

**PENGEMBANGAN SISTEM PENANGANAN KELUHAN BERBASIS
WEB (STUDI KASUS: FAKULTAS TEKNIK UNIVERSITAS
MUHAMMADIYAH YOGYAKARTA)**

Skripsi

Untuk memenuhi persyaratan mencapai derajat Sarjana S-1

Program Studi Teknologi Informasi



Disusun oleh:

Gerizky Saputra

20120140031

**PROGRAM TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA**

2016

HALAMAN PENGESAHAN I

SKRIPSI

Pengembangan Sistem Penanganan Keluhan Berbasis *Web*

(Studi Kasus: Fakultas Teknik Universitas Muhammadiyah Yogyakarta)



Telah Disetujui Oleh:

Pembimbing 1

Pembimbing 2

Haris Setyawan, S.T., M.Eng.
NIK : 19691111200004123043

Aprilia Kurnianti, S.T., M.Eng.
NIK : 19840418201504123069

HALAMAN PENGESAHAN II

SKRIPSI

Pengembangan Sistem Penanganan Keluhan Berbasis *Web*

(Studi Kasus: Fakultas Teknik Universitas Muhammadiyah Yogyakarta)

Diajukan Oleh:

GERIZKY SAPUTRA

20120140031

Skripsi ini telah Dipertahankan dan Disahkan di depan Dewan Penguji Program

Studi Teknologi Informasi Fakultas Teknik Universitas Muhammadiyah

Yogyakarta

Tanggal 19 Agustus 2016

Dewan Penguji:



Haris Setyawan, S.T., M.Eng.
NIK: 19691111200004123043



Aprilia Kurnianti, S.T., M.Eng.
NIK: 19840418201504123069



Dr. Ir. Dwijoko Purbohadi, S.T., M.T.
NIK: 19680202199502123019

PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi ini tidak mengandung karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 19 Agustus 2016



Gerizky Saputra

PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan barokahnya sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Sistem Penanganan Keluhan Berbasis *Web* (Studi Kasus: Fakultas Teknik Universitas Muhammadiyah Yogyakarta)”.

Laporan skripsi ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Teknik (S.T.) pada Program Studi S1 Teknologi Informasi Universitas Muhammadiyah Yogyakarta.

Dalam melakukan penelitian dan penyusunan laporan skripsi ini penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terima kasih yang tak terhingga kepada:

1. Haris Setyawan, S.T., M.Eng., selaku dosen pembimbing pertama yang telah bersedia meluangkan waktu untuk mengarahkan pengembangan aplikasi penanganan keluhan, serta membimbing penulis dengan kesabaran dan ketulusan. Memberikan masukan perihal pengembangan aplikasi yang baik dan benar.
2. Aprilia Kurnianti, S.T., M.Eng., selaku dosen pembimbing kedua yang telah bersedia meluangkan waktu untuk mengarahkan pengembangan aplikasi penanganan keluhan terkhusus terkait penulisan dengan penuh kesabaran dan ketulusan membimbing dan memberikan ilmu kepada penulis.
3. Dr. Ir. Dwijoko Purbohadi, S.T., M.T. selaku dosen penguji yang telah memberikan masukan yang berarti pada saat pendadaran atau sidang skripsi.
4. Para staf jurusan Teknologi Informasi UMY, Mas Andhy Kurniawan dan Mas Haris Firnandita yang senantiasa sabar dan membantu penulis dalam urusan administrasi.
5. Mama dan Bapak yang dari penulis lahir sampai sekarang selalu memberikan kasih sayang dan doanya kepada penulis sehingga penulis dapat mencapai ke tahap sekarang.

6. Rahmat, Retno, Fizela, Luthfia, Reza, Hilman, Iqbal, Cinoy, Luthfi, Dafa, Sabiq sebagai pendorong semangat kepada penulis dalam menyelesaikan skripsi. *You were built to be strong, fearless, confident and powerful.*
7. Aci dan Indy sebagai kakak yang bersedia berbagi cerita, dorongan serta motivasi. *Take a deep breath and Bismillah.*
8. Teman–teman Teknologi Informasi 2012 Universitas Muhammadiyah Yogyakarta yang selalu saling memberikan dukungan untuk menyelesaikan tugas akhir.
9. Pihak-pihak lainnya yang tidak dapat disebutkan satu per satu, yang telah memberikan bantuan kepada saya, sehingga saya dapat menyelesaikan penyusunan skripsi ini.

INTISARI

Aspirasi dan keluhan dari mahasiswa pada Fakultas Teknik Universitas Muhammadiyah Yogyakarta saat ini disampaikan secara langsung kepada staf Fakultas atau melalui kegiatan *Public Hearing* yang dilaksanakan secara berkala. Bentuk pendataan aspirasi dan keluhan tersebut disampaikan dalam forum sehingga waktunya tidak mencukupi untuk menampung keseluruhan aspirasi dan keluhan. Tujuan perancangan ini adalah untuk menghasilkan aplikasi pengaduan yang dapat memudahkan mahasiswa menyampaikan pengaduan serta membantu fakultas atau prodi untuk mengawasi program studi dalam menanggapi pengaduan tersebut. Dari pengujian dapat disimpulkan bahwa sistem bekerja dengan baik, menyediakan sarana untuk menampung keluhan, meneruskan keluhan tersebut ke dosen, dan mencatat tindakan apa yang diambil oleh program studi.

Kata kunci: aplikasi pengaduan, keluhan mahasiswa, penanganan keluhan

ABSTRACT

Aspirations and complaints of the students at the Faculty of Engineering, Universitas Muhammadiyah Yogyakarta currently delivered directly to the staff of the Faculty or through Public Hearing event that conducted periodically. Data collection of the aspirations and the complaints submitted in the forum so the time was not sufficient to accommodate the overall aspirations and complaints. The purpose of this design is to produce applications that can facilitate student complaints that filed and assist faculty or department to oversee the department in response to the complaints. From the results test can be concluded that the system works well, providing the means to register complaints, forwarding the complaints to the faculty, and note any actions taken by the department.

Keywords: application complaints, student complaints, the handling of complaints

DAFTAR ISI

HALAMAN PENGESAHAN I	Error! Bookmark not defined.
HALAMAN PENGESAHAN II	ii
PERNYATAAN.....	iii
PRAKATA	v
INTISARI.....	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.1.1 Rumusan Masalah.....	2
1.2 Tujuan Penelitian.....	3
1.3 Manfaat Penelitian.....	3
1.4 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Tinjauan Pustaka	5
2.2 Landasan Teori	8
2.2.1 Sistem Informasi	8
2.2.2 Pengertian Penanganan Keluhan	10
2.2.3 Teknologi Pengembangan Aplikasi	11
2.2.4 Pengembangan Sistem Berbasis <i>Web</i>	14
2.2.5 <i>PHP</i>	15
2.2.6 <i>MySQL</i>	15
2.2.7 <i>Software Development Life Cycle</i>	15
2.2.8 <i>Black Box Testing</i>	17
BAB III CARA DAN METODOLOGI PENELITIAN.....	18
3.1 Metodologi Penelitian	18
3.2 Tempat dan Waktu Penelitian	19

3.3 <i>Subject</i> Penelitian	19
3.4 Alat dan Bahan Penelitian	19
3.4.1 Perangkat Keras	20
3.4.2 Perangkat Lunak	20
3.5 Arsitektur	21
3.6 Teknik Pengumpulan Data	22
3.7 Rancangan	24
3.7.1 <i>Use Case</i> Diagram	24
3.7.2 <i>Activity</i> Diagram	26
3.7.3 <i>Class</i> Diagram.....	27
3.7.4 <i>ER</i> Diagram.....	29
3.7.5 Rancangan Antarmuka (<i>User Interface</i>).....	34
3.8 Pengujian	48
BAB IV HASIL DAN PEMBAHASAN	49
4.1 <i>Database</i>	49
4.2 Antarmuka	50
4.2.1 Halaman Utama	50
4.2.2 Halaman Artikel.....	50
4.2.3 Halaman <i>Login</i> Mahasiswa.....	51
4.2.4 Halaman Utama Pengaduan.....	51
4.2.5 Halaman Selesai Pengaduan	52
4.2.6 Halaman <i>Login</i> Admin	53
4.2.7 Halaman Utama Admin	53
4.2.8 Halaman Petugas.....	54
4.2.9 Halaman Mahasiswa	54
4.2.10 Halaman Fakultas	55
4.4.11 Halaman Kategori	56
4.4.12 Halaman Pengaduan	56
4.4.13 Halaman <i>Blog</i>	57
4.4.14 Halaman <i>Setting Slide</i>	57
4.4.15 Halaman <i>Setting</i> Tahun.....	58
4.4.16 Halaman Laporan Pengaduan	59

4.3 Pengujian	60
4.4 Pembahasan	67
BAB V KESIMPULAN DAN SARAN.....	68
5.1 Kesimpulan.....	68
5.2 Saran	68
DAFTAR PUSTAKA	69
LAMPIRAN	71

DAFTAR GAMBAR

Gambar 2.1.	Metode <i>Waterfall</i>	16
Gambar 3.1.	<i>SDLC</i> Alur Penelitian.....	18
Gambar 3.2.	Arsitektur Penanganan Keluhan.....	21
Gambar 3.3.	Alur Teknik Pengumpulan Data.....	23
Gambar 3.4.	<i>Use Case</i> Diagram.....	25
Gambar 3.5.	<i>Activity</i> Diagram.....	26
Gambar 3.6.	<i>Class</i> Diagram.....	27
Gambar 3.7.	<i>ER</i> Diagram.....	29
Gambar 3.8.	Rancangan Halaman Utama Aplikasi.....	34
Gambar 3.9.	Rancangan Halaman <i>login</i>	35
Gambar 3.10.	Rancangan Halaman <i>blog</i>	36
Gambar 3.11.	Rancangan Halaman pengaduan.....	37
Gambar 3.12.	Rancangan Halaman selesai pengaduan.....	38
Gambar 3.13.	Rancangan Halaman admin.....	39
Gambar 3.14.	Rancangan Halaman petugas.....	40
Gambar 3.15.	Rancangan Halaman mahasiswa.....	41
Gambar 3.16.	Rancangan Halaman fakultas.....	42
Gambar 3.17.	Rancangan Halaman kategori.....	43
Gambar 3.18.	Rancangan Halaman pengaduan.....	44
Gambar 3.19.	Rancangan Halaman pengeditan <i>blog</i>	45
Gambar 3.20.	Rancangan Halaman tahun.....	46
Gambar 3.21.	Rancangan Halaman jumlah.....	47
Gambar 4.1.	<i>Database</i> Aplikasi.....	49
Gambar 4.2.	Halaman Utama Aplikasi.....	50
Gambar 4.3.	Halaman Artikel.....	50
Gambar 4.4.	Halaman <i>Login</i> Mahasiswa.....	51
Gambar 4.5.	Halaman Utama Pengaduan.....	52
Gambar 4.6.	Halaman Selesai Pengaduan.....	52
Gambar 4.7.	Halaman <i>Login</i> Admin.....	53
Gambar 4.8.	Halaman Utama Admin.....	53
Gambar 4.9.	Halaman Petugas.....	54

Gambar 4.10.	Halaman Mahasiswa	54
Gambar 4.11.	Halaman Fakultas	55
Gambar 4.12.	Halaman Prodi	55
Gambar 4.13.	Halaman Kategori	56
Gambar 4.14.	Halaman Pengaduan	56
Gambar 4.15.	Halaman <i>Blog</i>	57
Gambar 4.16.	Halaman <i>Setting Slide</i>	57
Gambar 4.17.	Halaman <i>Setting Tahun</i>	58
Gambar 4.18.	Halaman <i>Setting Jumlah Pengaduan</i>	58
Gambar 4.19.	Halaman Laporan Pengaduan	59

DAFTAR TABEL

Tabel 2.1.	Simbol-simbol dalam <i>Use Case</i>	12
Tabel 2.2.	Simbol-simbol dalam <i>Activity Diagram</i>	13
Tabel 2.3.	Simbol-simbol dalam <i>Entity Relationship Diagram</i>	14
Tabel 3.1.	Keterangan komponen arsitektur penanganan keluhan.....	21
Tabel 3.2.	Struktur tabel kategori.....	30
Tabel 3.3.	Struktur tabel petugas.....	31
Tabel 3.4.	Struktur tabel <i>group_akses</i>	31
Tabel 3.5.	Struktur tabel mahasiswa	31
Tabel 3.6.	Struktur tabel pengaduan.....	32
Tabel 3.7.	Struktur tabel fakultas	32
Tabel 3.8.	Struktur tabel prodi	32
Tabel 3.9.	Struktur tabel tahun	33
Tabel 3.10.	Struktur tabel <i>setting</i>	33
Tabel 4.1.	Hasil Pengujian <i>Website</i>	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengaduan atau penyampaian aspirasi dan keluhan mahasiswa merupakan hal penting pada sebuah institusi atau universitas, karena dengan adanya penyampaian aspirasi tersebut sebuah universitas dapat dengan mudah memperbaiki dan meningkatkan kualitasnya.

Pengaduan mahasiswa merupakan suatu bentuk partisipasi mahasiswa agar penyedia layanan universitas yang dapat menampung keluhan dari mahasiswa. Hal tersebut dilakukan dengan tujuan agar pihak terkait dapat memperhatikan apa yang menjadi kebutuhan mahasiswa sehingga tercipta pelayanan yang lebih baik.

Fakultas Teknik Universitas Muhammadiyah Yogyakarta merupakan salah satu fakultas yang terdiri dari ribuan mahasiswa, sehingga wajar saja jika Fakultas Teknik Muhammadiyah Yogyakarta mendapatkan banyak aspirasi atau keluhan tentang sarana atau prasarana yang ada. Saat ini, FT UMY menerima keluhan melalui forum diskusi atau mendengar dari mahasiswa secara langsung. Forum diskusi tersebut dinamakan acara *public hearing*. Acara ini diselenggarakan secara berkala oleh himpunan mahasiswa. Dalam acara tersebut, mahasiswa dapat menyampaikan aspirasi, keluhan, masukan atau kritikan kepada pimpinan program studi atau fakultas. Acara tersebut bersifat terbuka seluruh mahasiswa dan dosen diharapkan hadir.

Acara *public hearing* tersebut sudah cukup baik, karena secara psikologis mahasiswa merasa diperhatikan. Keluhan mahasiswa bisa langsung ditanggapi, sehingga dapat mengurangi salah paham. Pimpinan segera tahu masalah-masalah yang dialami oleh mahasiswa. Meskipun demikian, acara tersebut memiliki keterbatasan, antara lain:

1. Waktunya terbatas, sehingga kemungkinan besar banyak mahasiswa yang ingin menyampaikan tidak diberi kesempatan untuk menyampaikan.

2. Ada sebagian mahasiswa yang merasa kurang nyaman jika berbicara di depan forum sehingga rasa penasaran mereka tidak tersampaikan. Hal ini dapat memicu persoalan baru di kalangan mahasiswa karena mahasiswa tersebut mengambil cara-cara yang kurang bijaksana atau tidak wajar.
3. Catatan kurang sempurna karena orang yang mencatat memiliki keterbatasan mengingat dan mencatat, sehingga keluhan yang disampaikan dan tanggapan yang diberikan tidak tercatat dengan sempurna. Ada kemungkinan beberapa informasi penting yang tidak tercatat.

Kritikan, keluhan, masukan, atau saran yang disampaikan bisa ditanggapi dengan penjelasan atau ditampung karena membutuhkan banyak langkah untuk menyelesaikan. Jika hal-hal yang disampaikan mahasiswa harus ditampung terlebih dahulu perlu adanya sistem yang sesuai. Sistem ini harus mampu menampung kritikan, keluhan, masukan, atau saran dari mahasiswa secara mudah. Selain itu, sistem ini harus mudah digunakan, mampu mencatat keluhan, mampu menampilkan kemajuan atau tindakan apa yang telah dilakukan oleh program studi. Sistem juga harus dapat menyampaikan informasi dalam bentuk laporan untuk pengelola program studi.

1.1.1 Rumusan Masalah

Meninjau pokok permasalahan yang telah diuraikan pada latar belakang di atas maka dapat diambil rumusan masalah sebagai berikut:

1. Pengaduan melalui *public hearing* waktunya terbatas.
2. Belum semua pengaduan tertampung dengan baik.
3. Pencatatan pengaduan atau aspirasi mahasiswa masih manual, sehingga kemungkinan besar ada informasi yang hilang.

1.2 Tujuan Penelitian

Tujuan penelitian diantaranya adalah sebagai berikut:

1. Membuat sebuah program aplikasi yang mampu menampung pengaduan secara *online*, mencatat kemajuan penanganan, mencatat dan menampilkan tanggapan, serta memudahkan administrator dalam membuat laporan.

1.3 Manfaat Penelitian

Jika sistem mampu memperbaiki cara-cara penyampaian keluhan, kritikan, masukan, dan saran dari mahasiswa maka manfaat bagi program studi FT UMY adalah:

1. Mahasiswa merasa diperhatikan sehingga masalah-masalah yang tidak ada hubungannya dengan kegiatan akademik dapat ditekan.
2. Hubungan harmonis antara mahasiswa dengan sesama mahasiswa, dosen, karyawan dan pimpinan dapat ditingkatkan.
3. Kehidupan akademik lebih kondusif sehingga motivasi belajar mahasiswa semakin meningkat.

1.4 Sistematika Penulisan

Sistematika laporan penelitian disusun dalam lima bab yang membahas beberapa hal sebagai berikut:

BAB I: PENDAHULUAN

Bab pendahuluan berisi penjelasan mengenai pendahuluan dari penelitian yang meliputi latar belakang permasalahan, rumusan masalah, tujuan, maksud dan sistematika penulisan yang digunakan pmbutan tugas akhir.

BAB II: TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab tinjauan pustaka dan landasan teori berisi penjelasan tentang kajian pustaka dan teori-teori penunjang yang digunakan sabagai dalam penelitian, yaitu konsep dan teknologi dalam pangkalan data dan segala sesuatu yang berhubungan dengan topik penelitian ini seperti hal-hal yang

berkaitan dengan metadata dalam pangkalan data.

BAB III: METODOLOGI PENELITIAN

Bab metodologi penelitian berisi penjelasan metode dan *tools* yang digunakan untuk merancang sistem dapat diimplementasikan dalam sistem yang sesuai harapan, mengacu pada teori-teori penunjang yang sudah dijelaskan pada bab II.

BAB IV: HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan berisi penjelasan mengenai hasil yang diperoleh dari seluruh penelitian dan dilakukan pengujian terhadap hasil implementasi sistem kemudian menganalisa kedalam sistem sesuai dengan penrancangan pada bab-bab sebelumnya.

BAB V: KESIMPULAN DAN SARAN

Bab kesimpulan dan saran merupakan penutup yang berisi kesimpulan dan saran yang diambil dari penelitian dan saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada tinjauan pustaka peneliti akan memaparkan tinjauan peneliti atas beberapa penelitian dan kajian ilmiah terdahulu serta beberapa konsep yang memiliki keterkaitan dengan penelitian yang akan dilakukan peneliti. Tinjauan pustaka pertama yaitu penelitian yang dilakukan oleh **Sad Dian Utomo** di tahun 2008 yang berjudul **Penanganan Pengaduan Masyarakat Mengenai Pelayanan Publik (Studi Pada Pusat Penanganan Pengaduan Pelayanan Publik Kota Semarang)**. Penelitian menggunakan pendekatan kualitatif dengan metode deskriptif dimana Sad Dian Utomo menitik beratkan penelitian pada bagaimana proses pengaduan dilaksanakan dan bagaimana keterlibatan masyarakat didalamnya. Banyaknya keluhan melalui kotak saran atau saluran lainnya belum banyak di tanggapi oleh penyedia pelayanan publik di Kota Semarang. Maka dari itu Pemerintah Kota Semarang membentuk unit penanganan pengaduan yang disebut Pusat Penanganan Pengaduan Pelayanan Publik Kota Semarang atau disingkat P5 sebagai instrumen partisipasi masyarakat yang diandalkan dan dapat digunakan untuk memperbaiki pelayanan publik.

Hasil penelitian yang dilakukan Sad Dian Utomo memaparkan bahwa sebagian besar *stake holders* memandang positif dan menilai P5 cukup efektif. Sementara itu dengan membandingkan instrumen partisipasi itu dengan teori *ladder of citizen participation* dari Arnstein dan teori *ladder of citizen empowerment* dari *Burn, Hambleton & Hogget* dapat disimpulkan bahwa tingkat partisipasi warga masyarakat belum mencapai titik ideal yaitu kontrol masyarakat. Pencapaian tingkat partisipasi dan efektivitas P5 itu dipengaruhi oleh faktor-faktor seperti kepemimpinan, regulasi atau peraturan, kewenangan P5, peran *civil society*, informasi dan momentum partisipasi. Kelebihan dari penelitian ini adalah Sad menggunakan pendekatan kualitatif dalam memaparkan suatu gejala sosial dimana partisipasi masyarakat dibutuhkan

dalam upaya memperbaiki kualitas pelayanan publik. Karena suatu gejala sosial sulit untuk diukur dengan pendekatan kuantitatif. Adapun kekurangannya adalah pemaparan teori yang kurang proporsional dimana lebih banyak teori partisipasi masyarakat sedangkan teori pengaduan masih belum banyak dideskripsikan.

Tinjauan pustaka yang kedua merupakan penelitian yang dilakukan oleh Panji Tri Nugroho di tahun 2010 yang berjudul Efektivitas Organisasi Ombudsman Republik Indonesia. Penelitian ini membahas efektivitas Ombudsman RI dilihat dari perspektif organisasi. Ombudsman merupakan lembaga yang bertugas untuk mengawasi pelaksanaan pelayanan publik. Selain itu lembaga ini bertugas menangani pengaduan masyarakat. Meskipun Ombudsman RI telah berdiri sejak tahun 2000, namun ternyata masih banyak masyarakat yang belum mengetahui lembaga ini. Hal inilah yang melatarbelakangi Panji untuk melakukan penelitian.

Dalam penelitiannya, Panji menggunakan pendekatan *positivis* dengan metode pengumpulan data dengan cara wawancara mendalam, studi literatur, dan kuesioner. Penelitian ini menggunakan pendekatan efektivitas organisasi dengan pendekatan sistem yang terdiri dari *input, process, output*, dan *outcome*. Tujuan dari penelitian ini adalah untuk mengetahui efektivitas Lembaga Ombudsman RI dan kendala-kendala yang dihadapi dalam mencapai efektivitas tersebut. Hasil penelitian yang dilakukan Panji menunjukkan bahwa Ombudsman RI belum sepenuhnya efektif karena masih terdapat sejumlah hambatan mulai dari tahap *input, proses, output*, hingga *outcome*.

Persamaan antara penelitian yang dilakukan oleh Panji dengan penelitian yang dilakukan oleh peneliti adalah sama-sama meneliti mengenai pengaduan masyarakat. Adapun perbedaannya adalah pemilihan site penelitian, Perbedaan lainnya adalah teori yang digunakan untuk mengukur efektivitas pengaduan. Kelebihan penelitian ini memberikan gambaran efektivitas organisasi Ombudsman Indonesia dari pendekatan sistem, dimana Panji meneliti keseluruhan efektivitas organisasi Ombudsman RI. Sedangkan kekurangannya adalah kurang dapat menjangkau lebih banyak pelapor dan terlapor dalam mengukur efektivitas pengaduan.

Tinjauan pustaka yang ketiga peneliti lakukan atas penelitian yang dilakukan oleh Suryadi di tahun 2010 yang berjudul Penanganan Keluhan Publik pada Birokrasi Dinas Perijinan Kota Malang. Penelitian ini dilakukan dengan pendekatan kualitatif dengan fokus penelitian untuk memberikan gambaran keluhan publik atas pelayanan perijinan birokrasi Dinas Perijinan Kota Malang, media penyaluran keluhan, dan penanganan atas keluhan publik yang dilakukan birokrasi Dinas Perijinan Kota Malang.

Hasil penelitian menunjukkan bahwa masih terdapat banyak masalah dalam penanganan keluhan publik di Dinas Perijinan Kota Malang seperti rumitnya persyaratan, yang implikasinya memperpanjang waktu yang dibutuhkan atau mahal biaya yang dibutuhkan. Adapun kekurangan dari penelitian ini adalah kekurangan data-data pendukung yang menggambarkan permasalahan. Sedangkan kelebihan adalah penelitian ini banyak mencantumkan kutipan wawancara dari masyarakat dan petugas Dinas Perijinan Kota Malang. Yang membedakan penelitian ini dengan penelitian yang dilakukan peneliti adalah pendekatan dan teori yang digunakan dalam menganalisis permasalahan.

Dalam penelitian yang berjudul Pengembangan Sistem Penanganan Keluhan Berbasis *Web* (Studi Kasus: Fakultas Teknik Universitas Muhammadiyah Yogyakarta), ini dibangun untuk digunakan dalam proses aspirasi atau keluhan pada Fakultas Teknik Universitas Muhammadiyah Yogyakarta, tujuannya adalah untuk memberikan kemudahan dalam proses pengolahan data aspirasi atau keluhan untuk petugas. Sistem dibangun menggunakan *PHP bootstrap* yang memudahkan *developer* dan *designer* untuk membuat sebuah aplikasi *web* menjadi cepat dan mudah dibandingkan dengan membuat dari awal. Keunggulannya adalah karena dukungan akan perpaduan *framework css* yang dapat meringkas pekerjaan dalam membuat sebuah *template. mysql* digunakan untuk pengolahan *database*. Sistem digunakan untuk mengolah data aspirasi atau keluhan mahasiswa. Mahasiswa bisa mengungkapkan aspirasi atau keluhan melalui *web*, selain itu mahasiswa juga dapat menentukan kepada siapa aspirasi dan keluhan tersebut ditujukan. Mahasiswa akan

mendapatkan akses *Id* dan *Password* yang telah ditetapkan oleh Fakultas Teknik Universitas Muhammadiyah Yogyakarta.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Menurut (Fat, 2012) pengertian sistem adalah sebagai berikut: “Sistem adalah suatu benda nyata atau abstrak (*a set of thing*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, ketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan *Unity* untuk mencapai tujuan tertentu secara efisien dan efektif”.

Pengertian Sistem Menurut (Murdick, 1991) Suatu sistem adalah seperangkat elemen yang membentuk kumpulan atau prosedur-prosedur pengolahan yang mencari suatu tujuan bagian atau tujuan bersama dengan mengoperasikan data atau barang pada waktu rujukan tertentu untuk menghasilkan informasi energi atau barang.

Definisi Sistem Menurut (Lani, 1995), “Sistem adalah himpunan dari bagian-bagian yang saling berhubungan yang secara keseluruhan merupakan suatu kesatuan secara fungsional”.

Sistem informasi adalah kerangka kerja yang mengkoordinir sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (*output*) berupa informasi guna mencapai sasaran. Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan atau *building block* (Jogiyanto, 2005).

Dengan demikian sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai suatu tujuan dari sistem tersebut. Maksud dari suatu sistem adalah untuk mencapai suatu tujuan dan sasaran dalam ruang lingkup yang sempit.

Menurut (Jogiyanto, 2005) Sistem merupakan “bagian yang saling berkaitan erat dan membentuk suatu kesatuan yang saling berinteraksi antara bagian satu dengan bagian lainnya untuk mencapai suatu tujuan”. Dari sumber

yang berbeda mendefinisikan sistem sebagai “Himpunan sesuatu benda nyata atau *abstrak (asset of things)* yang terdiri dari bagian-bagian atau komponen-komponen yang saling berhubungan, dan saling mendukung, yang secara keseluruhan bersatu dalam kesatuan (*unity*) untuk mencapai tujuan secara efisien dan efektif”. (Drs. Zulkifli Amsyah, 2000)

Menurut (Jogiyanto, 2005) Pada hakekatnya suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

1. Komponen sistem (*component*) Komponen, sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.
2. Batasan sistem (*boundary*), Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.
3. Lingkungan luar sistem (*environment*), Lingkungan luar dari sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.
4. Penghubung sistem (*interface*), Penghubung sistem merupakan media penghubung atau subsistem dengan subsistem lainnya. Dengan subsistem dapat berintegrasi dengan subsistem lainnya membentuk satu kesatuan.
5. Masukan sistem (*input*), Masukan sistem dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*sinyal input*). *Maintenance input* berupa sebuah program komputer, pada komputer data merupakan sinyal *input* untuk diolah menjadi informasi.
6. Pengolahan sistem (*process*), Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah *input* menjadi *output*.

7. Keluaran sistem (*output*), Keluaran sistem merupakan hasil dari pengolahan sistem dan mengklasifikasikan masukan menjadi keluaran.
8. Sasaran sistem (*objectives*), Suatu sistem pasti mempunyai tujuan atau sasaran. Sasaran sistem sangat menentukan sekali masukan yang dibutuhkan dan keluaran yang akan dikatakan berhasil bila mengenai sasaran dan tujuannya.

Sistem informasi didefinisikan oleh (Roscoe, 2013), sebagai “suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Sehingga dapat disimpulkan sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya manusia (manusia dan komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi) guna mencapai sasaran-sasaran.

2.2.2 Pengertian Penanganan Keluhan

Keluhan adalah salah satu bagian dari ekspresi negatif yang dihasilkan karena ke tidak sesuai kenyataan dengan keinginan seseorang (James, 2006) Keluhan adalah satu pernyataan atau ungkapan rasa kurang puas terhadap satu produk atau layanan, baik secara lisan maupun tertulis, dari pelanggan internal maupun external.

Manfaat prosedur penanganan keluhan antara lain:

- a. Tersedia prosedur yang jelas ketika terjadi keluhan.
- b. Menciptakan pemahaman dan keyakinan cara menangani keluhan.
- c. Membantu mengatasi rasa “bersalah” secara pribadi bagi orang yang menangani keluhan.
- d. Menerima keluhan sebagai umpan balik yang berharga, bukan sebagai kritik.
- e. Menghasilkan catatan yang dapat digunakan untuk menganalisa kemungkinan peningkatan layanan.

Secara definisi Keluhan diartikan sebagai satu pernyataan atau ungkapan rasa kurang puas terhadap satu produk atau layanan jasa, baik secara lisan maupun tertulis, dari penyampaian keluhan baik internal maupun eksternal. Atau sebuah ungkapan ke tidak puas antara harapan dengan fakta terhadap apa yang diterima dalam bentuk produk maupun layanan jasa.

Adanya keluhan dalam satu sisi merupakan alat kendali atau evaluasi terhadap pemberian kualitas pelayanan yang selama ini diberikan kepada masyarakat. Namun pada sisi lain adalah suatu hal yang perlu diperhatikan, yang menjadikan keluhan sebagai suatu masalah yang perlu dicari solusinya. Penanganan terhadap munculnya keluhan harus ditanggapi secara rasional dan sebaiknya faktor emosional. Sesungguhnya apabila terjadi keluhan sulit untuk membedakan sifat dari yang dikeluhkan yang biasanya keluhan berupa masalah yang serius. (Rekompak, n.d.)

2.2.3 Teknologi Pengembangan Aplikasi







2.2.3.1 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak. Penggunaan model *UML* bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model *UML* yang dipakai dalam pengembangan aplikasi Penanganan Keluhan *Berbasis Web* antara lain adalah *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *ER Diagram*. (Sholih). Bagian dari UML antara lain:

a. *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi yang dibutuhkan. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.1.




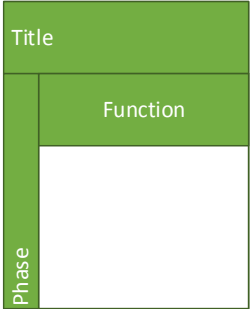
Tabel 2.1. Simbol-simbol dalam *Use Case*

No	Gambar	Nama Gambar	Keterangan
1		<i>Use Case</i>	Merupakan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan dengan <i>actor</i> .
2		<i>Actor</i>	Merupakan <i>abstraction</i> dari orang yang mengaktifkan fungsi dari target sistem dan merupakan orang yang berinteraksi dengan <i>use case</i> .
3		<i>Association</i>	Digambarkan dengan garis tanpa panah yang mengindikasikan siapa yang berinteraksi secara langsung dengan sistem.
4		<i>Generalization</i>	Mengindikasikan siapa yang berinteraksi secara pasif dengan sistem.
5		<i>Include</i>	Mengidentifikasi hubungan antar dua <i>use case</i> dimana satu <i>usecase</i> memanggil <i>usecase</i> yang lain.
6		<i>Extend</i>	Merupakan perluasan dari <i>use case</i> jika kondisi atau syarat terpenuhi.

a. *Activity Diagram*

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2. Simbol-simbol dalam *Activity Diagram*

No	Gambar	Nama Gambar	Keterangan
1		<i>Start Point</i>	Merupakan awal dalam aktifitas.
2		<i>End Point</i>	Merupakan akhir dalam aktifitas.
3		<i>Activities</i>	Menggambarkan suatu proses atau kegiatan bisnis
4		<i>Decision Point</i>	Menggambarkan pilihan untuk pengambilan keputusan dalam aktifitas.
5		<i>Swimlane</i>	Digunakan untuk pembagian <i>activity diagram</i> yang menunjukkan siapa yang melakukan aktifitas.

b. *Class Diagram*





Class Diagram merupakan diagram yang digunakan untuk menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang

berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Nama Kelas (*Class Name*), Atribut (*Attributes*), Operasi (*Operations*), dan Relasi (*Relationships*).

c. *Entity Relationship Diagram (ERD)*

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* digunakan untuk memodelkan struktur data dan hubungan antar data (expresiku, n.d.), untuk menggambarkannya digunakan beberapa notasi dan simbol seperti pada table 2.3.

Tabel 2.3. Simbol-simbol dalam *Entity Relationship Diagram*

No	Gambar	Nama Gambar	Keterangan
1		<i>Entitas</i>	Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
2		<i>Relasi</i>	Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
3		<i>Atribut</i>	Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
4		<i>Association</i>	Sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

2.2.4 Pengembangan Sistem Berbasis Web

Pengembangan sistem berbasis *web* adalah aplikasi yang sejak awal dirancang untuk dieksekusi di lingkungan berbasis *web*. Definisi ini mengungkapkan dua aspek penting dari aplikasi ini (Simarmata, 2009) sebagai berikut:

1. Suatu aplikasi *web* dirancang agar dapat berjalan di dalam lingkungan berbasis *web*. Artinya, aspek-aspek *hipermedia* dalam kaitannya dengan *hiperteks* dan multimedia di dalam kombinasi dengan kelola aplikasi tradisional harus diperhitungkan di seluruh hidup aplikasi.
2. Aplikasi *web* adalah suatu aplikasi yang tidak hanya berupa sekumpulan halaman-halaman *web*.

2.2.5 PHP

Hypertext Preprocessor (PHP) adalah *script* pemrograman yang terletak dan dieksekusi di *server*. Salah satunya adalah untuk menerima, mengelola, dan menampilkan data dari dan ke seluruh situs.

Data akan diolah ke sebuah *database server* (pemrograman *database* yang terletak di sisi *server*) untuk memudahkan hasilnya ditampilkan di *browser* sebuah situs (Madcoms, 2011).

PHP adalah pemrograman yang digunakan untuk membuat *software* yang merupakan bagian dari sebuah situs *web*. *PHP* dirancang untuk berbaur dengan HTML yang digunakan untuk membuat halaman *web* (Sudarmo, 2006).

2.2.6 MySQL

MySQL adalah salah satu program yang dapat digunakan sebagai *database*, dan merupakan salah satu *software* untuk *database server* yang banyak digunakan. *MySQL* bersifat *open source* dan menggunakan *SQL*. *MySQL* bisa dijalankan diberbagai *platform*, *windows* dan *linuk* (Madcoms, 2011).

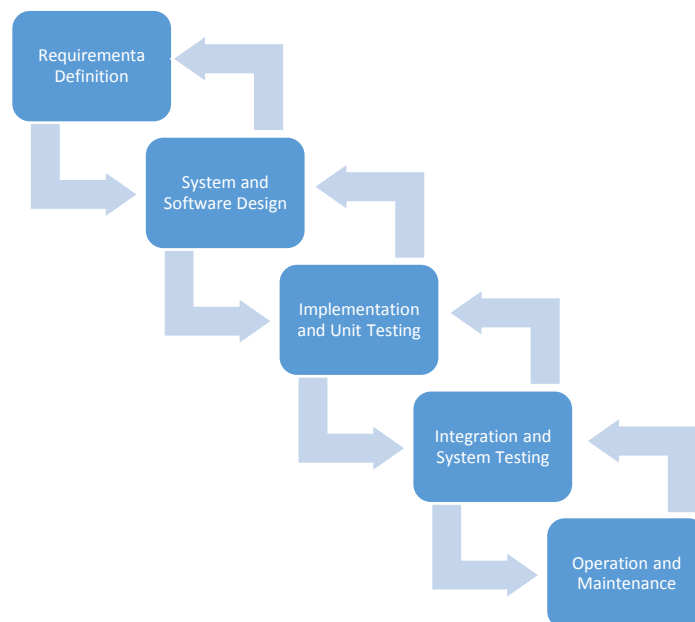
MySQL adalah *software* sistem manajemen *database*. *Database* adalah suatu koleksi data yang terstruktur. *Database* bisa berupa daftar belanja sederhana sampai informasi yang sangat besar dari suatu perusahaan internasional. Untuk menambahkan, mengakses, dan memproses data disimpan di komputer (Rickyanto, 2002).

2.2.7 Software Development Life Cycle

Dalam alur penelitian, metode yang digunakan adalah model SDLC (*Software Development Life Cycle*). SDLC adalah suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap

dalam pembuatan sebuah *software* (Fatta, 2007). Selain itu, SDLC juga penting untuk proses *maintenance software* itu sendiri.

Model SDLC yang dipakai dalam pengembangan aplikasi adalah model *Waterfall*. Menurut (Sommerville, 2011), *waterfall model* adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 2.1. Metode *Waterfall*

Terdapat 5 tahapan metode *Waterfall* yang dapat dilihat pada gambar 2.1 di mulai dari analisis dan definisi kebutuhan hingga operasi dan pemeliharaan. Berikut penjelasan dari tahapan metode *Waterfall*:

1. Analisis dan definisi kebutuhan. Layanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user*.
2. Perancangan sistem dan perangkat lunak. Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan.

Perancangan melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar.

3. Implementasi dan pengujian unit. Pada tahap ini, perancangan perangkat lunak direalisasikan dengan program atau unit program. Pengujian ini melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.
4. Integrasi dan pengujian sistem. Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah dipenuhi.
5. Operasi dan pemeliharaan, yaitu mengoperasikan program di lingkungannya dan melakukan pemeliharaan. Biasanya ini merupakan fase siklus hidup yang paling lama. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan pada tahap-tahap sebelumnya, melakukan perbaikan atas implementasi unit sistem dan pengembangan layanan sistem, dan persyaratan-persyaratan baru ditambahkan.

2.2.8 Black Box Testing

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, yang digunakan pada skripsi adalah *black box testing* dengan alasan pembuat hanya menguji apakah fungsionalitas dalam aplikasi sudah sesuai dengan yang diharapkan.

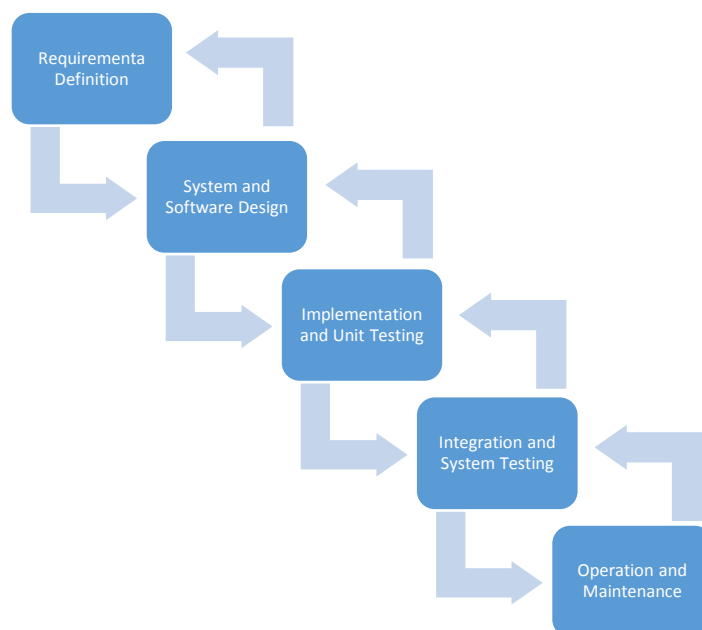
Menurut (Roger S, 2002), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.

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*.

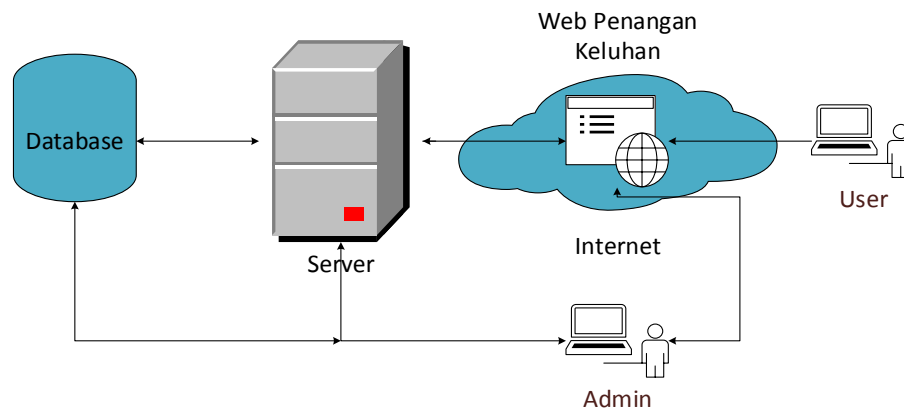
d. *Bootstrap*

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

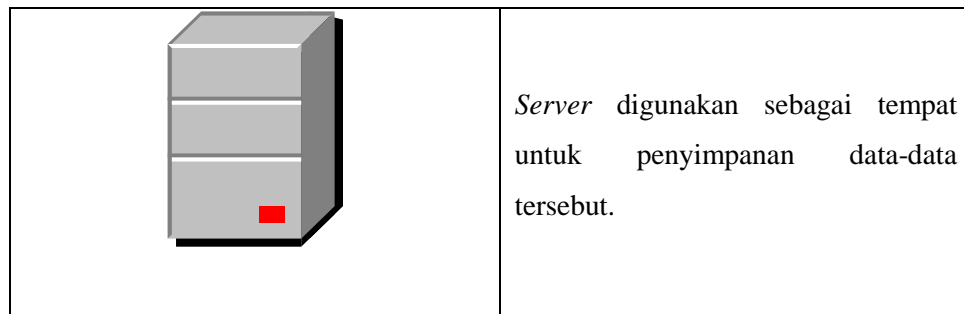


Gambar 3.2. Arsitektur Penanganan Keluhan

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 kuisioner 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

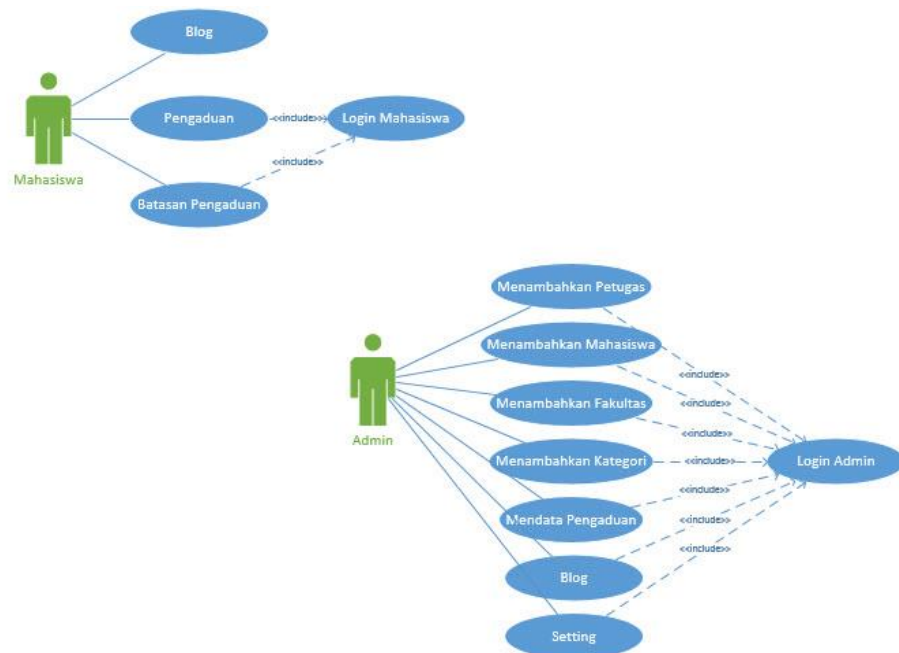
Sistem akan diuji sesuai dengan *requirement* yang telah ditentukan sebelumnya. Seluruh *requirement* harus terpenuhi dan tidak ada yang tertinggal ataupun tidak sesuai dengan *requirement*. Pengujian akan dilakukan dengan menggunakan metode *balckbox*.

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.



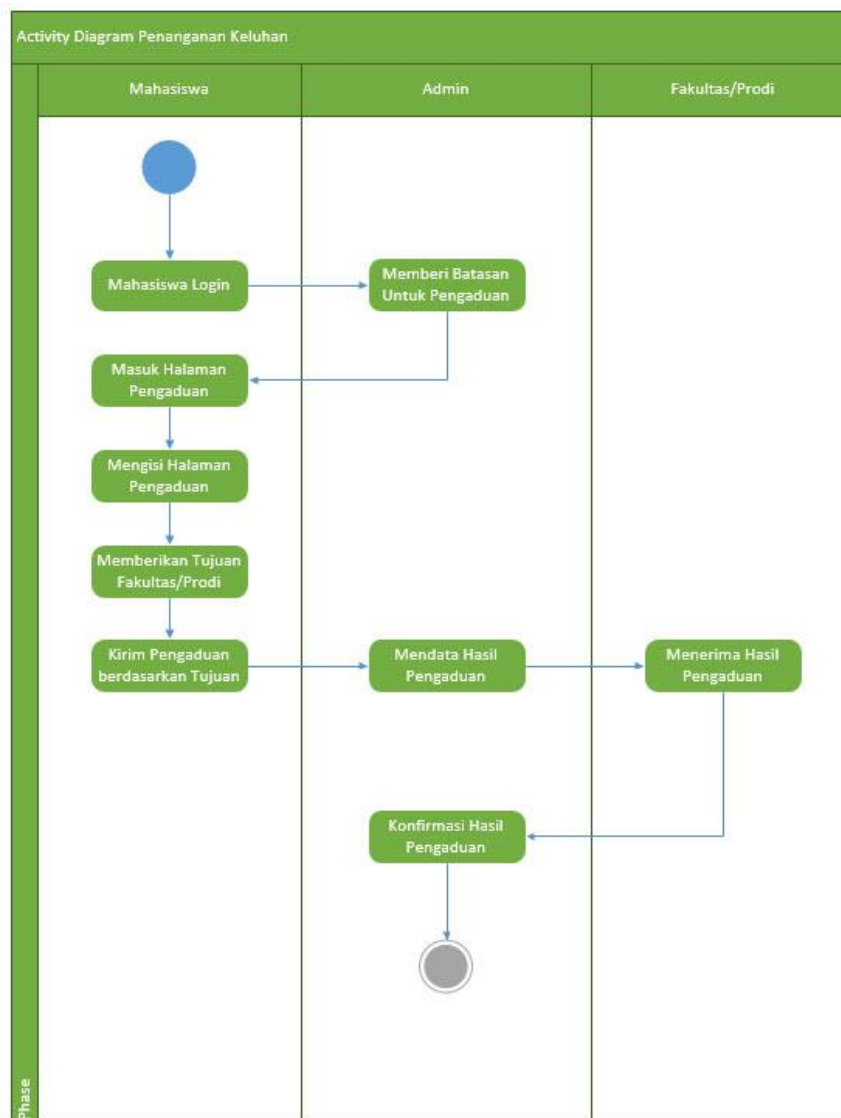
Gambar 3.4. Use Case Diagram

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

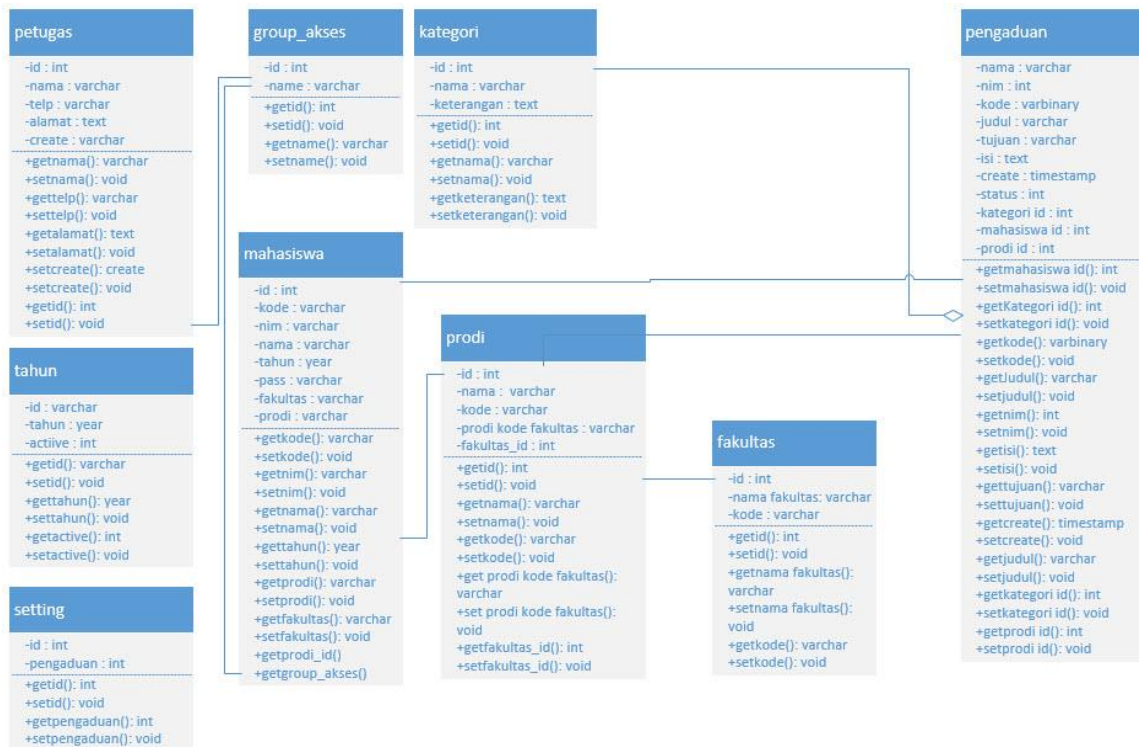
Gambar 3.5 menunjukkan *Activity Diagram* pada kegiatan *web* penanganan keluhan. Mahasiswa melakukan *login*, setelah itu admin memberikan batasan untuk pengaduannya, mahasiswa masuk ke halaman pengaduan, kemudian mahasiswa mengisi halaman pengaduan, mahasiswa memilih tujuan untuk ke pengaduannya yaitu 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.

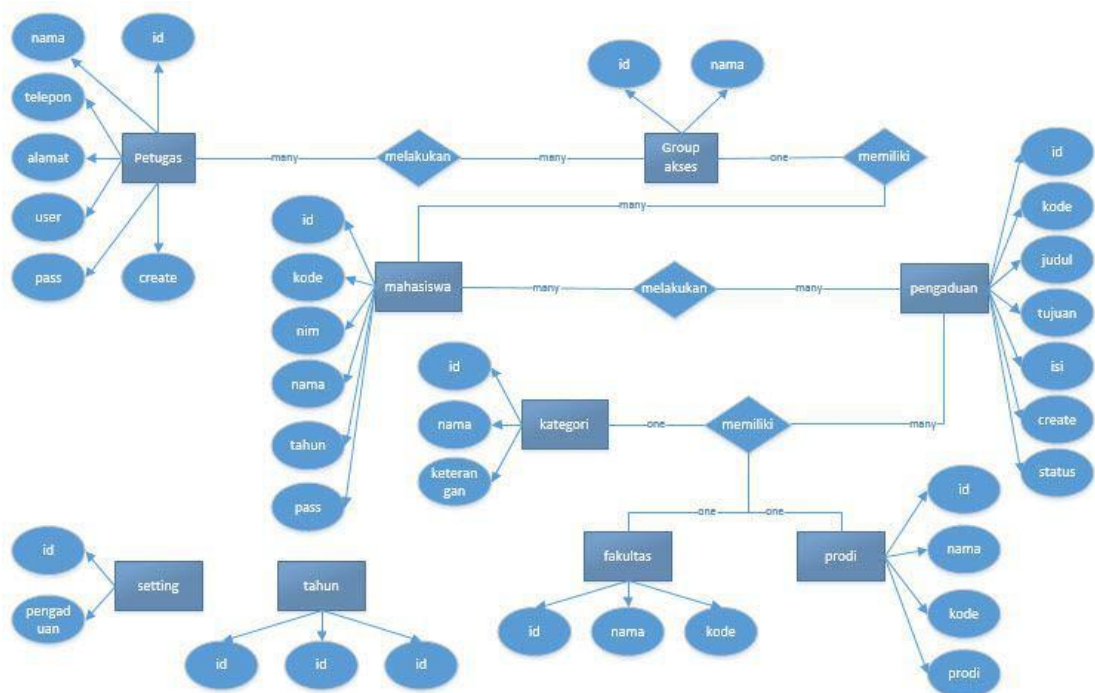
Class mahasiswa memiliki *association* dengan *class* pengaduan, *class* mahasiswa boleh melakukan akses pengaduan melalui *method* *Getmahasiswaid()*.

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:

- a. Petugas
- b. Group_akses
- c. Mahasiswa
- d. Pengaduan
- e. Kategori
- f. Fakultas
- g. 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*

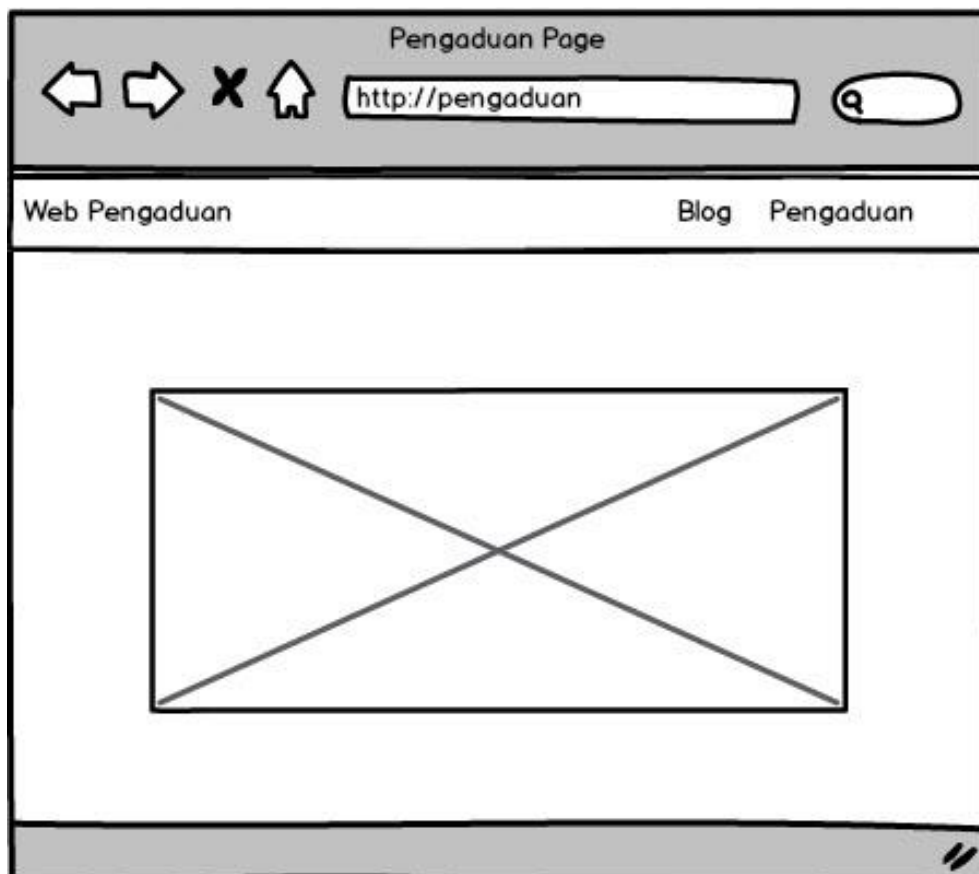
No	Field Name	Key Type	Null	Data Type	Max. Length
1	Set_id	PK	NN	Int	11
2	Set_pengaduan		NN	Int	11

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

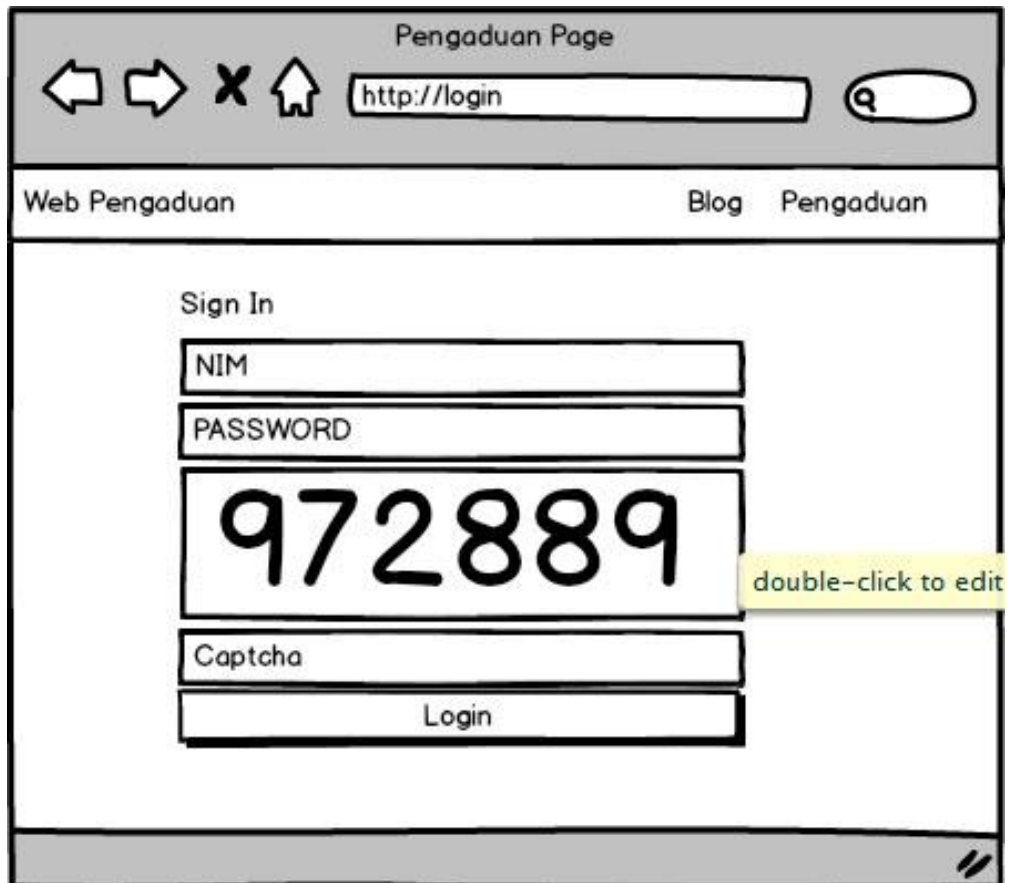
Rancangan antarmuka halaman utama merupakan tampilan utama dari aplikasi yang dapat dilihat oleh mahasiswa dan admin. Gambaran rancangan antarmuka halaman utama aplikasi dapat dilihat pada Gambar 3.8.



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.



Pengaduan Page

Web Pengaduan Blog Pengaduan

Sign In

NIM

PASSWORD

972889

Captcha

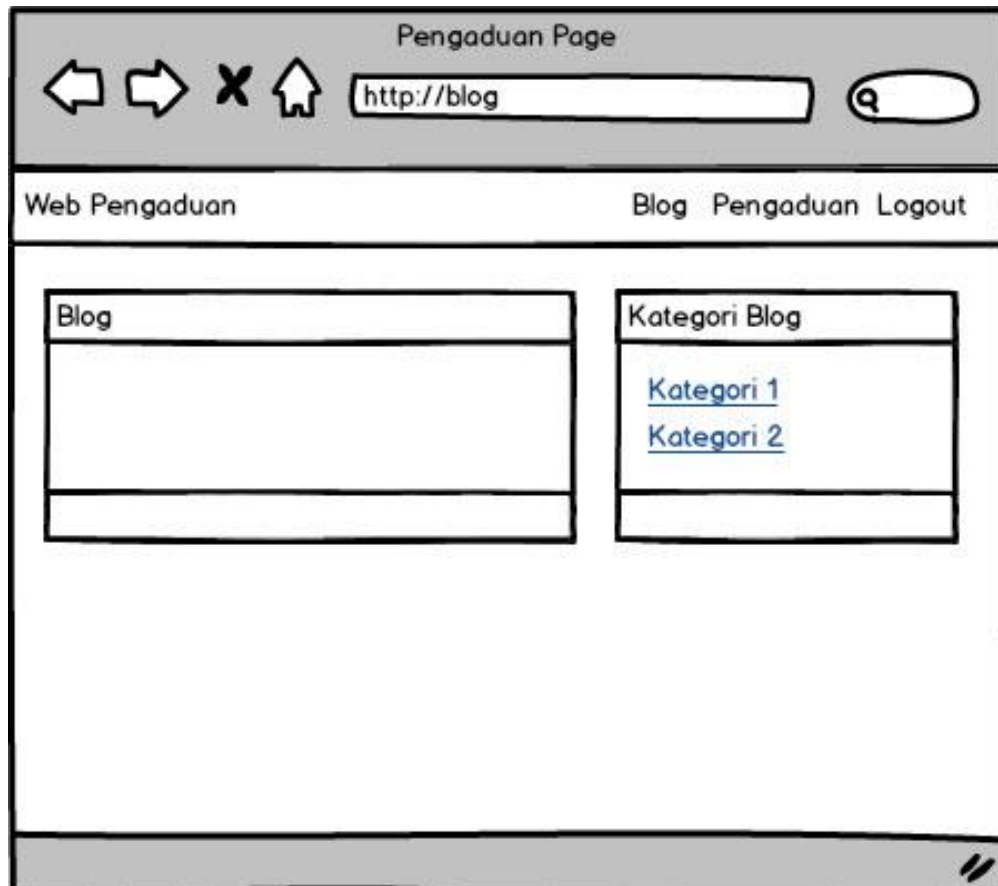
Login

double-click to edit

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.

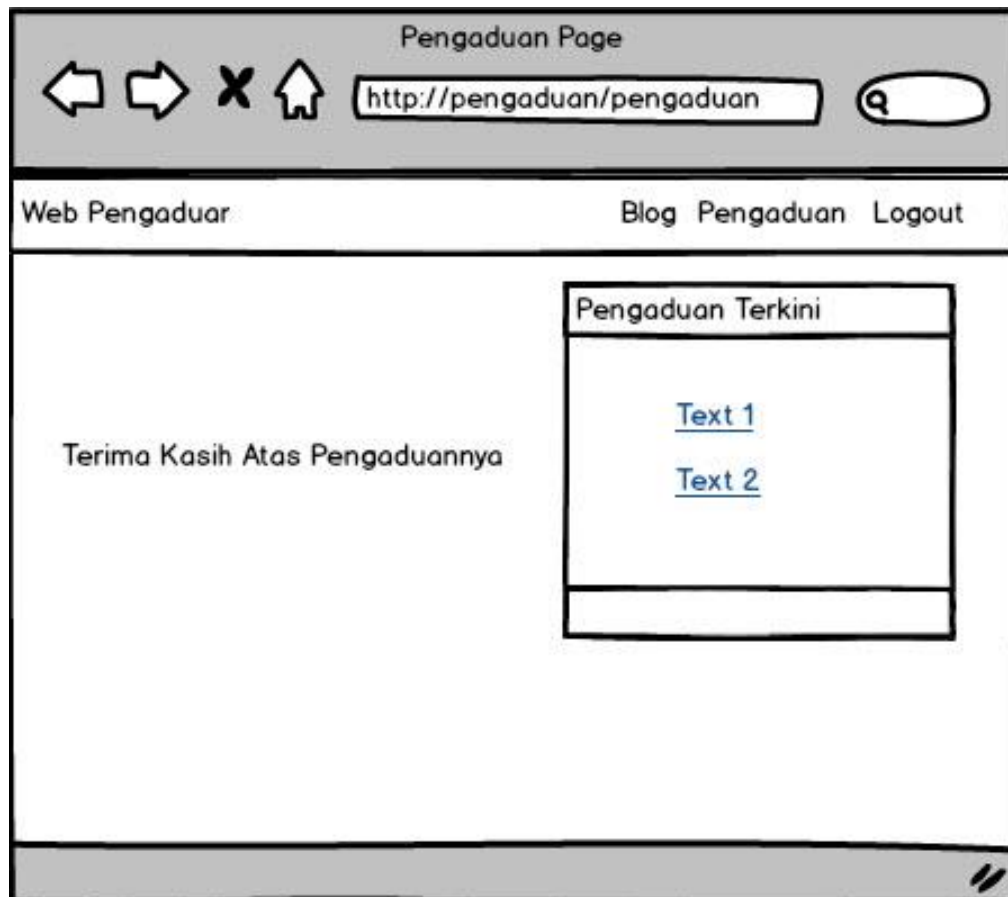
The image shows a web browser window titled "Pengaduan Page" with the URL "http://pengaduan/pengaduan". The page content is as follows:

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

Gambar 3.11. Rancangan Halaman pengaduan

3.7.5.5 Rancangan antarmuka halaman selesai pengaduan

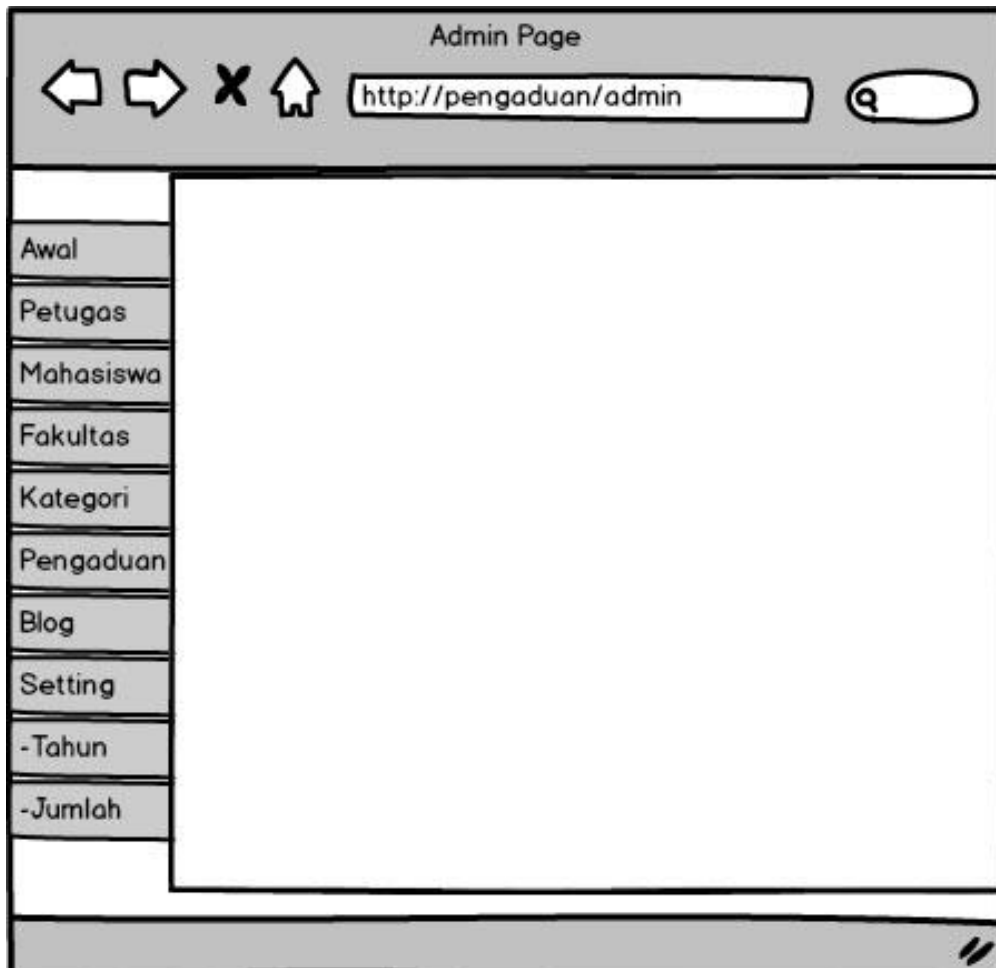
Rancangan antarmuka halaman selesai pengaduan merupakan halaman yang digunakan untuk memastikan bahwa mahasiswa sudah memberikan aspirasi atau keluhan. Gambaran rancangan antarmuka halaman selesai pengaduan dapat dilihat pada Gambar 3.12.



Gambar 3.12. Rancangan Halaman selesai pengaduan

3.7.5.6 Rancangan antarmuka halaman admin

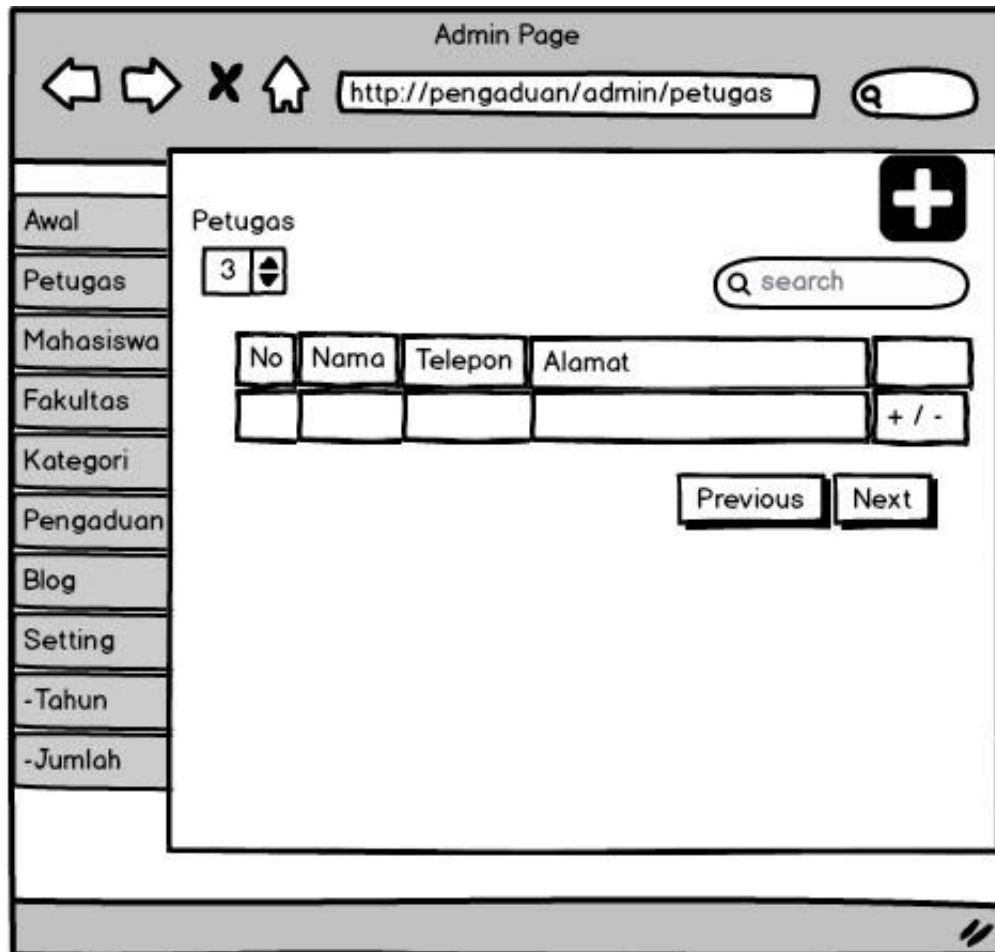
Rancangan antarmuka halaman utama admin merupakan tampilan utama dari aplikasi yang dapat dilihat oleh admin. Gambaran rancangan antarmuka halaman admin dapat dilihat pada Gambar 3.13.



Gambar 3.13. Rancangan Halaman admin

3.7.5.7 Rancangan antarmuka halaman petugas

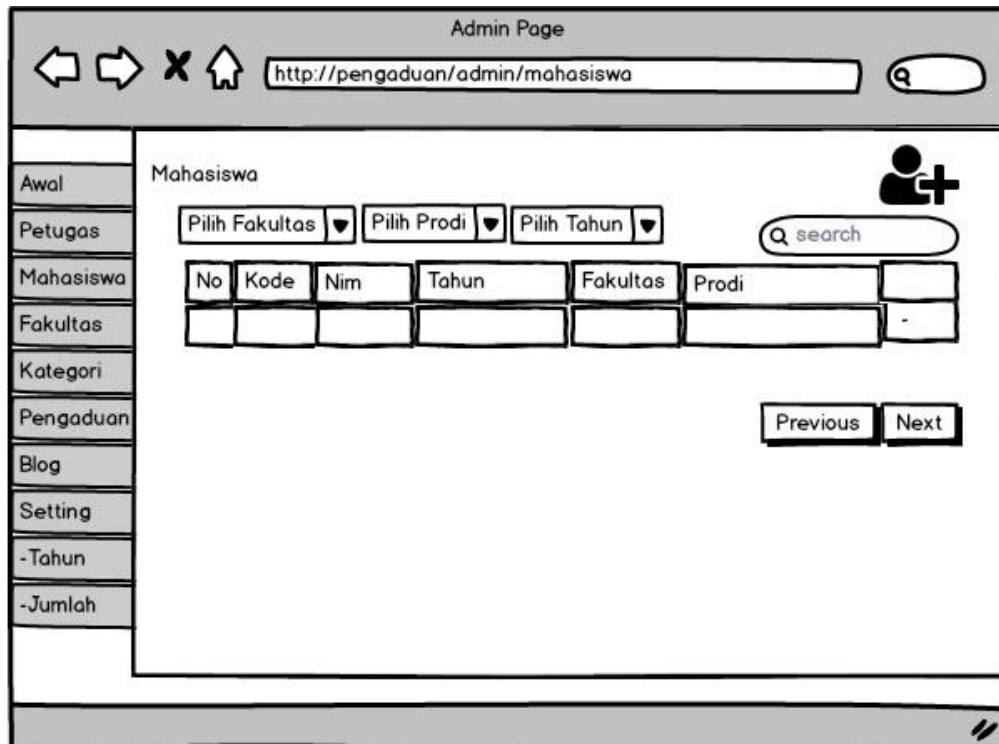
Rancangan antarmuka halaman petugas merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mencari petugas. Gambaran rancangan antarmuka halaman petugas dapat dilihat pada Gambar 3.14.



Gambar 3.14. Rancangan Halaman petugas

3.7.5.8 Rancangan antarmuka halaman mahasiswa

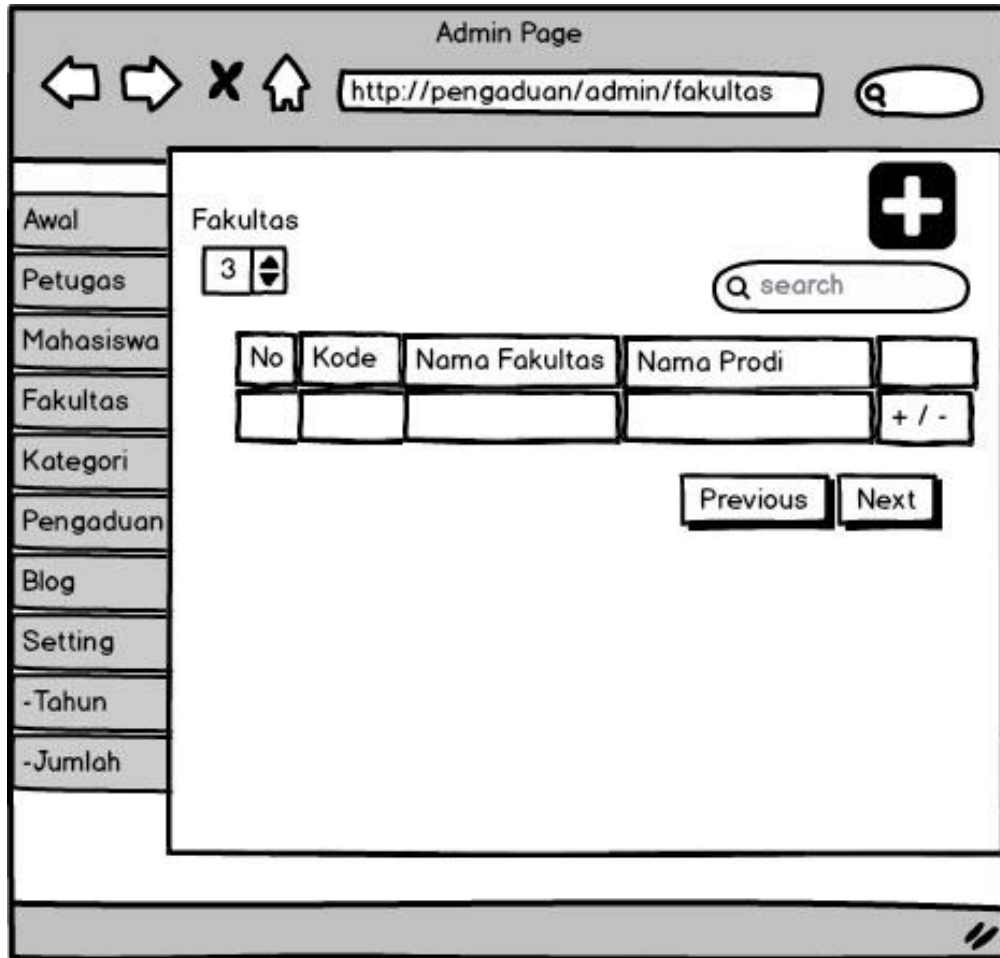
Rancangan antarmuka halaman mahasiswa merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mencari mahasiswa sesuai fakultas, prodi dan tahun angkatan. Gambaran rancangan antarmuka halaman mahasiswa dapat dilihat pada Gambar 3.15.



Gambar 3.15. Rancangan Halaman mahasiswa

3.7.5.9 Rancangan antarmuka halaman fakultas

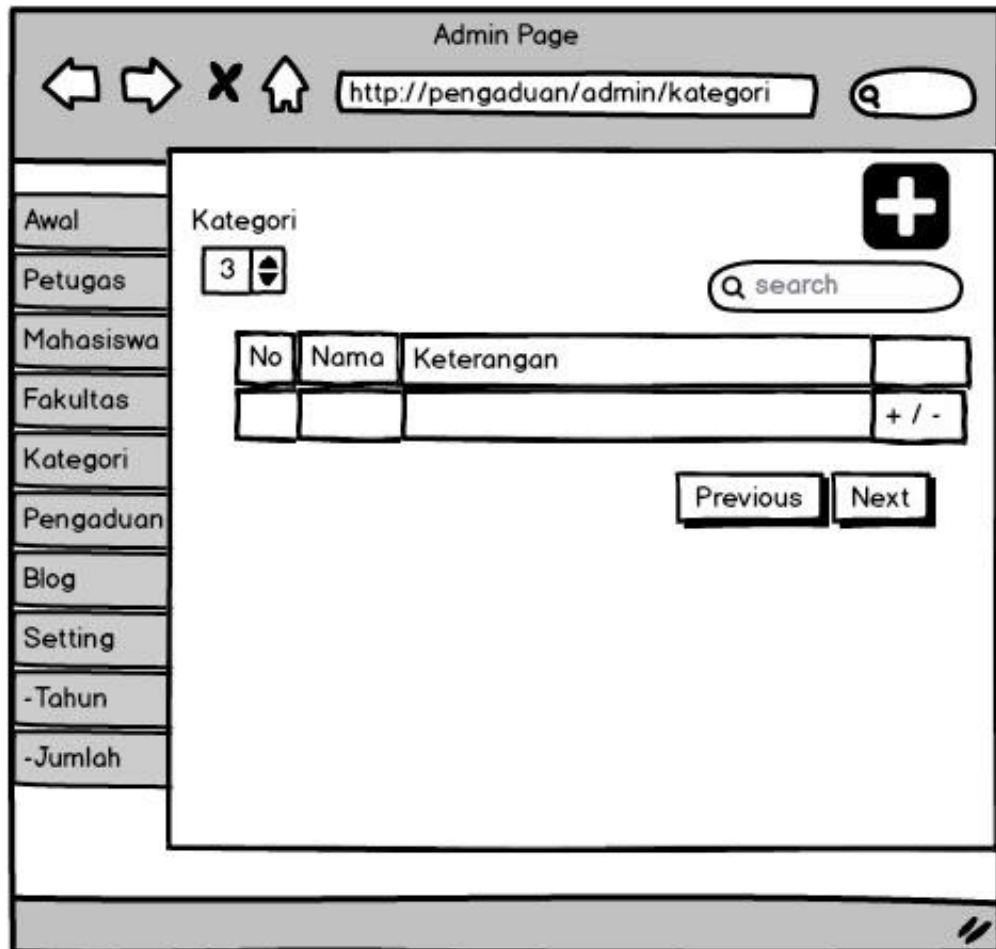
Rancangan antarmuka halaman fakultas merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mengubah data fakultas dan prodi. Gambaran rancangan antarmuka halaman fakultas dapat dilihat pada Gambar 3.16.



Gambar 3.16. Rancangan Halaman fakultas

3.7.5.10 Rancangan antarmuka halaman kategori

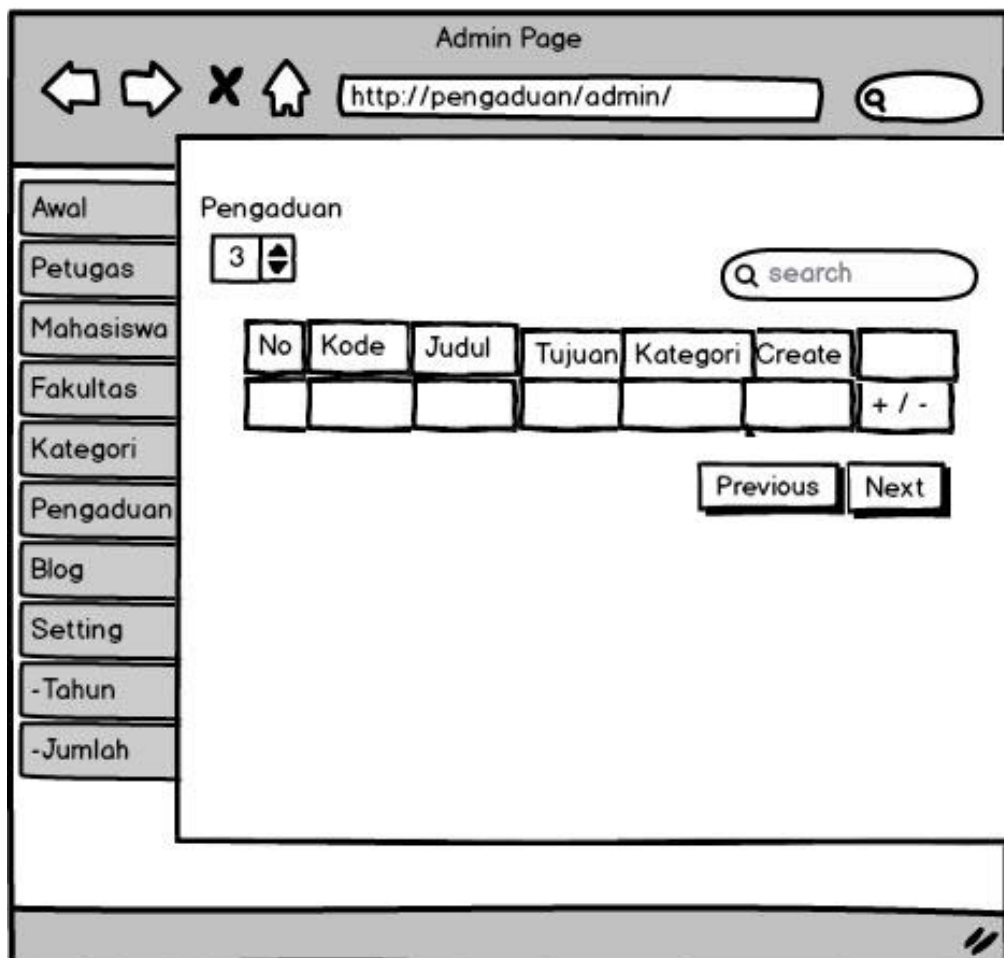
Rancangan antarmuka halaman kategori merupakan halaman yang digunakan untuk admin menambah atau menghapus dan mengubah data tujuan kategori. Gambaran rancangan antarmuka halaman kategori dapat dilihat pada Gambar 3.17.



Gambar 3.17. Rancangan Halaman kategori

3.7.5.11 Rancangan antarmuka halaman pengaduan

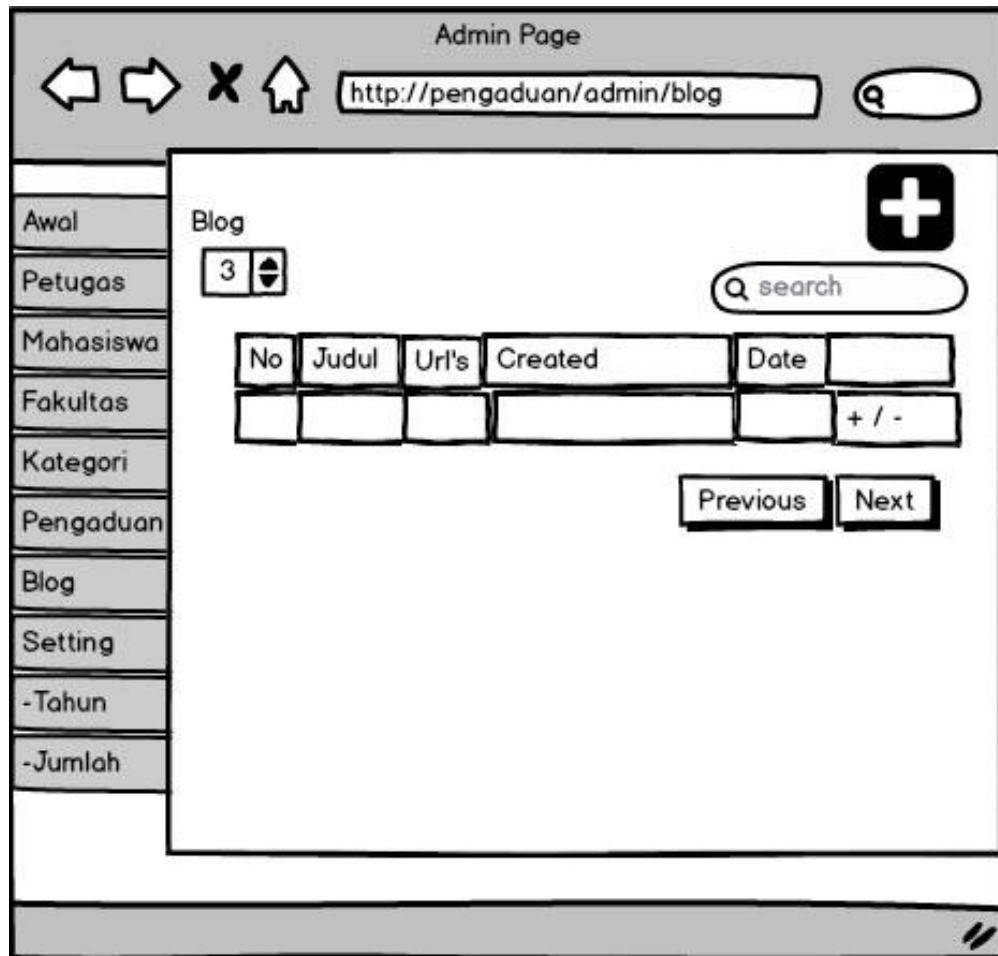
Rancangan antarmuka halaman pengaduan merupakan halaman yang digunakan untuk admin mendata keluhan atau aspirasi mahasiswa berdasarkan tujuan dan mengkonfirmasi data yang sudah dijalankan berdasarkan hasil keluhan dan aspirasi. Gambaran rancangan antarmuka halaman pengaduan dapat dilihat pada Gambar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



Gambar 3.18. Rancangan Halaman pengaduan

3.7.5.12 Rancangan antarmuka halaman pengeditan *blog*

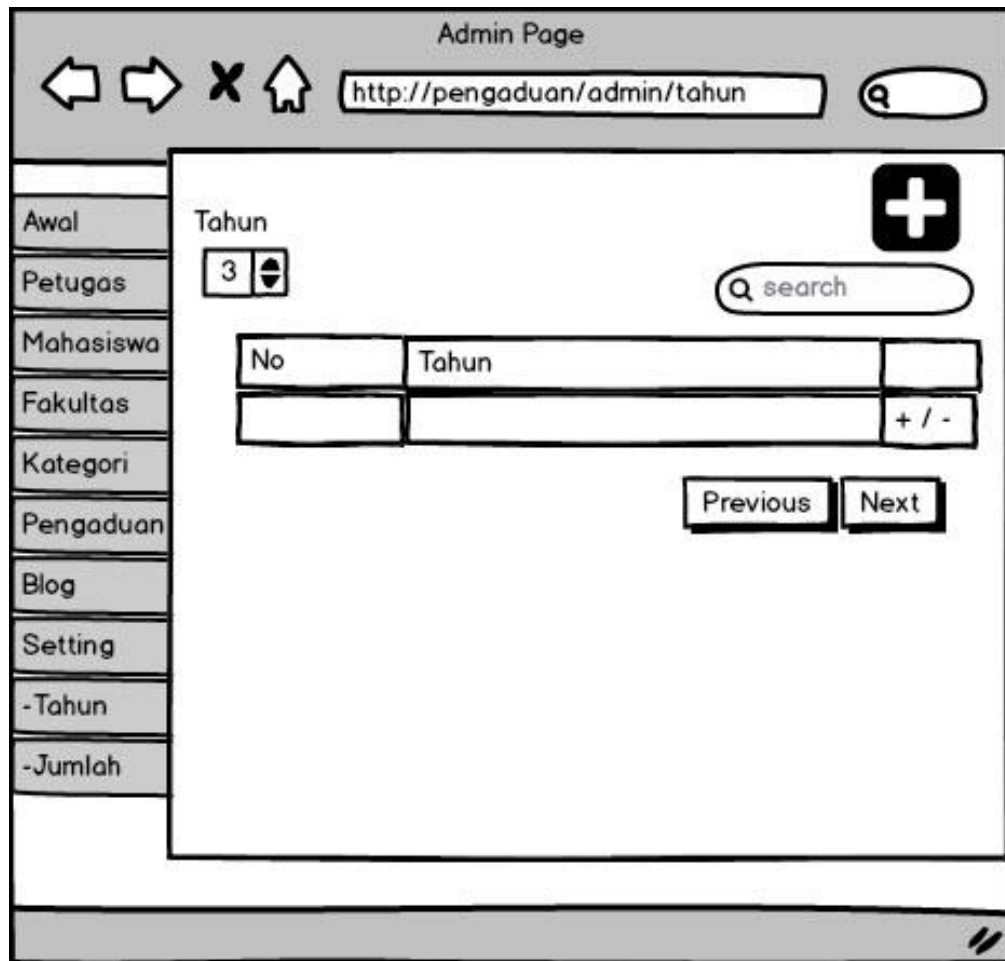
Rancangan antarmuka halaman pengeditan *blog* merupakan halaman yang digunakan untuk admin mengisi, mengedit, dan menambahkan isi *blog* dari admin. Gambaran rancangan antarmuka halaman pengeditan *blog* dapat dilihat pada Gambar 3.19.



Gambar 3.19. Rancangan Halaman pengeditan *blog*

3.7.5.13 Rancangan antarmuka halaman tahun

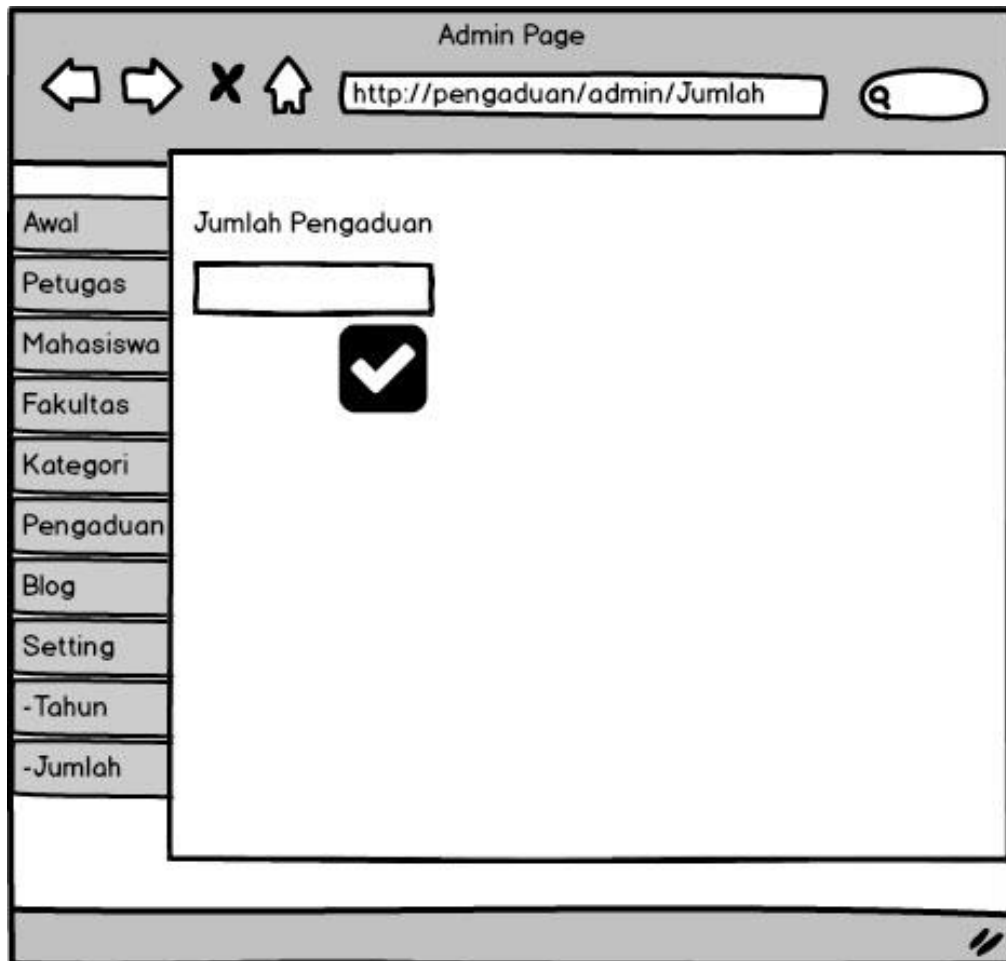
Rancangan antarmuka halaman tahun merupakan halaman yang digunakan untuk admin mengedit tahun angkatan untuk menambahkan mahasiswa berdasarkan tahun angkatan mahasiswa. Gambaran rancangan antarmuka halaman tahun dapat dilihat pada Gambar 3.20.



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.

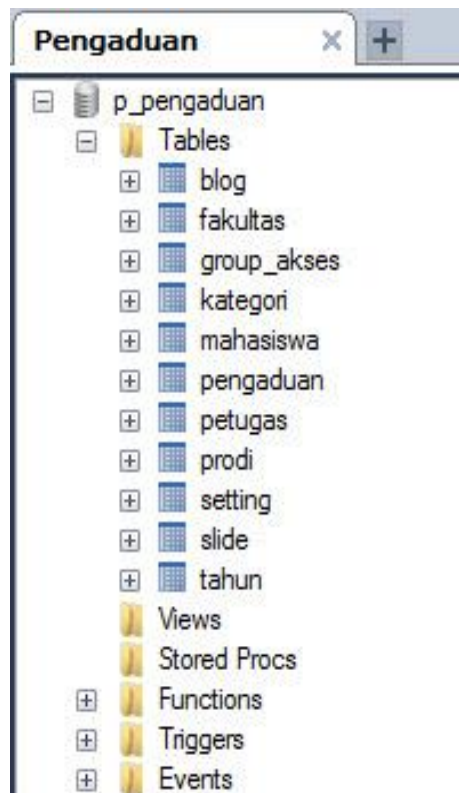
Hal-hal yang menjadi perhatian dalam pengujian adalah sebagai berikut:

- 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.

BAB IV HASIL DAN PEMBAHASAN

4.1 Database

Konfigurasi *database MySQL* dilakukan dengan menggunakan perangkat lunak pendukung *phpdesigner*. Hasil dari pembuatan *database* dalam aplikasi dapat dilihat pada gambar 4.1.

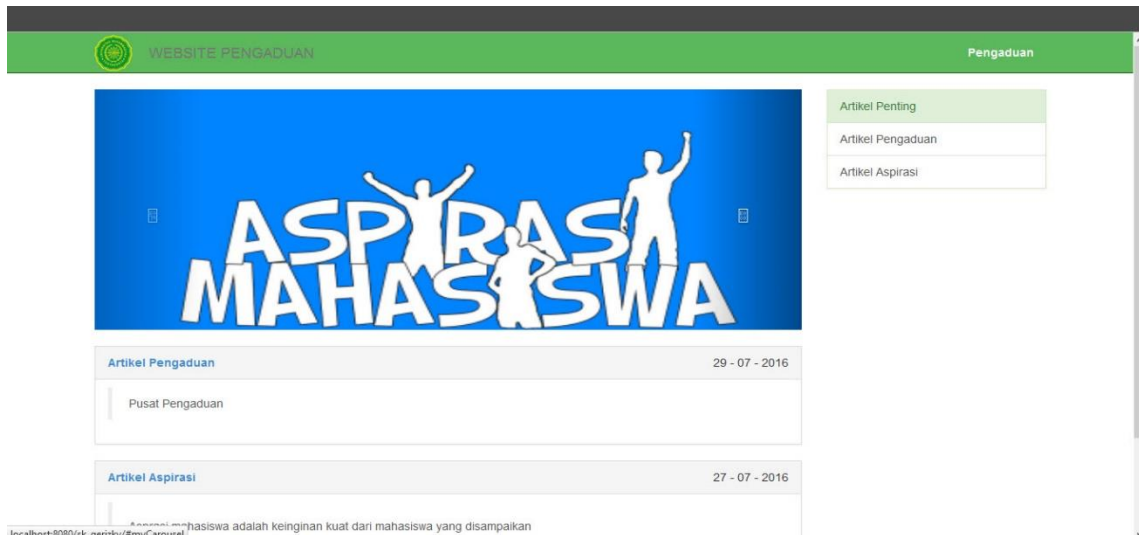


Gambar 4.1. Database Aplikasi

4.2 Antarmuka

4.2.1 Halaman Utama

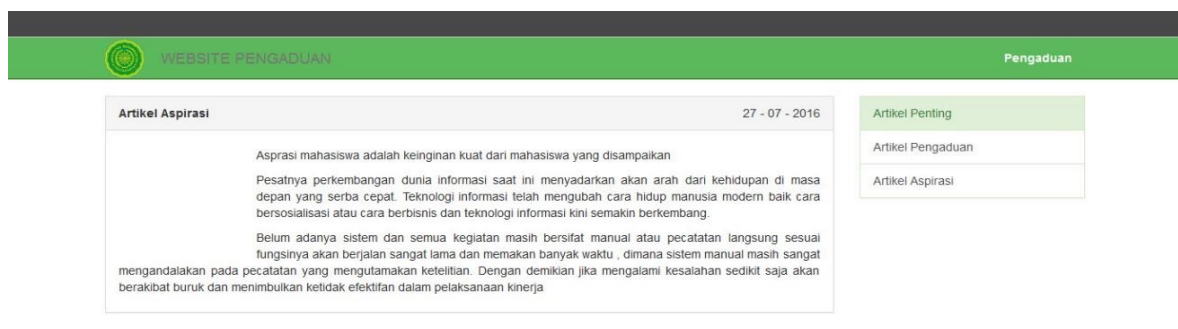
Halaman utama merupakan tampilan utama dari aplikasi yang dapat dilihat oleh mahasiswa dan admin. Tampilan halaman utama aplikasi dapat dilihat pada Gambar 4.2.



Gambar 4.2. Halaman Utama Aplikasi

4.2.2 Halaman Artikel

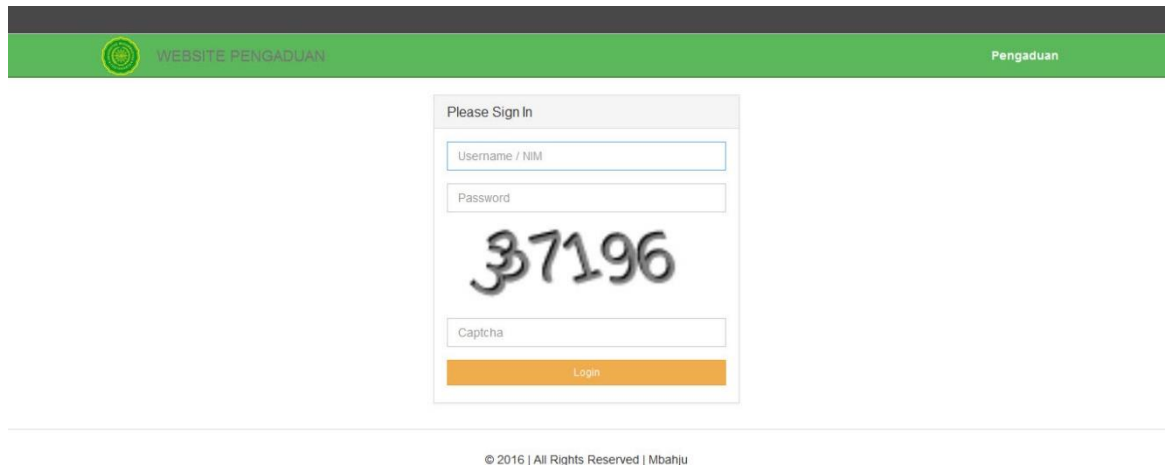
Halaman artikel berisi hal-hal yang berkaitan dengan kampus yang dapat dilihat oleh mahasiswa dan admin. Tampilan halaman artikel dapat dilihat pada gambar 4.3.



Gambar 4.3. Halaman Artikel

4.2.3 Halaman *Login* Mahasiswa

Halaman *login* mahasiswa berfungsi untuk memberikan keamanan di dalam sistem. Mahasiswa terlebih dahulu harus mengisi *Username* atau nim, *password* dan *captcha* yang benar sebelum masuk ke dalam sistem pengaduan. Tampilan halaman *login* mahasiswa dapat dilihat pada Gambar 4.4.



The image shows a web page for a student login system. At the top, there is a green navigation bar with a circular logo on the left, the text "WEBSITE PENGADUAN" in the center, and "Pengaduan" on the right. Below this is a white login form titled "Please Sign In". The form contains three input fields: "Username / NIM", "Password", and "Captcha". The "Captcha" field displays a distorted image of the number "37196". Below the input fields is an orange "Login" button. At the bottom of the page, there is a small copyright notice: "© 2016 | All Rights Reserved | Mbahju".

Gambar 4.4. Halaman *Login* Mahasiswa

4.2.4 Halaman Utama Pengaduan

Halaman utama pengaduan berfungsi untuk memberikan aspirasi atau keluhan kepada fakultas atau pun prodi berdasarkan kategori. Sebelum mahasiswa memberikan aspirasi dan keluhannya, mahasiswa harus mengisi judul, memilih tujuan pengaduan fakultas atau prodi, dan memilih kategori yang terdiri dari fasilitas, pelayanan, dan akademik. Tampilan halaman utama pengaduan dapat dilihat pada Gambar 4.5.

Gambar 4.5. Halaman Utama Pengaduan

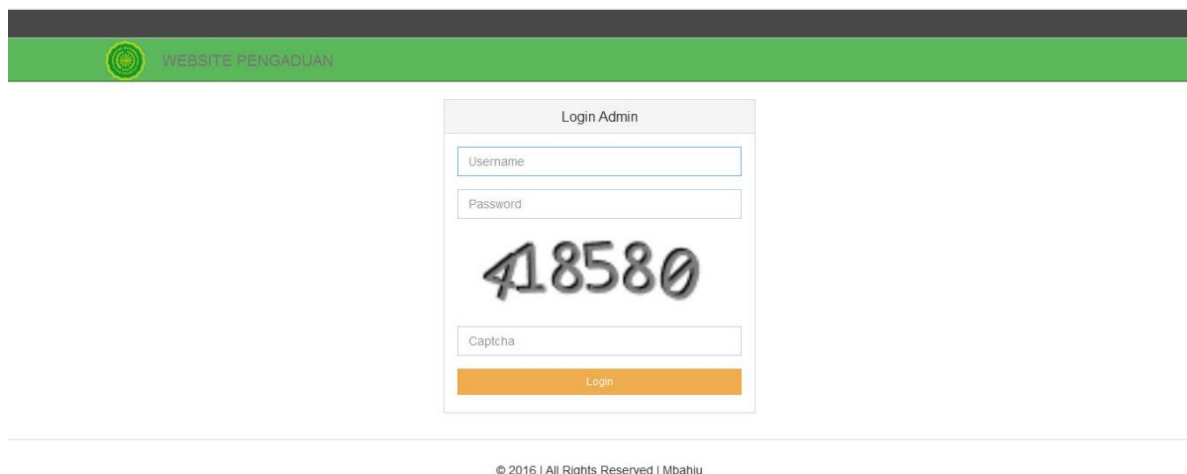
4.2.5 Halaman Selesai Pengaduan

Halaman selesai pengaduan merupakan tampilan dari mahasiswa yang sudah memberikan aspirasi dan keluhannya berdasarkan kategori dan tujuannya. Tampilan halaman selesai pengaduan dapat dilihat pada Gambar 4.6.

Gambar 4.6. Halaman Selesai Pengaduan

4.2.6 Halaman *Login Admin*

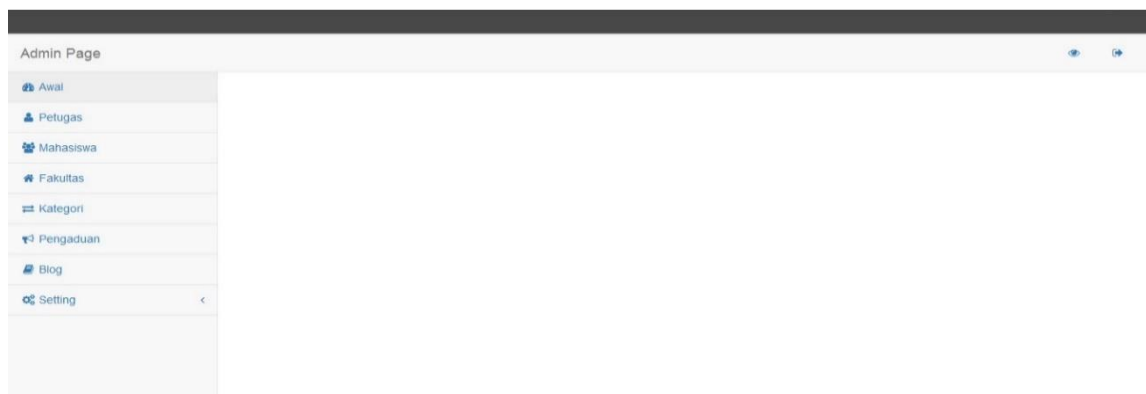
Halaman *login* merupakan halaman yang pertama muncul pada saat mulai menjalankan aplikasi. Halaman *login* berfungsi untuk memberikan keamanan di dalam sistem, admin terlebih dahulu harus mengisi *username*, *password* dan *captcha* yang benar sebelum masuk ke dalam sistem. Tampilan halaman *login* admin dapat dilihat pada Gambar 4.7.



Gambar 4.7. Halaman *Login Admin*

4.2.7 Halaman Utama Admin

Halaman utama admin merupakan tampilan utama dari aplikasi yang dapat dilihat oleh admin. Tampilan halaman utama admin dapat dilihat pada Gambar 4.8.



Gambar 4.8. Halaman Utama Admin

4.2.8 Halaman Petugas

Halaman petugas berfungsi untuk mendata petugas, menambahkan petugas, mencari petugas, mengubah data dan mengubah *password* petugas dan menghapus data petugas. Tampilan halaman petugas dapat dilihat pada Gambar 4.9.

	NAMA	TELP	USERNAME	ALAMAT
1	Gerizky	08991336214	gerizky	Taman Tirto kasihan bantul
2	admin	123	admin	UMY

Gambar 4.9. Halaman Petugas

4.2.9 Halaman Mahasiswa

Halaman mahasiswa berfungsi untuk mendata mahasiswa, menambahkan mahasiswa, mencari mahasiswa berdasarkan fakultas, prodi, dan tahun angkatan, menghapus data mahasiswa. Tampilan halaman mahasiswa dapat dilihat pada Gambar 4.10.

	KODE	NIM	NAMA	TAHUN	PRODI
FAKULTAS TEKNIK					
1	0001	20120110001	Tri Wahyu	2012	TEKNIK SIPIL
2	0001	20120120001	Dinda	2012	TEKNIK ELEKTRO
3	0001	20120130001	Aan Sukarno	2012	TEKNIK MESIN
4	0001	20120140001	Gerizky Saputra	2012	TEKNOLOGI INFORMASI
5	0002	20120140002	Hilman	2012	TEKNOLOGI INFORMASI
6	0003	20120140003	Wahyu	2012	TEKNOLOGI INFORMASI
FAKULTAS PERTANIAN					
7	0001	20120210001	Iqbaludin	2012	AGROTEKNOLOGI
8	0001	20120220001	Doni Ardi	2012	AGRIBISNIS
FAKULTAS KEDOKTERAN DAN ILMU KESEHATAN					

Gambar 4.10. Halaman Mahasiswa

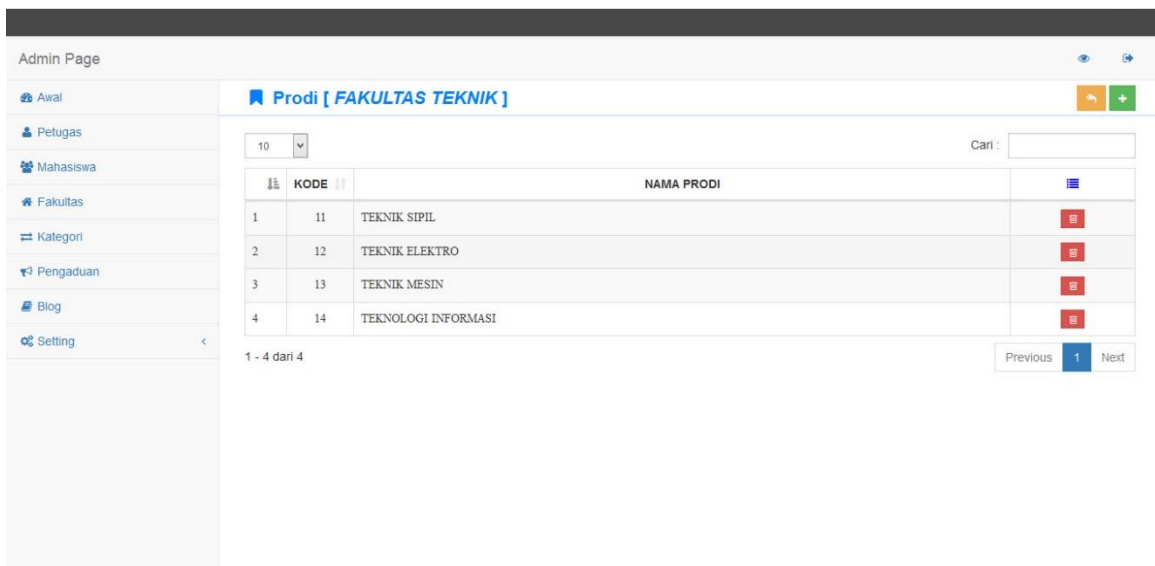
4.2.10 Halaman Fakultas

Halaman Fakultas berfungsi untuk mendata fakultas berdasarkan kode fakultas, menambah fakultas, mencari fakultas, dan menghapus fakultas. Tampilan halaman fakultas dapat dilihat pada Gambar 4.11.



Gambar 4.11. Halaman Fakultas

Halaman fakultas juga terdapat halaman prodi. Isi dari prodi tersebut berdasarkan data setiap fakultasnya. Tampilan halaman prodi dapat dilihat pada Gambar 4.12.



Gambar 4.12. Halaman Prodi

4.4.11 Halaman Kategori

Halaman kategori berfungsi untuk memberikan tujuan kepada mahasiswa, halaman kategori bisa menambahkan kategori, menghapus kategori, dan mencari kategori. Tampilan pada halaman kategori bisa dilihat pada Gambar 4.13.

	NAMA	KETERANGAN	
1	Akademik	bermasalah dengan akademik kampus	[Edit] [Delete]
2	Fasilitas	bermasalah dengan fasilitas kampus	[Edit] [Delete]
3	Pelayanan	bermasalah dengan pelayanan kampus	[Edit] [Delete]

Gambar 4.13. Halaman Kategori

4.4.12 Halaman Pengaduan

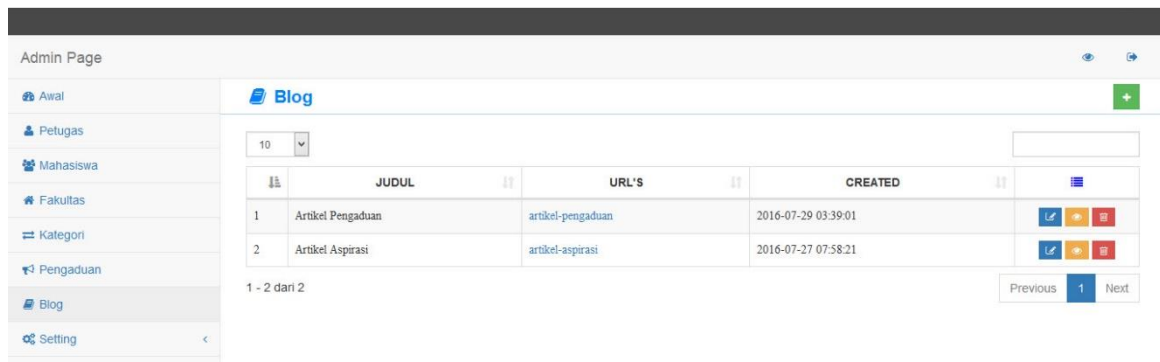
Halaman pengaduan berfungsi untuk mendata hasil pengaduan dari aspirasi dan keluhan mahasiswa, data pengaduan juga sudah menempatkan berdasarkan tujuan dan kategori. Halaman pengaduan juga berfungsi untuk mencari berdasarkan tujuan dan ketegori, petugas atau admin bisa konfirmasi pengaduan sudah ditangani atau pun belum ditangani, menghapus pengaduan yang sudah di tangani, dan mencetak hasil laporan. Tampilan halaman pengaduan dapat dilihat pada Gambar 4.14.

ID	KODE	JUDUL	TUJUAN	KATEGORI	CREATE	
1	VSPBWS	Sampah di tempat elektro	FAKULTAS TEKNIK	Pelayanan	24/08/2016	[Status] [Edit] [Delete]
2	VK1AXJ	Air mati di Lab elektro	TEKNIK ELEKTRO	Fasilitas	24/08/2016	[Status] [Edit] [Delete]
3	QKHFS	Tempat sampah penuh	FAKULTAS TEKNIK	Pelayanan	23/08/2016	[Status] [Edit] [Delete]
4	RWMPLN	Mencabul Nilai	FAKULTAS TEKNIK	Pelayanan	21/08/2016	[Status] [Edit] [Delete]
5	D3GOEP	AC Mati Lab D	TEKNIK SIPIL	Fasilitas	19/08/2016	[Status] [Edit] [Delete]
6	KCJVAZ	Bank uang tetap	FAKULTAS TEKNIK	Akademik	24/08/2016	[Status] [Edit] [Delete]
7	SOELV5	Mosor lab bermasalah	TEKNOLOGI INFORMASI	Fasilitas	22/08/2016	[Status] [Edit] [Delete]
8	JDOQIC	Ruang Angin Mati di Lab A	TEKNIK SIPIL	Fasilitas	20/08/2016	[Status] [Edit] [Delete]

Gambar 4.14. Halaman Pengaduan

4.4.13 Halaman *Blog*

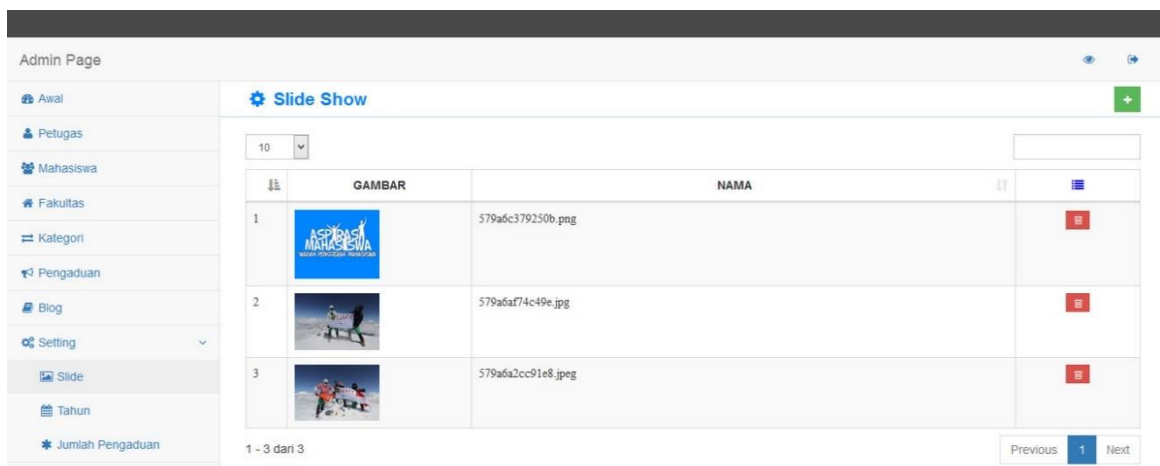
Halaman *blog* berfungsi untuk menambahkan artikel, mengedit artikel, menghapus artikel, mencari artikel. Tampilan halaman *blog* dapat dilihat pada Gambar 4.15.



Gambar 4.15. Halaman *Blog*

4.4.14 Halaman *Setting Slide*

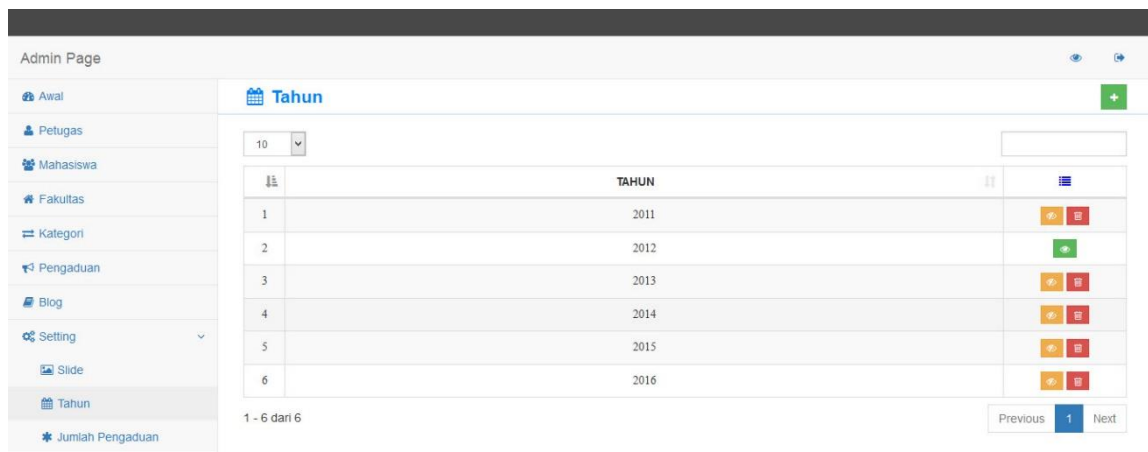
Halaman *setting slide* berfungsi untuk menambahkan gambar pada halaman utama agar lebih menarik, halaman *setting slide* juga bisa menghapus gambar. Tampilan halaman *setting slide* bisa dilihat pada Gambar 4.16.



Gambar 4.16. Halaman *Setting Slide*

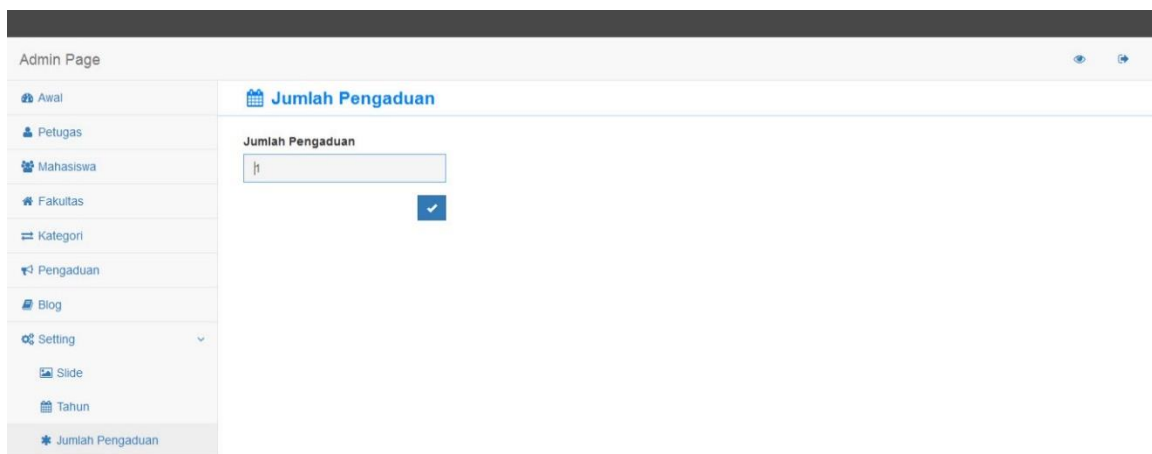
4.4.15 Halaman *Setting* Tahun

Halaman *setting* tahun berfungsi untuk mengaktifkan tahun angkatan mahasiswa yang akan di *input* pada halaman mahasiswa, halaman *setting* tahun bisa menambahkan tahun angkatan, menghapus tahun angkatan dan mengaktifkan tahun angkatan. Tampilan halaman *setting* tahun dapat dilihat pada Gambar 4.17.



Gambar 4.17. Halaman *Setting* Tahun

Halaman *setting* jumlah pengaduan berfungsi untuk memberi batasan pengaduan kepada mahasiswa yang akan memberikan aspirasi dan keluhan. Tampilan halaman *setting* jumlah pengaduan dapat dilihat pada Gambar 4.18.











Gambar 4.18. Halaman *Setting* Jumlah Pengaduan

4.4.16 Halaman Laporan Pengaduan

Halaman data pengaduan berfungsi untuk mendata hasil pengaduan yang akan di proses, data hasil pengaduan sudah berdasarkan filter pada tanggal, tujuan dan kategori yang mau di proses. Tampilan halaman Laporan pengaduan dapat dilihat pada Gambar 4.19

.: Data Pengaduan .:

Filter Pada : 19-08-2016 s/d 25-08-2016




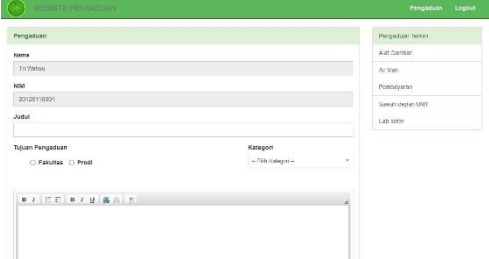
No.	Judul	Tujuan dan Isi	Kategori	Tanggal
1	Sampah di tempat elektro Kode : I2P8H8 Pada : 24/08/2016	FAKULTAS TEKNIK 	Pelayanan	2016-08-24 11:32:00
2	Ac mati di Lab elektro Kode : IKTLEJ Pada : 24/08/2016	TEKNIK ELEKTRO 	Facilitas	2016-08-24 11:30:07
3	Tempat sampah penuh Kode : QKHFLS Pada : 24/08/2016	FAKULTAS TEKNIK 	Pelayanan	2016-08-23 02:29:27
4	Masalah Nilai Kode : RHWAPN Pada : 24/08/2016	FAKULTAS TEKNIK 	Pelayanan	2016-08-21 02:21:52
5	AC Mati Lab D Kode : DSGOEP Pada : 19/08/2016	TEKNIK SIPIL 	Facilitas	2016-08-19 02:17:16
6	Bank sering tutup Kode : KZTLGZ Pada : 24/08/2016	FAKULTAS TEKNIK 	Akademik	2016-08-24 02:30:42
7	Mouse lab bermasalah Kode : SOEFLS Pada : 22/08/2016	TEKNOLOGI INFORMASI 	Facilitas	2016-08-22 02:23:29
8	Kipas Angin Mati di Lab A Kode : JEDQNK Pada : 20/08/2016	TEKNIK SIPIL 	Facilitas	2016-08-20 02:19:06

Gambar 4.19. Halaman Laporan Pengaduan

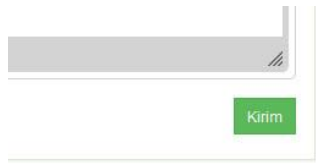



4.3 Pengujian





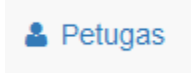

Pengujian dilakukan bertujuan untuk melihat hasil eksekusi dan fungsionalitas berdasarkan analisis kebutuhan. Hasil pengujian dapat dilihat pada Tabel 4.1.


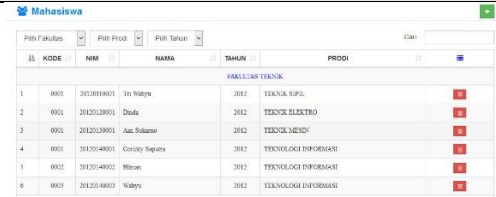


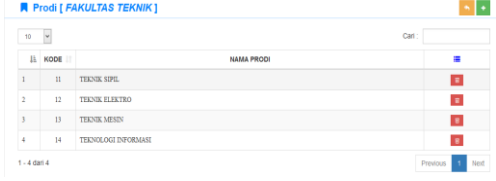
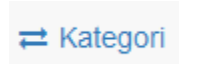
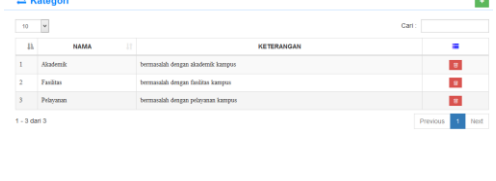
Tabel 4.1. Hasil Pengujian *Website*

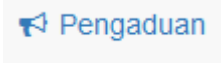
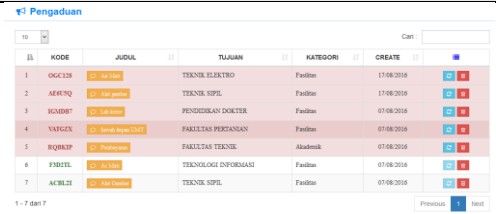

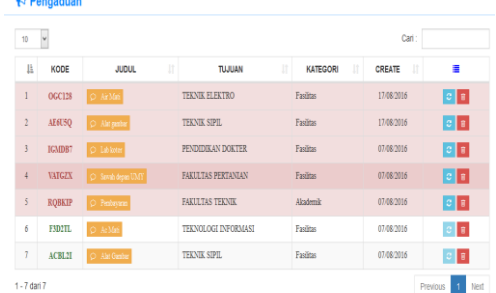

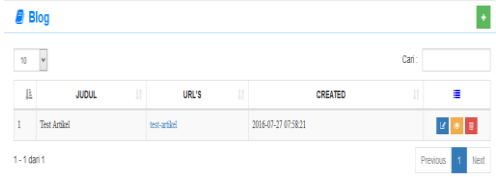
No	Kasus Uji	Test Case	Hasil yang Diharapkan	Hasil	Status
1.	Tombol PENGADUAN pada halaman masuk		Pengguna masuk ke halaman <i>login</i> mahasiswa		Berhasil
2.	LOGIN pada halaman masuk pengaduan dengan <i>authentication</i> valid		Pengguna masuk ke halaman pengaduan		Berhasil

No	Kasus Uji	Test Case	Hasil yang Diharapkan	Hasil	Status
3.	LOGIN pada halaman pengaduan dengan <i>authentication</i> tidak valid		Pengguna tidak dapat masuk ke halaman pengaduan		Berhasil
4.	Tombol PENGADUAN pada halaman masuk		Pengguna masuk ke halaman isi pengaduan		Berhasil
5.	Tombol KIRIM pada halaman isi pengaduan		Data dari pengguna masuk ke <i>database</i> , Setelah tombol diklik		Berhasil

No	Kasus Uji	Test Case	Hasil yang Diharapkan	Hasil	Status
6.	Data yang sudah sukses dikirim di halaman pengaduan		Diberikan peringatan kepada pengguna		Berhasil
7.	Tombol LOGOUT pada halaman pengaduan		Pengguna masuk ke halaman awal pengaduan		Berhasil

No	Kasus Uji	Test Case	Hasil yang Diharapkan	Hasil	Status
8.	LOGIN pada halaman login admin dengan authentication valid		Setelah tombol diklik pengguna masuk ke halaman admin		Berhasil
9.	LOGIN pada halaman login admin dengan authentication tidak valid		Setelah tombol diklik pengguna tidak dapat masuk ke halaman admin		Berhasil
10.	Tombol PETUGAS pada halaman admin		Setelah tombol petugas diklik pengguna masuk ke halaman petugas		Berhasil

No	Kasus Uji	Test Case	Hasil yang Diharapkan	Hasil	Status
11.	Tombol MAHASISWA pada halaman admin		Setelah tombol petugas diklik pengguna masuk ke halaman mahasiswa		Berhasil
12.	Tombol FAKULTAS pada halaman admin		Setelah tombol diklik pengguna masuk ke halaman fakultas		Berhasil
13.	Tombol PRODI pada halaman fakultas		Setelah tombol diklik pengguna masuk ke halaman prodi		Berhasil
14.	Tombol KATEGORI pada halaman admin		Setelah tombol diklik pengguna masuk ke halaman kategori		Berhasil

No	Kasus Uji	Test Case	Hasil yang Diharapkan	Hasil	Status																																																
15.	Tombol PENGADUAN pada halaman admin		Setelah tombol diklik pengguna masuk ke halaman pengaduan	 <table border="1"> <thead> <tr> <th>ID</th> <th>KODE</th> <th>JUDUL</th> <th>TUJUAN</th> <th>KATEGORI</th> <th>CREATE</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>OGC128</td> <td>Isi Matri</td> <td>TEKNIK ELEKTRO</td> <td>Fasilitas</td> <td>17-08-2016</td> </tr> <tr> <td>2</td> <td>AEK45Q</td> <td>Isi gambar</td> <td>TEKNIK SIPIL</td> <td>Fasilitas</td> <td>17-08-2016</td> </tr> <tr> <td>3</td> <td>IKM087</td> <td>Isi kata</td> <td>PENGEKSIAN DOKTER</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> <tr> <td>4</td> <td>VATGZX</td> <td>Isi link Apps CSIT</td> <td>FAKULTAS PERTANIAN</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> <tr> <td>5</td> <td>BQBRKP</td> <td>Isi dokumen</td> <td>FAKULTAS TEKNIK</td> <td>Akademik</td> <td>07-08-2016</td> </tr> <tr> <td>6</td> <td>FIDUTL</td> <td>Isi Matri</td> <td>TEKNOLOGI INFORMASI</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> <tr> <td>7</td> <td>ACBLLI</td> <td>Isi Gambar</td> <td>TEKNIK SIPIL</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> </tbody> </table>	ID	KODE	JUDUL	TUJUAN	KATEGORI	CREATE	1	OGC128	Isi Matri	TEKNIK ELEKTRO	Fasilitas	17-08-2016	2	AEK45Q	Isi gambar	TEKNIK SIPIL	Fasilitas	17-08-2016	3	IKM087	Isi kata	PENGEKSIAN DOKTER	Fasilitas	07-08-2016	4	VATGZX	Isi link Apps CSIT	FAKULTAS PERTANIAN	Fasilitas	07-08-2016	5	BQBRKP	Isi dokumen	FAKULTAS TEKNIK	Akademik	07-08-2016	6	FIDUTL	Isi Matri	TEKNOLOGI INFORMASI	Fasilitas	07-08-2016	7	ACBLLI	Isi Gambar	TEKNIK SIPIL	Fasilitas	07-08-2016	Berhasil
ID	KODE	JUDUL	TUJUAN	KATEGORI	CREATE																																																
1	OGC128	Isi Matri	TEKNIK ELEKTRO	Fasilitas	17-08-2016																																																
2	AEK45Q	Isi gambar	TEKNIK SIPIL	Fasilitas	17-08-2016																																																
3	IKM087	Isi kata	PENGEKSIAN DOKTER	Fasilitas	07-08-2016																																																
4	VATGZX	Isi link Apps CSIT	FAKULTAS PERTANIAN	Fasilitas	07-08-2016																																																
5	BQBRKP	Isi dokumen	FAKULTAS TEKNIK	Akademik	07-08-2016																																																
6	FIDUTL	Isi Matri	TEKNOLOGI INFORMASI	Fasilitas	07-08-2016																																																
7	ACBLLI	Isi Gambar	TEKNIK SIPIL	Fasilitas	07-08-2016																																																
16.	Data pengaduan yang pernah dikirim di halaman isi pengaduan		Data diambil dari <i>database</i> dan di tampilkan pada halaman pengaduan berdasarkan judul, tujuan, kategori, dan tanggal	 <table border="1"> <thead> <tr> <th>ID</th> <th>KODE</th> <th>JUDUL</th> <th>TUJUAN</th> <th>KATEGORI</th> <th>CREATE</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>OGC128</td> <td>Isi Matri</td> <td>TEKNIK ELEKTRO</td> <td>Fasilitas</td> <td>17-08-2016</td> </tr> <tr> <td>2</td> <td>AEK45Q</td> <td>Isi gambar</td> <td>TEKNIK SIPIL</td> <td>Fasilitas</td> <td>17-08-2016</td> </tr> <tr> <td>3</td> <td>IKM087</td> <td>Isi kata</td> <td>PENGEKSIAN DOKTER</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> <tr> <td>4</td> <td>VATGZX</td> <td>Isi link Apps CSIT</td> <td>FAKULTAS PERTANIAN</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> <tr> <td>5</td> <td>BQBRKP</td> <td>Isi dokumen</td> <td>FAKULTAS TEKNIK</td> <td>Akademik</td> <td>07-08-2016</td> </tr> <tr> <td>6</td> <td>FIDUTL</td> <td>Isi Matri</td> <td>TEKNOLOGI INFORMASI</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> <tr> <td>7</td> <td>ACBLLI</td> <td>Isi Gambar</td> <td>TEKNIK SIPIL</td> <td>Fasilitas</td> <td>07-08-2016</td> </tr> </tbody> </table>	ID	KODE	JUDUL	TUJUAN	KATEGORI	CREATE	1	OGC128	Isi Matri	TEKNIK ELEKTRO	Fasilitas	17-08-2016	2	AEK45Q	Isi gambar	TEKNIK SIPIL	Fasilitas	17-08-2016	3	IKM087	Isi kata	PENGEKSIAN DOKTER	Fasilitas	07-08-2016	4	VATGZX	Isi link Apps CSIT	FAKULTAS PERTANIAN	Fasilitas	07-08-2016	5	BQBRKP	Isi dokumen	FAKULTAS TEKNIK	Akademik	07-08-2016	6	FIDUTL	Isi Matri	TEKNOLOGI INFORMASI	Fasilitas	07-08-2016	7	ACBLLI	Isi Gambar	TEKNIK SIPIL	Fasilitas	07-08-2016	Berhasil
ID	KODE	JUDUL	TUJUAN	KATEGORI	CREATE																																																
1	OGC128	Isi Matri	TEKNIK ELEKTRO	Fasilitas	17-08-2016																																																
2	AEK45Q	Isi gambar	TEKNIK SIPIL	Fasilitas	17-08-2016																																																
3	IKM087	Isi kata	PENGEKSIAN DOKTER	Fasilitas	07-08-2016																																																
4	VATGZX	Isi link Apps CSIT	FAKULTAS PERTANIAN	Fasilitas	07-08-2016																																																
5	BQBRKP	Isi dokumen	FAKULTAS TEKNIK	Akademik	07-08-2016																																																
6	FIDUTL	Isi Matri	TEKNOLOGI INFORMASI	Fasilitas	07-08-2016																																																
7	ACBLLI	Isi Gambar	TEKNIK SIPIL	Fasilitas	07-08-2016																																																
17.	Tombol <i>BLOG</i> pada halaman admin		Setelah tombol diklik pengguna masuk ke halaman <i>blog</i>	 <table border="1"> <thead> <tr> <th>ID</th> <th>JUDUL</th> <th>URL'S</th> <th>CREATED</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Test Artikel</td> <td>test-artikel</td> <td>2016-07-27 07:58:21</td> </tr> </tbody> </table>	ID	JUDUL	URL'S	CREATED	1	Test Artikel	test-artikel	2016-07-27 07:58:21	Berhasil																																								
ID	JUDUL	URL'S	CREATED																																																		
1	Test Artikel	test-artikel	2016-07-27 07:58:21																																																		

No	Kasus Uji	Test Case	Hasil yang Diharapkan	Hasil	Status
18.	Tombol <i>SLIDE</i> pada halaman admin		Setelah tombol diklik pengguna masuk ke halaman <i>slide</i>		Berhasil
19.	Tombol TAHUN pada halaman admin		Setelah tombol diklik pengguna masuk ke halaman tahun		Berhasil
20.	Tombol JUMLAH PENGADUAN pada halaman admin		Setelah tombol diklik pengguna masuk ke halaman jumlah pengaduan		Berhasil

4.4 Pembahasan

Dari pengujian perangkat lunak dapat dirangkum sebagai berikut:

1. Informasi dari mahasiswa dapat disampaikan melalui fasilitas *web* penanganan keluhan, Informasi yang dapat disampaikan oleh mahasiswa adalah:
 - a. Akademik.
 - b. Fasilitas.
 - c. Pelayanan.
2. Tanggapan admin dapat disampaikan melalui fasilitas *web* penanganan keluhan Tanggapan yang dapat disampaikan admin dapat berbentuk:
 - a. Konfirmasi dalam bentuk *notification* ✓ bahwa pengaduan atau keluhan sudah di tangani.
 - b. Konfirmasi dalam bentuk *notification* ✗ bahwa pengaduan atau keluhan belum di tangani.
3. *Software* ini memiliki kelebihan sebagai berikut:
 - c. Mudah untuk digunakan oleh mahasiswa.
 - d. *Software* mampung menampung keluhan mahasiswa secara *online*.
 - e. Memiliki fitur tujuan berdasarkan kategori, fakultas, dan prodi.
 - f. *Software* penanganan keluhan bisa memberikan batasan pengaduan kepada mahasiswa.
 - g. Mampu menghasilkan laporan pengaduan.
4. *Software* ini memiliki keterbatasan sebagai berikut:
 - a. *Software* belum bisa memberikan tindakan secara bertahap, dan tercatat hanya sekedar *notification*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil dan pembahasan yang telah dilakukan dalam Pengembangan Sistem Penanganan Keluhan Berbasis *Web* (Studi Kasus: Fakultas Teknik Universitas Muhammadiyah Yogyakarta), maka diperoleh kesimpulan sebagai berikut:

1. Perangkat lunak mampu menampung keluhan secara *online* yang dapat digunakan oleh mahasiswa Fakultas Teknik Universitas Muhammadiyah Yogyakarta dalam memberikan keluhan atau aspirasi mahasiswa berdasarkan kategori dan tujuan.
2. Admin program studi dapat memberikan tanggapan dalam bentuk *notification* bahwa pengaduan atau keluhan sudah di tangani dan dapat dilihat oleh mahasiswa.

5.2 Saran

Hal yang masih dapat dikembangkan ke dalam sistem adalah:

1. Penyempurnaan pada tampilan dibuat lebih menarik.
2. Tambahan fitur-fitur yang sesuai kebutuhan dari fakultas.
3. Disediakan penanganan keluhan dalam bentuk *mobile* agar mahasiswa lebih mudah dalam memberikan keluhan atau aspirasi.

DAFTAR PUSTAKA

- Drs. Zulkifli Amsyah, M. (2000). In *Sistem* (p. 4).
- expresiku. (n.d.). *Analisa dan Perancangan Basis Data*. Retrieved from expresiku.wordpress.com:
<https://expresiku.wordpress.com/category/ilmu-komputer/analisa-dan-perancangan-basis-data/>
- Fat. (2012, 11 1). *Sistem Informasi*. Retrieved from [Sarjanaku.com](http://www.sarjanaku.com):
<http://www.sarjanaku.com/2012/11/pengertian-sistem-menurut-para-ahli.html>
- Fatta, A. (2007). Andi. In *Analisis dan Perancangan Sistem Informasi*. Yogyakarta.
- James. (2006). *Pengertian dan Info*. Retrieved from [Pengertian-pengertian-info.blogspot.com](http://pengertian-pengertian-info.blogspot.com): <http://pengertian-pengertian-info.blogspot.co.id/2015/11/pengertian-penanganan-keluhan-menurut.html>
- Jogiyanto. (2005, 3 1). *Pengertian Sistem Informasi Menurut Para Ahli*. In S. T. Informasi, *Sistem Teknologi Informasi*. Yogyakarta: Andi Offset. Retrieved from [ArtikelTeknologi.com](http://www.artikelteknologi.com):
<http://www.artikelteknologi.com/2015/11/pengertian-sistem-informasi-menurut-ahli.html>
- Lani, S. (1995, 11 1). *Sistem Informasi*. Retrieved from [Sarjanaku.com](http://www.sarjanaku.com):
<http://www.sarjanaku.com/2012/11/pengertian-sistem-menurut-para-ahli.html>
- Madcoms, L. (2011). In *Aplikasi Web Database Dengan Dreamweaver dan php-MySQL*. Yogyakarta: Andi.
- Murdick. (1991, 11 1). *Sistem Informasi Untuk Manajemen Modern*. Jakarta: Erlangga. Retrieved from [Sarjanaku.com](http://www.sarjanaku.com):
<http://www.sarjanaku.com/2012/11/pengertian-sistem-menurut-para-ahli.html>
- Rekompak. (n.d.). *Artikel*. Retrieved from [rekompakjrf.com](http://www.rekompakjrf.com):
<http://www.rekompakjrf.org/?act=detailartikel&id=28>
- Rickyanto. (2002). In *Desain Web Dengan Dreamweaver MX*. Elex Media. Komputindo.
- Roger S, p. (2002). In *“Rekayasa Perangkat Lunak Pendekatan Praktisi”*, edisi pertama. Yogyakarta: Andi.

- Roscoe, R. A. (2013, 6 1). *Pengertian Sistem Informasi*. Retrieved from Kumpulan arti & contoh: <http://articontohnya.blogspot.co.id/2013/06/pengertian-sistem-informasi-oleh-robert.html>
- Sholiq. (n.d.). Sholiq. In “*Pemodalan Sistem Informasi Berorientasi Objek dengan UML, edisi pertama*”. Yogyakarta: Graha Ilmu.
- Simarmata. (2009). In *Pengembangan Sistem Berbasis Web*.
- Sommerville. (2011). In “*Rekayasa Perangkat Lunak*”. Jakarta: Erlangga.
- Sudarmo. (2006). Unggul. In *PHP*. Jakarta: Phibeta.

LAMPIRAN

<pre> Pengaduan: <script src="<?php echo \$dir; ?>assets/js/jquery- 1.11.1.min.js"></script> <script type="text/javascript" src="<?php echo ROOT_DOCUMENT; ?>assets/plugin/ckeditor/ckeditor.js"></script> <script src="<?php echo \$dir; ?>assets/js/jquery.validate.js"></script> <link href="<?php echo \$dir; ?>assets/plugin/chosen/chosen.css" rel="stylesheet"/> <div class="col-sm-9"> <div class="panel panel-success"> <div class="panel-heading"> <i class="fa fa-bullhorn fa-flat fa- fw"></i> Pengaduan </div> <div class="panel-body"> <?php \$cnpng = mysql_query("SELECT COUNT(pg_id) AS cndt FROM pengaduan WHERE DATE_FORMAT(pg_create,'%y-%m-%d') = DATE_FORMAT(CURRENT_DATE,'%y-%m- %d') AND mhs_id = ".\$_SESSION['id']."' LIMIT 1"); if(mysql_num_rows(\$cnpng) > 0){ \$count_data = mysql_fetch_object(\$cnpng); \$q_jum = mysql_query("SELECT * FROM setting LIMIT 1"); if(mysql_num_rows(\$q_jum) < 0){ \$jum = 2; } else { \$d = mysql_fetch_object(\$q_jum); \$jum = (!empty(\$d- >set_pengaduan) ? \$d->set_pengaduan : '2'); } if (\$count_data->cndt < \$jum) { </pre>	<pre> Pengaduan pada Admin: <?php require_once('../core/koneksi.php'); require_once('../core/root.php'); ?> <script src="<?php echo \$dir; ?>assets/js/jquery- 1.11.1.min.js"></script> <?php if(!empty(\$_GET['d'])) { \$d = \$_GET['d']; \$query = mysql_query("SELECT pengaduan.*, kategori.*, mahasiswa.* FROM pengaduan LEFT JOIN kategori on (pengaduan.kat_id = kategori.kat_id) LEFT JOIN mahasiswa on (pengaduan.mhs_id = mahasiswa.mhs_id) WHERE pengaduan.pg_id = ".\$d." limit 1"); \$data = mysql_fetch_object(\$query); } ?> <div class="modal-dialog modal-lg"> <div class="modal-content"> <form action="<?php echo ADMIN_ROOT; ?>postdata" method="post" accept-charset="utf-8" enctype="multipart/form-data" id="myForm"> <div class="modal-header"> <button type="button" class="close" data- dismiss="modal" aria- hidden="true">&times;</button> <h4 class="modal-title"><i class="fa fa- bullhorn fa-fw"></i> Pengaduan</h4> </div> <div class="modal-body"> <div class=""> <div class="row"> <div class="col-sm-3"> NIM </div> <div class="col-sm-9"> : <?php echo (!empty(\$data- >mhs_nim) ? \$data->mhs_nim : ""); ?> </div> </pre>
--	---

<pre> \$querydata = mysql_query("SELECT mahasiswa.mhs_id, mahasiswa.mhs_nim, mahasiswa.mhs_nama, fakultas.fk_nama, prodi.pr_nama FROM mahasiswa LEFT JOIN prodi on (mahasiswa.pr_id = prodi.pr_id) LEFT JOIN fakultas on (fakultas.fk_id = prodi.fk_id) WHERE mahasiswa.mhs_id = ".\$_SESSION['id']." LIMIT 1"); \$datamhs = mysql_fetch_object(\$querydata); ?> <form class="" method="post" action=""<?php echo ROOT_DOCUMENT.'trims.html'; ?> id="myForm" name="myForm"> <div class="form-group"> <label>Nama</label> <input class="form-control" value=""<?php echo (!empty(\$datamhs- >mhs_nama) ? \$datamhs->mhs_nama : ""); ?> readonly=""> </div> <div class="form-group"> <label>NIM</label> <input class="form-control" value=""<?php echo (!empty(\$datamhs- >mhs_nim) ? \$datamhs->mhs_nim : ""); ?> readonly=""> </div> <div class="form-group"> <label>Judul</label> <input class="form-control" name="pg_judul" value=""> <!-- <p class="help-block">Judul Pengaduan</p --> </div> <div class="row"> <div class="col-sm-4"> </pre>	<pre> </div> <div class="row"> <div class="col-sm-3"> Mahasiswa </div> <div class="col-sm-9"> : <?php echo (!empty(\$data- >mhs_nama) ? \$data->mhs_nama : ""); ?> </div> </div> <div class="row"> <div class="col-sm-3"> Judul </div> <div class="col-sm-9"> : <?php echo (!empty(\$data- >pg_judul) ? \$data->pg_judul : ""); ?> </div> </div> <div class="row"> <div class="col-sm-3"> Kategori </div> <div class="col-sm-9"> : <?php echo (!empty(\$data- >kat_nama) ? \$data->kat_nama : ""); ?> </div> </div>
 <div class="row"> <div class="col-sm-3"> Isi Pengaduan </div> <div class="col-sm-9"> <div class=""> <blockquote> <?php echo (!empty(\$data- >pg_isi) ? \$data->pg_isi : ""); ?> </blockquote> </div> <div class="pull-right"> </pre>
---	--

<pre> <div class="form-group"> <label>Tujuan Pengaduan</label> <div class="my-radio"> <label class="radio-inline"> <input type="radio" name="pg_tujuan" value="<?php echo (!empty(\$datamhs->fk_nama) ? \$datamhs- >fk_nama : 'Fakultas'); ?>" /> Fakultas </label> <label class="radio-inline"> <input type="radio" name="pg_tujuan" value="<?php echo (!empty(\$datamhs->pr_nama) ? \$datamhs- >pr_nama : 'Prodi'); ?>" /> Prodi </label> </div> </div> </div> <div class="col-sm-4"> </div> <div class="col-sm-4"> <div class="form-group"> <div class="form-group"> <label>Kategori</label> <select class="form-control" name="kat_id" id="kategori"> <option value="">-- Pilih Kategori --</option> <?php \$kategori = mysql_query("SELECT * FROM kategori ORDER BY kat_nama ASC"); if(mysql_num_rows(\$kategori) > 0){ while(\$data = mysql_fetch_object(\$kategori)){ ?> <option value="<?php echo \$data->kat_id; ?>"><?php echo \$data- >kat_nama; ?></option> <?php }} ?> </select> </div> </div> </div> </pre>	<pre> <?php echo date("d/m/Y", strtotime(\$data->pg_create)); ?> </div> </div> </div> </div> <div style="clear: both;"></div> <div class="modal-footer"> <button type="button" class="btn btn- default btn-sm" data-dismiss="modal" id="clsfm"><i class="glyphicon glyphicon- remove"></i></button> </div> </form> </div> </div> Pengaduan pada Admin: <link href="<?php echo \$dir; ?>assets/plugin/datatables/media/css/dataTables.bo otstrap.css" rel="stylesheet"/> <script src="<?php echo \$dir; ?>assets/plugin/datatables/media/js/jquery.dataTabl es.js"></script> <script src="<?php echo \$dir; ?>assets/plugin/datatables/media/js/dataTables.boot strap.js"></script> <div class="row"> <div class="col-sm-12"> <div class="pull-right mdfbuttonup"> </div> <h1 class="page-header"><i class="fa fa- bullhorn fa-fw"></i> Pengaduan</h1> </div> </div> <div class="row"> <div class="col-lg-12"> <div class="mdfcontainer"> <div class="table-responsive"> <table class="table table-striped table- bordered table-hover" id="tabeldata"> <thead> <tr> <th width="10"></th> <th>Kode</th> <th>Judul</th> <th>Tujuan</th> </pre>
--	--

<pre> </div> </div> </div> <div class="form-group"> <label></label> <textarea class="form-control" name="pg_isi" id="pg_isi"></textarea> </div> <div class="form-group"> <button name="terima" class="btn btn-sm btn-success pull-right"><i class="fa fa- flat fa-fw fa-envelope"></i> Kirim</button> </div> </form> <?php } else { ?> <center><h4>::: Terimakasih Atas Partisipasi Anda :::</h4></center> <?php }} ?> </div> </div> </div> <?php include_once("__sidebar_2.php"); ?> <script src="<?php echo \$dir; ?>assets/plugin/chosen/chosen.jquery.js" type="text/javascript"></script> <script> \$(document).ready(function () { \$('#kategori').chosen(); var conditionalSelect = \$('#prodi'); var options = \$('##prodi').children(".conditional").clone(); \$('##prodi').children(".conditional").remove(); \$('##fakultas').change(function(){ var value = \$(this).val(); conditionalSelect.children(".conditional").remov e(); </pre>	<pre> <th>Kategori</th> <th class="text- center">Create</th> <th style="width: 130px;"></th> </tr> </thead> <tbody> <?php \$query = mysql_query("SELECT pengaduan.*, prodi.*, kategori.* FROM pengaduan LEFT JOIN prodi ON (prodi.pr_id = pengaduan.pr_id) LEFT JOIN kategori on (pengaduan.kat_id = kategori.kat_id) ORDER BY pengaduan.pg_status ASC, pengaduan.pg_create DESC"); \$no = 1; if(mysql_num_rows(\$query) > 0){ while(\$data = mysql_fetch_object(\$query)){ ?> <tr <?php echo ((\$data->pg_status === '0') ? 'class="danger"' : ''); ?> > <td class="text- center"><?php echo \$no++; ?></td> <td class="text-center <?php echo ((\$data->pg_status === '0') ? 'text-danger' : 'text- success'); ?>"><?php echo \$data->pg_kode; ?></td> <td><a id="pengaduanlihat" data- id="<?php echo \$data->pg_id; ?>" data- toggle="modal" data-target="#modaldata" class="btn btn-xs btn-warning"><i class="fa fa- comment-o fa-flat fa-fw"></i> &nbsp; <?php echo \$data->pg_judul; ?></td> <td><?php echo \$data- >pg_tujuan; ?></td> <td><?php echo \$data- >kat_nama; ?></td> <td class="text- center"><?php echo date("d/m/Y", strtotime(\$data- >pg_create)); ?></td> </pre>
---	---

