pengujian akan dilakukan dengan menggunakan metode *balckbox*.

3.7 Rancangan

Dalam pembuatan aplikasi dilakukan perancangan *database* menggunakan bantuan Diagram ER. Metode perancangan lain yang digunakan dalam aplikasi adalah *Unified Markup Language* (UML). Model UML yang dipakai dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

3.7.1 Use Case Diagram

Gambaran *Use Case Diagram* yang digunakan dalam aplikasi dapat dilihat pada gambar 3.4.



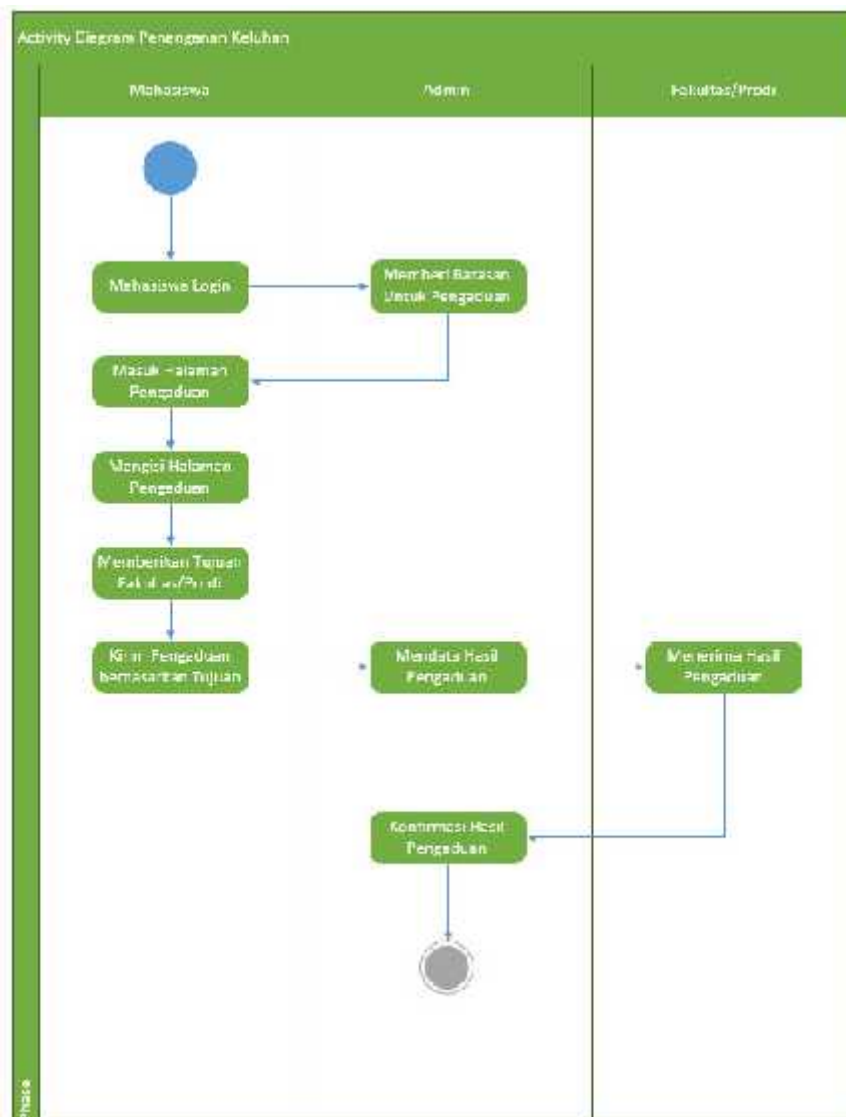
Gambar 3.4. Use Case Diagram

Berikut Penjelasan tentang Gambar 3.4:

1. Terdapat 2 actor pada use case diagram aplikasi yakni mahasiswa dan admin
2. Actor mahasiswa tanpa melakukan login mahasiswa hanya bisa mengakses blog.
3. Pada use case login mahasiswa berhubungan include dengan use case pengaduan dan batasan pengaduan yang artinya bahwa use case login mahasiswa memerlukan use case pengaduan dan batasan pengaduan untuk melakukan tugasnya.
4. Pada use case login admin berhubungan include dengan use case menambahkan petugas, menambahkan mahasiswa, menambahkan fakultas, menambahkan kategori, mendata pengaduan, blog, dan setting yang artinya bahwa use case login admin memerlukan use case menambahkan petugas, menambahkan mahasiswa, menambahkan fakultas, menambahkan kategori, mendata pengaduan, blog, dan setting untuk melakukan tugasnya.

3.7.2 Activity Diagram

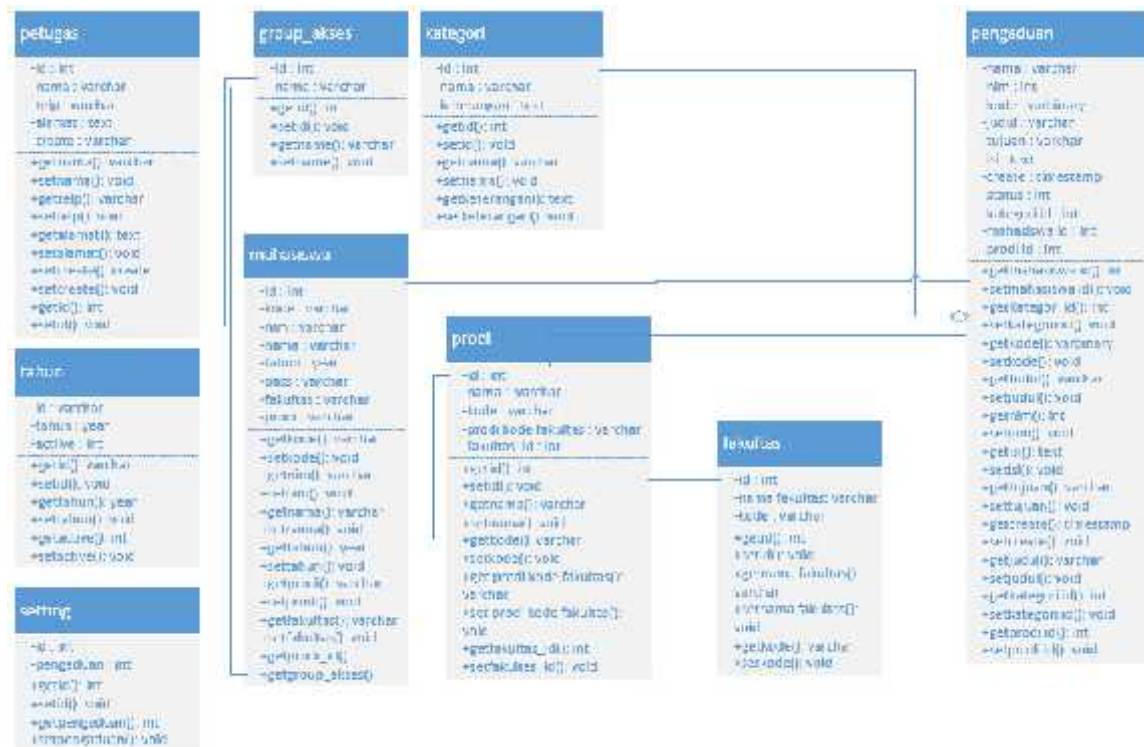
Gambar 3.5 menunjukkan *Activity Diagram* pada kegiatan *web* penanganan keluhan. Mahasiswa melakukan *login*, setelah itu admin memberikan batasan untuk pengaduannya, mahasiswa masuk ke halaman pengaduan, kemudian mahasiswa mengisi halaman pengaduan, mahasiswa memilih tujuan untuk ke pengaduannya yaitu fakultas atau prodi, kemudian mahasiswa mengirim pengaduan berdasarkan tujuannya. Admin mendata hasil pengaduannya, kemudian admin memberikan hasil pengaduannya ke fakultas atau prodi yang dituju pada mahasiswa. Lalu fakultas atau pun prodi memberikan konfirmasi hasil pengaduan tersebut ke pada admin.



Gambar 3.5. Activity Diagram

3.7.3 Class Diagram

Gambaran *Class Diagram* yang digunakan dalam aplikasi dapat dilihat pada gambar 3.6.



Gambar 3.6. *Class Diagram*

Berikut adalah penjelasan *class* diagram pada gambar 3.6.

- Pada *class* petugas, memiliki fungsi untuk menyimpan data petugas, di dalam *class* petugas petugas bisa melakukan menambah petugas, *edit* petugas dan menghapus data petugas,
- Pada *class* group_akses, memiliki fungsi untuk memberikan akses pada petugas dan mahasiswa untuk melakukan akses pengaduan.
- Pada *class* mahasiswa memiliki fungsi menyimpan data mahasiswa untuk melakukan akses pengaduan, di dalam *class* mahasiswa petugas dapat melakukan menambah mahasiswa dan menghapus data mahasiswa.
- Pada *class* fakultas, memiliki fungsi untuk memberikan tujuan pada setiap pengaduan ke masing-masing fakultas, di dalam *class* fakultas petugas dapat melakukan menambah fakultas dan menghapus data fakultas.

- e. Pada *class* prodi, memiliki fungsi untuk memberikan tujuan pada setiap pengaduan ke masing-masing prodi, di dalam *class* prodi petugas dapat melakukan menambah prodi dan menghapus data prodi.
- f. Pada *class* kategori, memiliki fungsi untuk memberikan tujuan pada setiap pengaduan yang terdiri dari fasilitas, pelayanan, dan akademik, di dalam *class* kategori petugas dapat melakukan menambah kategori dan menghapus kategori.
- g. Pada *class* pengaduan, memiliki fungsi untuk mengumpulkan data setiap pengaduan dari mahasiswa, di dalam *class* prodi petugas bisa mendata pengaduan berdasarkan fakultas, prodi, kategori, dan tanggal, petugas juga bisa melakukan konfirmasi pengaduan yang sudah ditangani dan petugas juga bisa menghapus data pengaduan, sedangkan mahasiswa bisa melakukan isi judul, memilih tujuan antara prodi dan fakultas, memilih kategori.
- h. Pada *class* tahun, memiliki fungsi untuk melakukan *input* data tahun angkatan mahasiswa, di dalam *class* tahun petugas dapat melakukan menambahkan tahun angkatan dan mengaktifkan tahun angkatan.
- i. Pada *class* *setting*, memiliki fungsi untuk megubah batasan akses pengaduan pada setiap mahasiswa, di dalam *class* *setting* petugas bisa merubah batasan pengaduan untuk mahasiswa.

Class petugas memiliki *association* dengan *class* *group_akses*, *class* petugas dapat mengakses apa saja yang tersedia di dalam *website* melalui *method* *Getid()*.

Class *group_akses* memiliki *association* dengan *class* mahasiswa, *class* mahasiswa dapat mengakses pengaduan melalui *method* *Getid()*.

Class kategori memiliki *composition* dengan *class* pengaduan, artinya *class* kategori merupakan bagian dari *class* pengaduan. *Class* kategori tidak dapat berdiri sendiri apabila *class* pengaduan tidak ada.

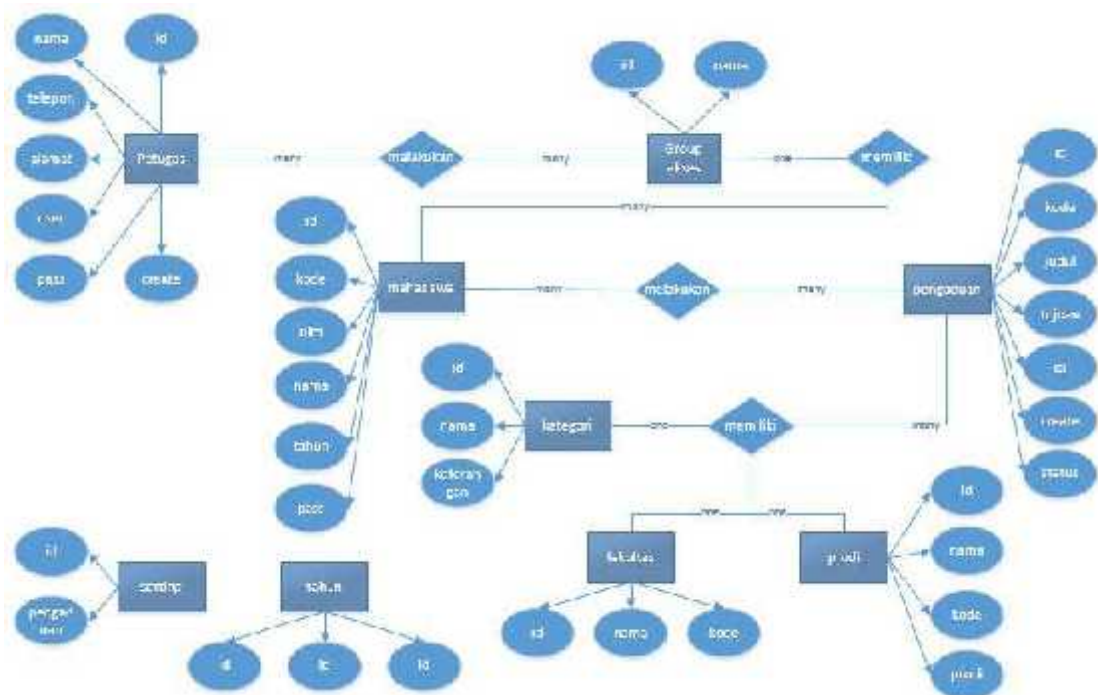
Class mahasiswa memiliki *association* dengan *class* pengaduan, *class* mahasiswa boleh melakukan akses pengaduan melalui *method* *Getmahasiswaid()*.

Class mahasiswa memiliki *association* dengan *class* prodi, *class* mahasiswa dapat mengakses prodi melalui *method* Getid().

Class fakultas memiliki *association* dengan *class* prodi, *class* fakultas dapat mengakses prodi melalui *method* Getid().

3.7.4 ER Diagram

Gambaran *ER* Diagram yang digunakan dalam aplikasi dapat dilihat pada gambar 3.7



Gambar 3.7. ER Diagram

Pada gambar 3.7 dapat dilihat *database* yang dirancang memiliki 9 buah entitas yaitu:

- Petugas
- Group_akses
- Mahasiswa
- Pengaduan
- Kategori
- Fakultas
- Prodi

h. Tahun

i. *Setting*

Pada entitas tahun dan *setting* tidak terdapat relasi ke entitas lainnya, relasi antar entitas dimiliki oleh entitas petugas berelasi *one-to-one* dengan entitas group_akses serta entitas mahasiswa berelasi *one-to-many* dengan entitas pengaduan, dan entitas kategori berelasi *many-to-many* dengan entitas fakultas dan entitas prodi.

Relasi *one-to-one* antara entitas petugas dan entitas group_akses mempunyai arti bahwa satu data pada entitas petugas hanya bisa mempunyai satu data pada entitas group_akses, dan satu data entitas group_akses hanya bisa mempunyai satu data pada entitas petugas.

Relasi *one-to-many* antara entitas mahasiswa dan entitas pengaduan mempunyai arti bahwa satu data pada entitas mahasiswa bisa mempunyai banyak data pada entitas pengaduan, sedangkan setiap data pada entitas pengaduan hanya mempunyai satu data pada entitas mahasiswa

Relasi *many-to-many* antara entitas kategori dengan entitas fakultas, entitas prodi mempunyai arti bahwa satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua. Artinya ada banyak baris di tabel satu dan tabel dua yang saling berhubungan satu sama lain. Relasi antar entitas kategori dan entitas fakultas. Satu baris prodi bisa berhubungan dengan banyak baris fakultas begitu juga sebaliknya. Tabel-tabel tersebut antara lain:

Tabel 3.2. Struktur tabel kategori

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Kat_id	PK	NN	Int	11
2	Kat_nama		NN	Varchar	50
3	Kat_keterangan		NN	Text	

Tabel 3.3. Struktur tabel petugas

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Ptg_id	PK	NN	Int	11
2	Ptg_nama		NN	Varchar	30
3	Ptg_telp		NN	Varchar	15
4	Ptg_alamat		NN	Text	
5	Ptg_user		NN	Varchar	10
6	Ptg_pass		NN	Varchar	10
7	Ptg_create		NN	Timestamp	
8	Ga_id		NN	Int	11

Tabel 3.4. Struktur tabel *group_akses*

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Ga_id	PK	NN	Int	11
2	Ga_nama		NN	Varchar	50

Tabel 3.5. Struktur tabel mahasiswa

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Mhs_id	PK	NN	Int	11
2	Mhs_kode		NN	Varchar	20
3	Mhs_nim		NN	Varchar	50
4	Mhs_nama		NN	Varchar	50
5	Mhs_tahun		NN	Year	4
6	Mhs_pass		NN	Varchar	50
7	Pr_id		NN	Int	11
8	Ga_id		NN	Int	11

Tabel 3.6. Struktur tabel pengaduan

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Pg_id	PK	NN	Int	11
2	Pg_kode		NN	Varbinary	50
3	Pg_judul		NN	Varchar	100
4	Pg_tujuan		NN	Varchar	100
5	Pg_isi		NN	Text	
6	Pg_create		NN	Timestamp	
7	Pg_status		NN	Int	11
8	Pr_id		NN	Int	11
9	Kat_id		NN	Int	11
10	Mhs_id		NN	int	11

Tabel 3.7. Struktur tabel fakultas

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Fk_id	PK	NN	Int	11
2	Fk_nama		NN	Varchar	100
3	Fk_kode		NN	Varchar	100

Tabel 3.8. Struktur tabel prodi

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Pr_id	PK	NN	Int	11
2	Pr_nama		NN	Varchar	50
3	Pr_kode		NN	Varchar	50
4	Pr_kode_fk		NN	Varchar	10
5	Fk_id		NN	Int	10

Tabel 3.9. Struktur tabel tahun

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Th_id	PK	NN	Int	11
2	Th_tahun		NN	Year	4
3	Th_active		NN	Int	11

Tabel 3.10. Struktur tabel *setting*

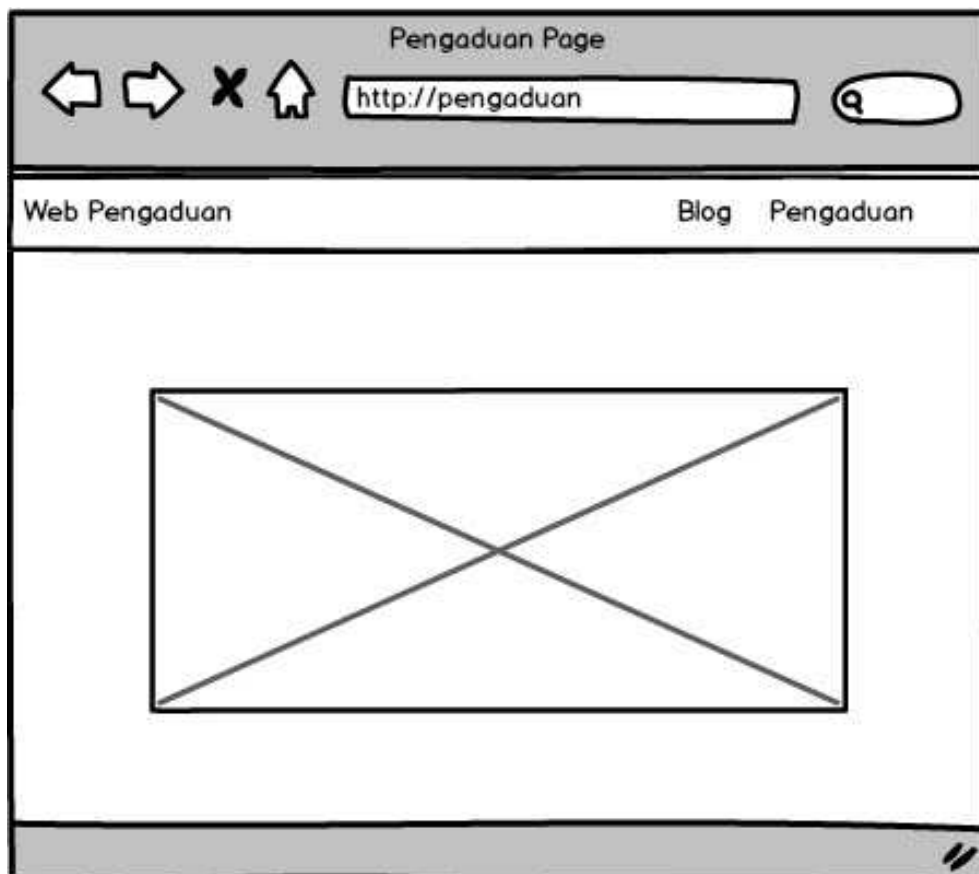
No	Field Name	Key Type	Null	Data Type	Max. Length
1	Set_id	PK	NN	Int	11
2	Set_pengaduan		NN	Int	11

3.7.5 Rancangan Antarmuka (*User Interface*)

User Interface sangatlah penting dalam suatu aplikasi karena merupakan bagian dari perangkat lunak yang menjadi sarana komunikasi antar pengguna dengan sistem serta dapat memberikan kemudahan bagi pengguna dalam melakukan aktivitasnya.

3.7.5.1 Rancangan Antarmuka Halaman Utama

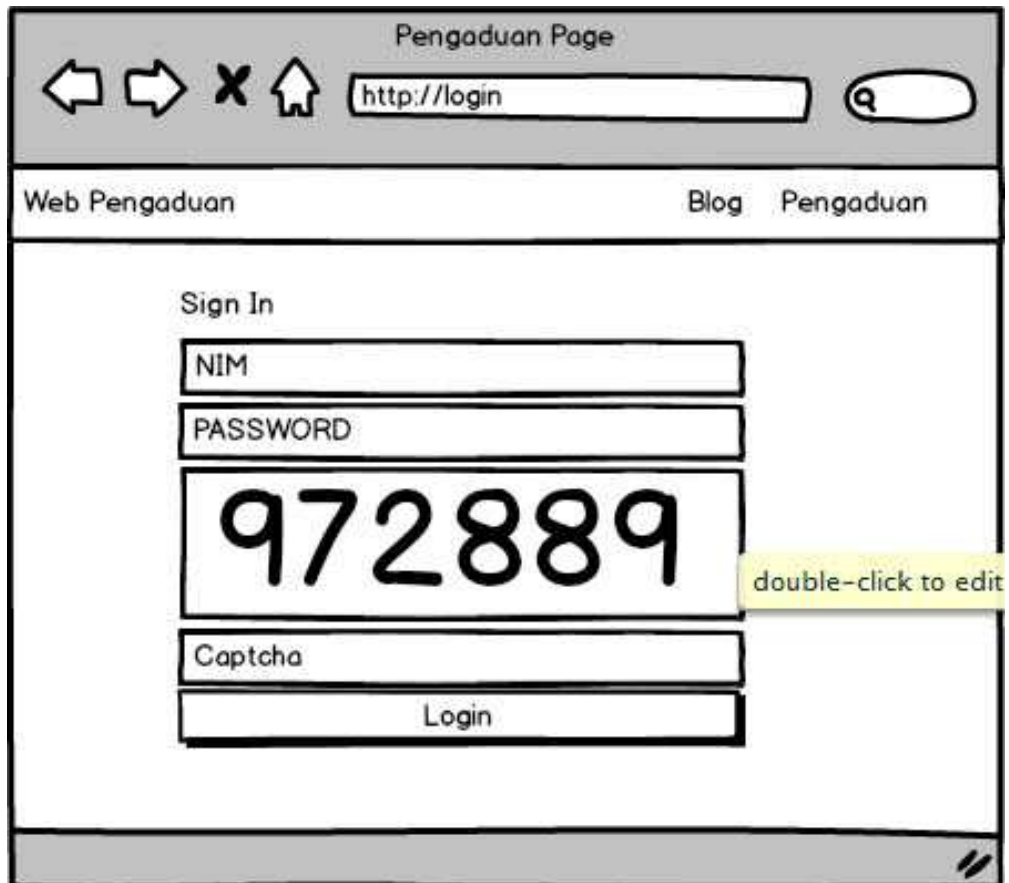
Rancangan antarmuka halaman utama merupakan tampilan utama dari aplikasi yang dapat dilihat oleh mahasiswa dan admin. Gambaran rancangan antarmuka halaman utama aplikasi dapat dilihat pada Gambar 3.8.



Gambar 3.8. Rancangan Halaman Utama Aplikasi

3.7.5.2 Rancangan antarmuka halaman *login*

Rancangan untuk halaman *login* mahasiswa. Berfungsi untuk dapat masuk ke halaman pengaduan. Gambaran rancangan antarmuka menu *login* dapat dilihat pada Gambar 3.9.

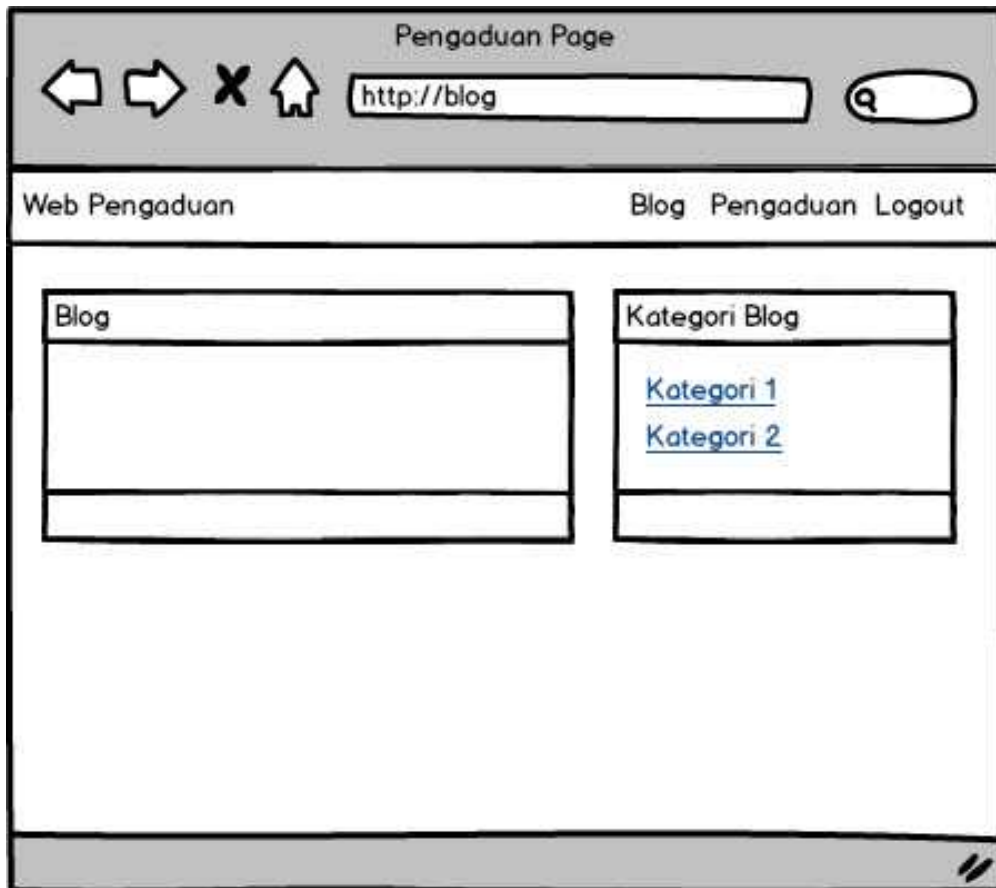


The image shows a wireframe of a web browser window. The browser's title bar reads "Pengaduan Page". The address bar contains "http://login". Below the browser window, there is a navigation bar with "Web Pengaduan" on the left and "Blog" and "Pengaduan" on the right. The main content area is titled "Sign In" and contains a form with the following elements: a text input field labeled "NIM", a text input field labeled "PASSWORD", a captcha image displaying the number "972889", a text input field labeled "Captcha", and a "Login" button. A yellow tooltip with the text "double-click to edit" is positioned over the captcha image. The browser window has standard navigation icons (back, forward, stop, home) and a search icon.

Gambar 3.9. Rancangan Halaman *login*

3.7.5.3 Rancangan antarmuka halaman *blog*

Rancangan antarmuka halaman *blog* yang dapat dilihat mahasiswa dan admin. Halaman *blog* berfungsi untuk memberikan informasi tentang tata cara menggunakan *web* pengaduan. Gambaran rancangan antarmuka halaman *blog* dapat dilihat pada Gambar 3.10.



Gambar 3.10. Rancangan Halaman *blog*

3.7.5.4 Rancangan antarmuka halaman pengaduan

Rancangan antarmuka halaman pengaduan merupakan halaman yang digunakan untuk memasukkan keluhan atau aspirasi yang di isi oleh mahasiswa beserta tujuannya. Gambaran rancangan antarmuka halaman pengaduan dapat dilihat pada Gambar 3.11.

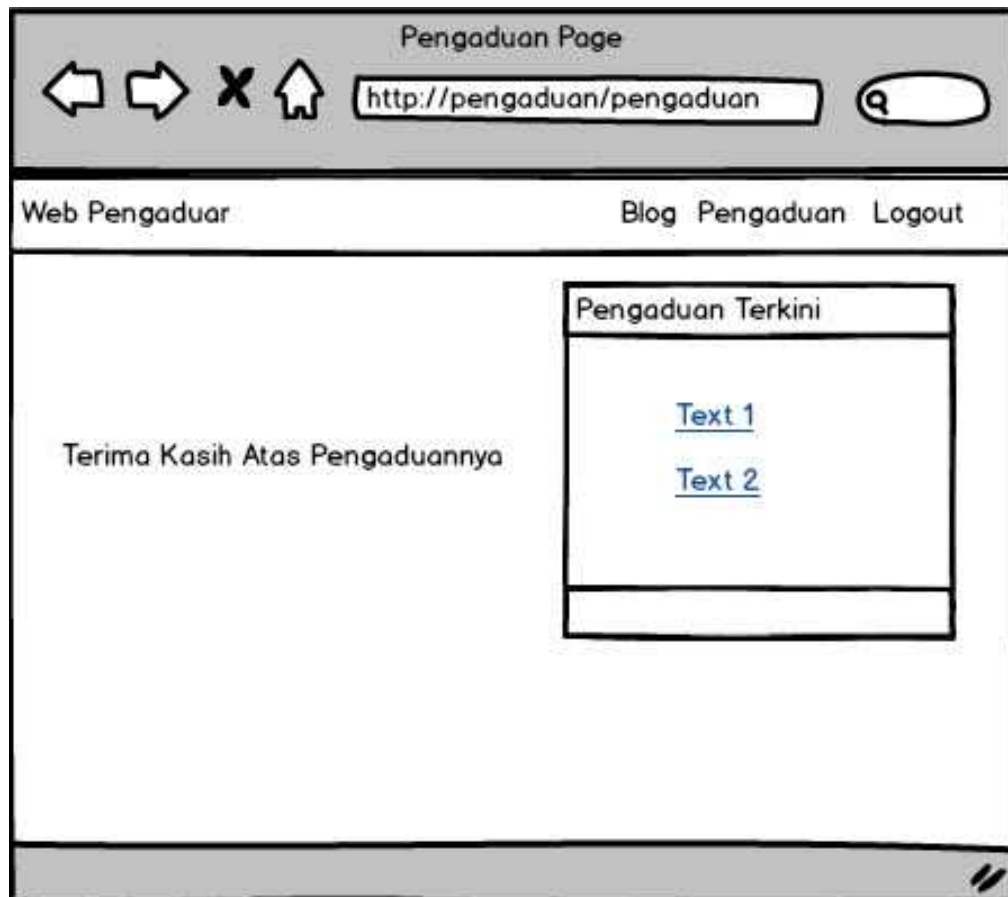
The image shows a web browser window titled "Pengaduan Page" with the URL "http://pengaduan/pengaduan". The page content is as follows:

- Navigation: "Web Pengaduan" (left), "Blog Pengaduan Logout" (right).
- Form Title: "Pengaduan".
- Fields:
 - Nama: Text input containing "Gerizky Saputra".
 - Nim: Text input containing "20120140031".
 - Judul: Empty text input.
 - Tujuan Pengaduan: Radio buttons for "Fakultas" and "Prad".
 - Kategori: Dropdown menu labeled "Pilih Kategori".
 - Rich text editor with a toolbar containing icons for bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, and smiley.
- Buttons: "Kirim" button at the bottom right of the form.
- Right Sidebar: "Pengaduan Terbaru" section containing two links: "Text 1" and "Text 2".

Gambar 3.11. Rancangan Halaman pengaduan

3.7.5.5 Rancangan antarmuka halaman selesai pengaduan

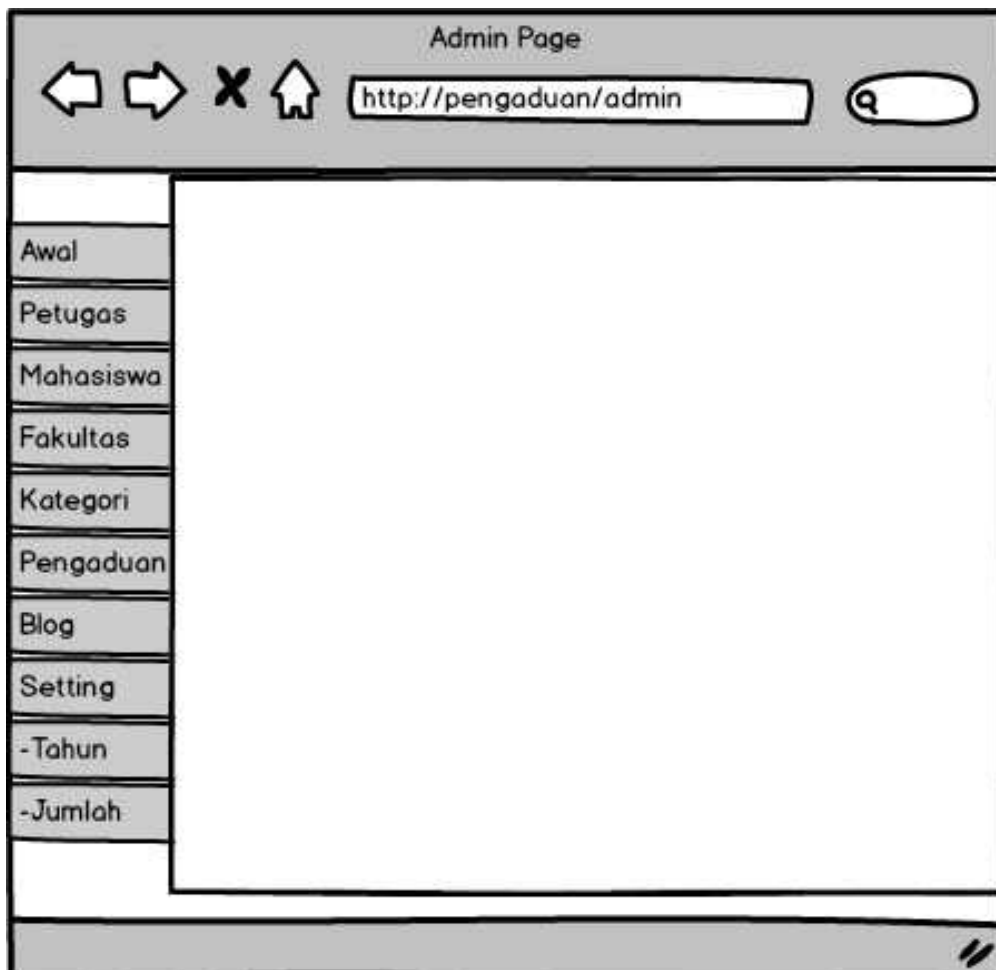
Rancangan antarmuka halaman selesai pengaduan merupakan halaman yang digunakan untuk memastikan bahwa mahasiswa sudah memberikan aspirasi atau keluhan. Gambaran rancangan antarmuka halaman selesai pengaduan dapat dilihat pada Gambar 3.12.



Gambar 3.12. Rancangan Halaman selesai pengaduan

3.7.5.6 Rancangan antarmuka halaman admin

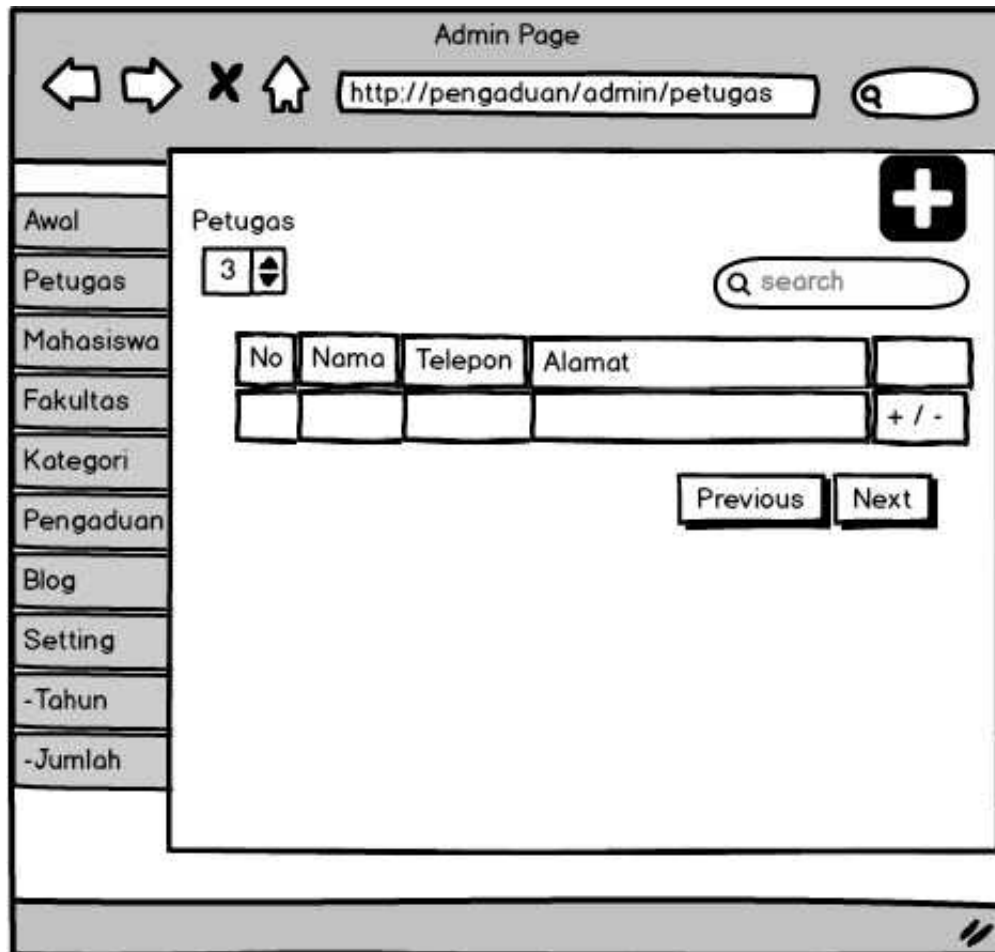
Rancangan antarmuka halaman utama admin merupakan tampilan utama dari aplikasi yang dapat dilihat oleh admin. Gambaran rancangan antarmuka halaman admin dapat dilihat pada Gambar 3.13.



Gambar 3.13. Rancangan Halaman admin

3.7.5.7 Rancangan antarmuka halaman petugas

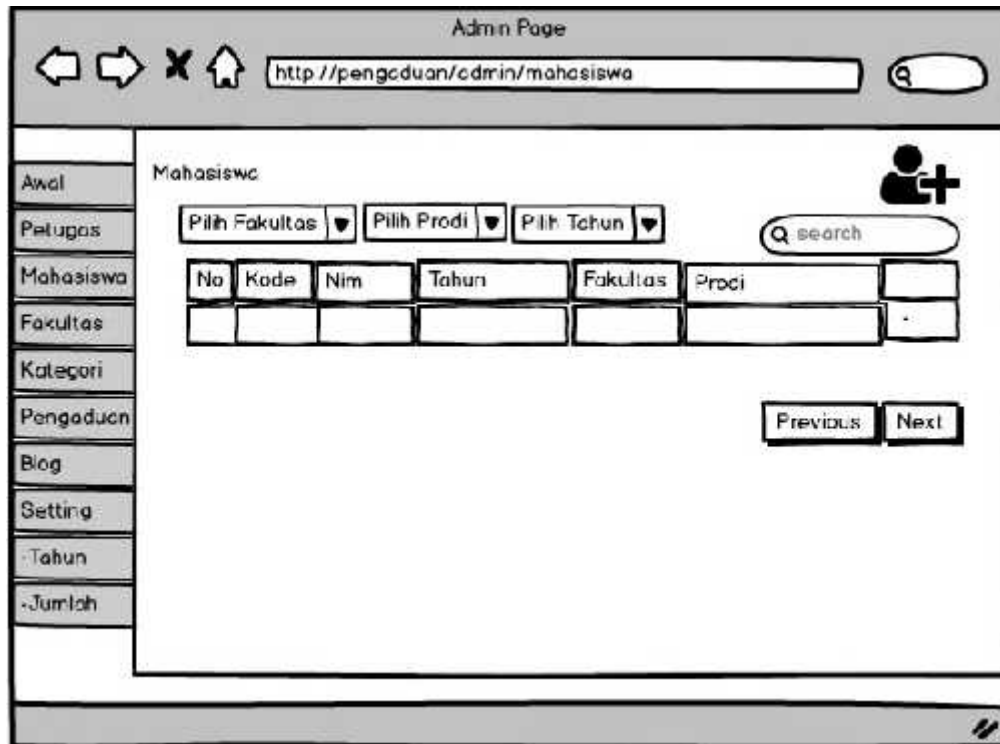
Rancangan antarmuka halaman petugas merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mencari petugas. Gambaran rancangan antarmuka halaman petugas dapat dilihat pada Gambar 3.14.



Gambar 3.14. Rancangan Halaman petugas

3.7.5.8 Rancangan antarmuka halaman mahasiswa

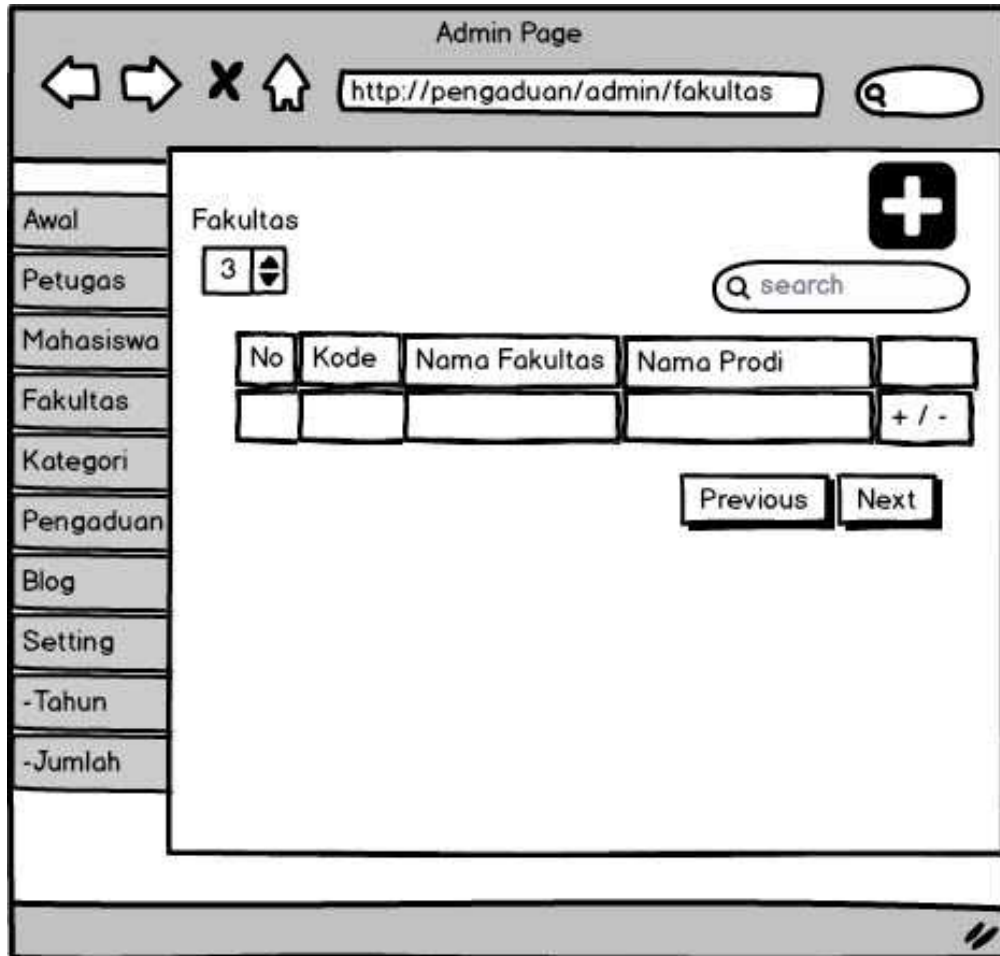
Rancangan antarmuka halaman mahasiswa merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mencari mahasiswa sesuai fakultas, prodi dan tahun angkatan. Gambaran rancangan antarmuka halaman mahasiswa dapat dilihat pada Gambar 3.15.



Gambar 3.15. Rancangan Halaman mahasiswa

3.7.5.9 Rancangan antarmuka halaman fakultas

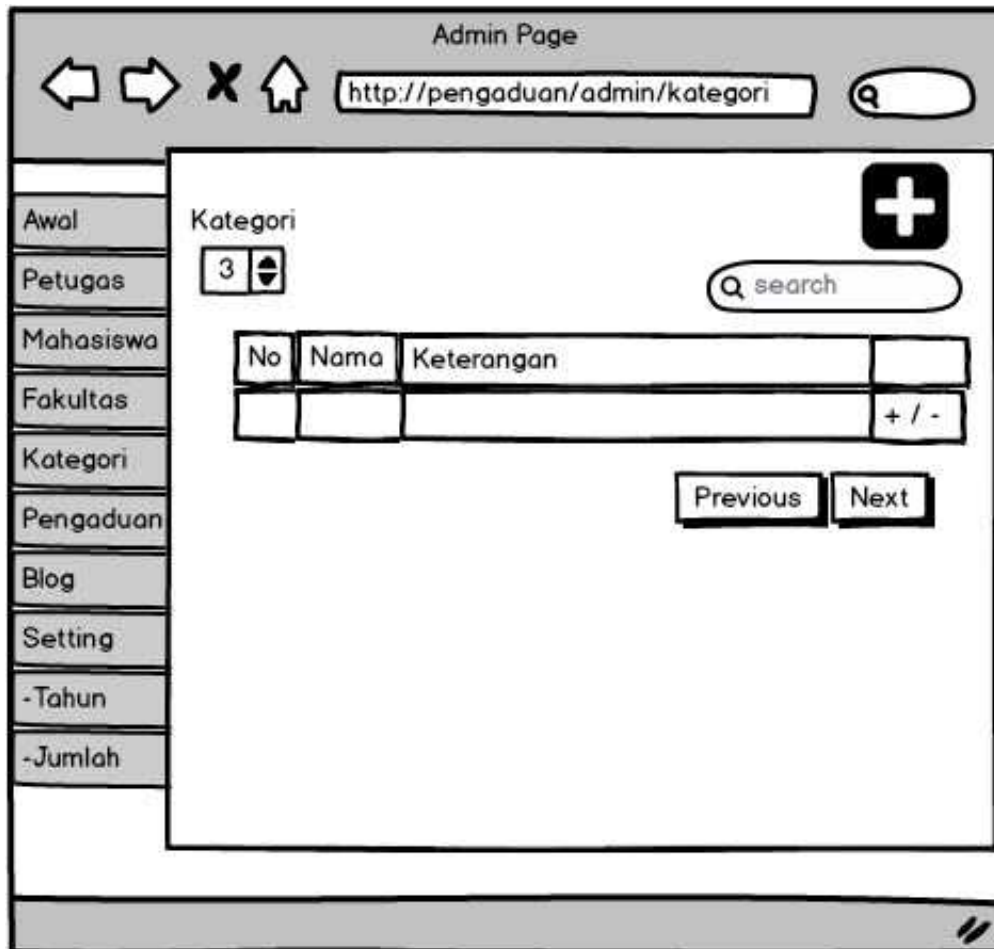
Rancangan antarmuka halaman fakultas merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mengubah data fakultas dan prodi. Gambaran rancangan antarmuka halaman fakultas dapat dilihat pada Gambar 3.16.



Gambar 3.16. Rancangan Halaman fakultas

3.7.5.10 Rancangan antarmuka halaman kategori

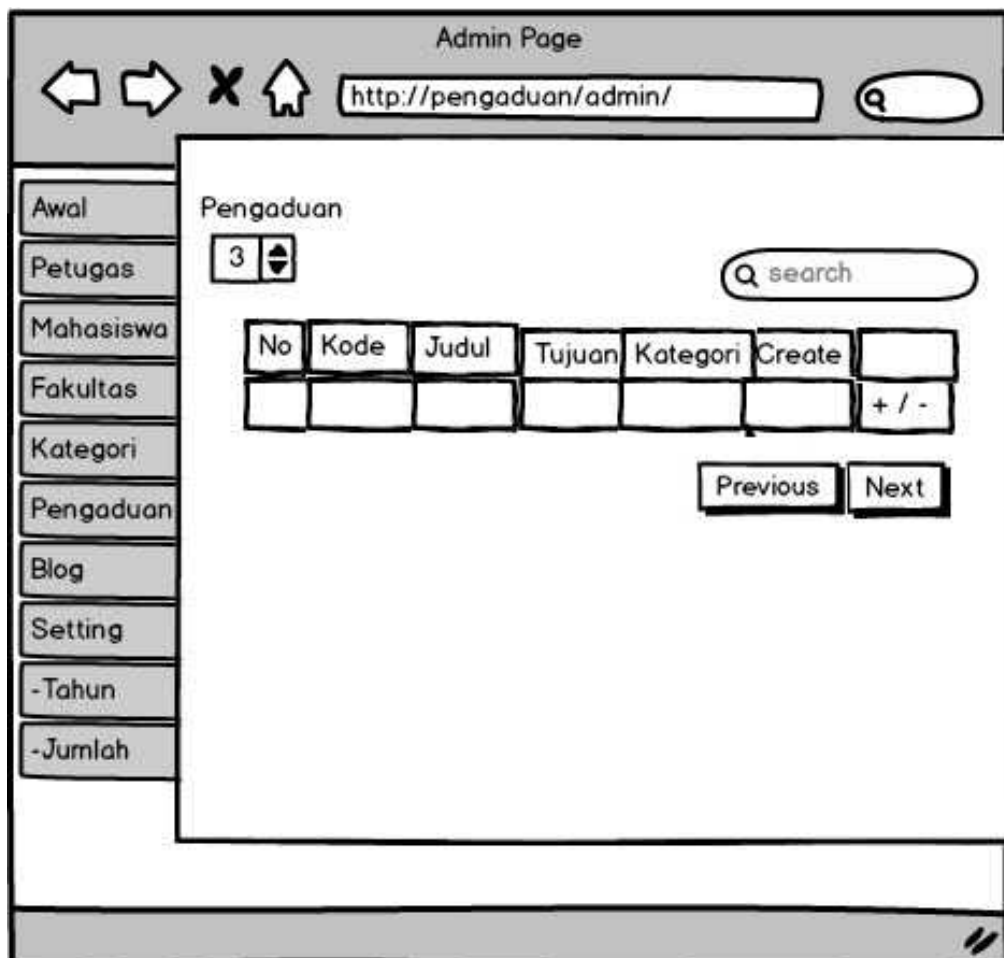
Rancangan antarmuka halaman kategori merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mengubah data tujuan kategori. Gambaran rancangan antarmuka halaman kategori dapat dilihat pada Gambar 3.17.



Gambar 3.17. Rancangan Halaman kategori

3.7.5.11 Rancangan antarmuka halaman pengaduan

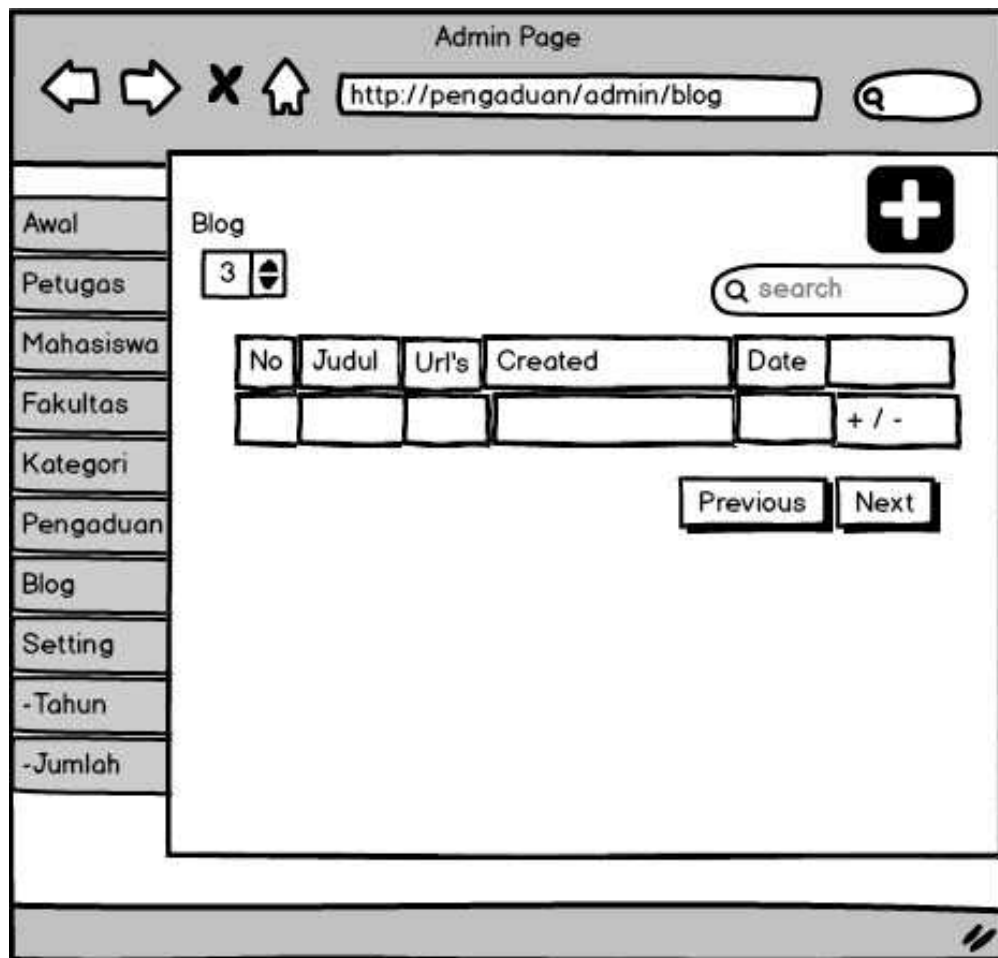
Rancangan antarmuka halaman pengaduan merupakan halaman yang digunakan untuk admin mendata keluhan atau aspirasi mahasiswa berdasarkan tujuan dan mengkonfirmasi data yang sudah dijalankan berdasarkan hasil keluhan dan aspirasi. Gambaran rancangan antarmuka halaman pengaduan dapat dilihat pada Gambar 3.18.



Gambar 3.18. Rancangan Halaman pengaduan

3.7.5.12 Rancangan antarmuka halaman pengeditan *blog*

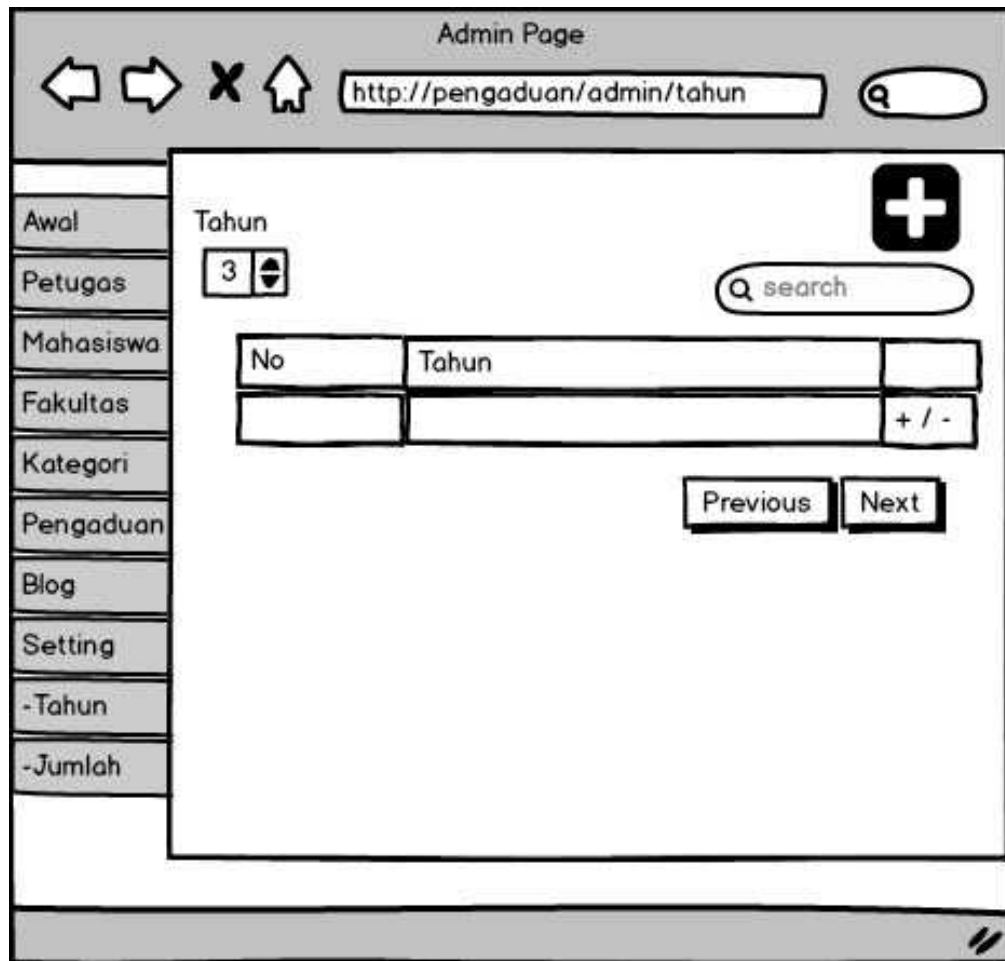
Rancangan antarmuka halaman pengeditan *blog* merupakan halaman yang digunakan untuk admin mengisi, mengedit, dan menambahkan isi *blog* dari admin. Gambaran rancangan antarmuka halaman pengeditan *blog* dapat dilihat pada Gambar 3.19.



Gambar 3.19. Rancangan Halaman pengeditan *blog*

3.7.5.13 Rancangan antarmuka halaman tahun

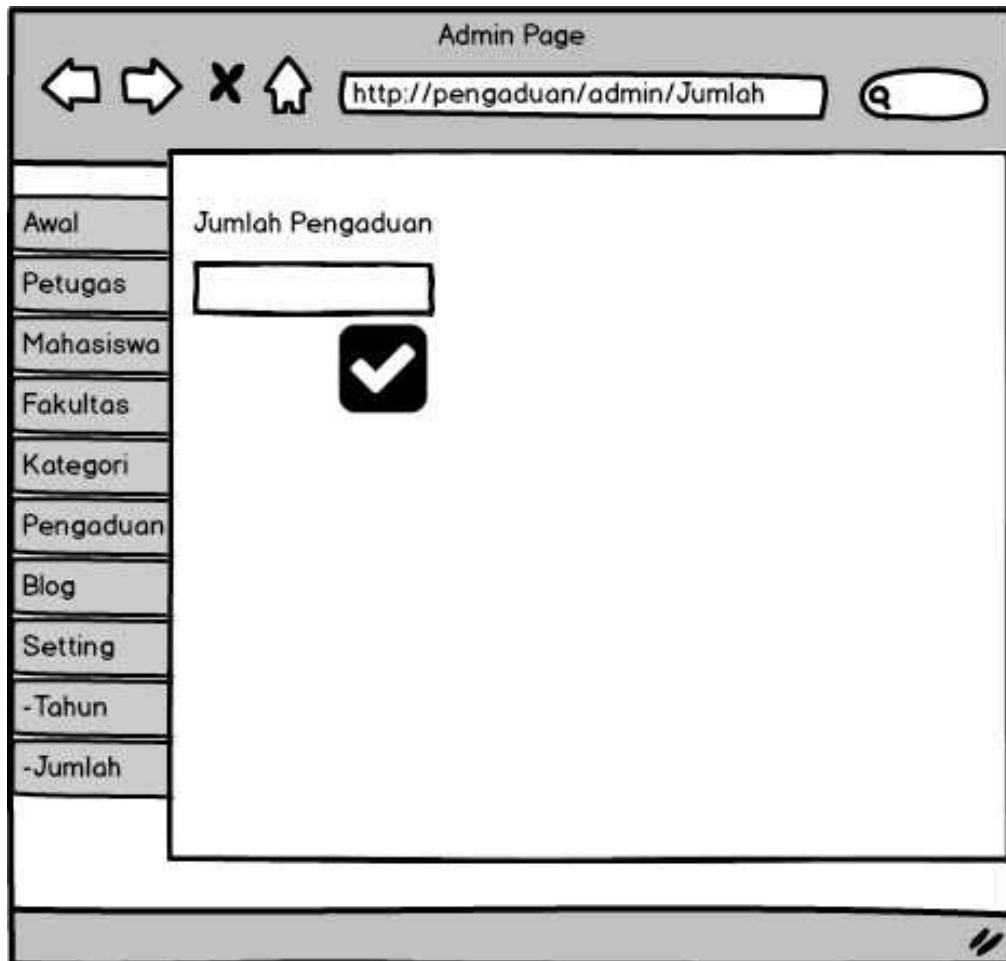
Rancangan antarmuka halaman tahun merupakan halaman yang digunakan untuk admin mengedit tahun angkatan untuk menambahkan mahasiswa berdasarkan tahun angkatan mahasiswa. Gambaran rancangan antarmuka halaman tahun dapat dilihat pada Gambar 3.20.



Gambar 3.20. Rancangan Halaman tahun

3.7.5.14 Rancangan antarmuka halaman jumlah

Rancangan antarmuka halaman jumlah merupakan halaman yang digunakan untuk admin mengedit batasan pengaduan agar tidak terjadi penumpukan data pengaduan. Gambaran rancangan antarmuka halaman jumlah dapat dilihat pada Gambar 3.21.



Gambar 3.21. Rancangan Halaman jumlah

3.8 Pengujian

Pengujian perangkat lunak merupakan suatu kegiatan yang dilakukan untuk memperoleh informasi serta mengevaluasi mengenai kualitas dari produk atau layanan yang sedang diuji. Tujuan pengujian dalam pengembangan aplikasi adalah untuk dapat memenuhi kebutuhan yang diperlukan dengan mendasari pada rancangan dan pengembangan perangkat lunak.

Metode pengujian yang dipakai dalam pengembangan aplikasi adalah *black box testing*. *Black box testing* atau tes fungsional adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak yang dikembangkan.

Hal-hal yang menjadi perhatian dalam pengujian adalah sebagai berikut:

- a. Aplikasi dapat menerima dan menyimpan hasil keluhan atau aspirasi mahasiswa.
- b. Aplikasi dapat menerima dan menyimpan hasil keluhan atau aspirasi berdasarkan kategori dan tujuan
- c. Aplikasi dapat menampilkan hasil keluhan atau aspirasi berdasarkan fakultas atau prodi.

