

**RANCANG BANGUN SISTEM PENGADUAN MAHASISWA
TERHADAP PRODI BERBASIS *WEBSITE***

SKRIPSI

Diajukan Oleh

ILHAM ADZMI PINIM

NIM.160212099

**Mahasiswa Fakultas Tarbiyah dan Keguruan
Prodi Pendidikan Teknologi Informasi**



**FAKULTAS TARBIYAH DAN KEGURUAN
UNIVERSITAS ISLAM NEGERI AR-RANIRY
DARUSALAM-BANDA ACEH
2021 M/1443 H**

**RANCANG BANGUN SISTEM PENGADUAN MAHASISWA
TERHADAP PRODI BERBASIS *WEBSITE*.
SKRIPSI**

Diajukan Kepada Fakultas Tarbiyah dan Keguruan (FTK)
Universitas Islam Negeri Ar-Raniry Darussalam Banda Aceh
Sebagai Beban Studi Untuk Memperoleh Gelar Sarjana
Dalam Ilmu Pendidikan Teknologi Informasi

Oleh:

ILHAM ADZMI PINIM

NIM . 160212099

Mahasiswa Fakultas Tarbiyah dan Keguruan
Prodi Pendidikan Teknologi Informasi

Disetujui Oleh:

Pembimbing I

Pembimbing II


Yusran, M.Pd

NIP. 197106261997021003


Mira Maisura, M.Sc

NIDN. 2027058602

جامعة الرانيري

AR - RANIRY

**RANCANG BANGUN SISTEM PENGADUAN
MAHASISWA PTI TERHADAP PRODI PTI
BERBASIS *WEBSITE***

SKRIPSI

telah diuji oleh Panitia Ujian Munaqasyah Skripsi

Fakultas Tarbiyah dan Keguruan Uin Ar-Raniry dan dinyatakan Lulus

Serta di terima Sebagai Salah Satu Beban Studi Program Sarjana (S1)

Dalam Ilmu Pendidikan Teknologi Informasi

Pada hari dan tanggal

Rabu, 8 Desember 2021 M
4 Jumadil Awal 1443

Panitia Ujian Muqasyah Skripsi

Ketua

Yusran, M.Pd
NIP.197106261997021003

Penguji I

Mira Maisura, M.Sc.
NIP.198605272019032011

Sekretaris

Nurul Fajri

Penguji II

Basrul, M.S
NIDN. 2027038701

Mengetahui,

Dekan Fakultas Tarbiyah Dan Keguruan UIN Ar-Raniry
Durussalam Banda Aceh



Dr. Muslimah Bazali, S. H. Ag
NIP. 195907091989031001

LEMBAR PERNYATAAN KEASLIAN KARYA ILMIAH/SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Ilham Adzmi Pinim

NIM : 160212099

Prodi : Pendidikan Teknologi informasi

Fakultas : Tarbiyah dan Keguruan

Judul Skripsi : Rancang Bangun Sistem Pengaduan Mahasiswa Terhadap Prodi Berbasis *Website*.

Dengan ini menyatakan bahwa dalam penulisan skripsi ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggung jawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggung jawab atas karya ini.

Bila di kemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggung jawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Tarbiyah dan Keguruan UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun

Banda Aceh, Desember 2021

Yang Menyatakan,



Ilham Adzmi Pinim

NIM.160212099

ABSTRAK

Nama : Ilham Adzmi Pinim
NIM : 160212099
Fakultas/Prodi : Tarbiyah dan Keguruan / Pendidikan Teknologi Informasi
Judul : Rancang Bangun Sistem Pengaduan Mahasiswa Terhadap Prodi Berbasis *Website*
Tebal Skripsi : 95 Lembar
Pembimbing I : Yusran, M.Pd
Pembimbing II : Mira Maisura, M.Sc
Kata Kunci : Sistem, Pengaduan mahasiswa, *Website*

Penelitian ini dilakukan untuk merancang Sistem pengaduan mahasiswa terhadap prodi berbasis *website* terhadap mahasiswa dan petugas prodi Pendidikan Teknologi Informasi. Penelitian ini menggunakan metode *Research and Development (R&D)* dengan model pengembangan RAD yaitu: Proses persyaratan, proses desain dan proses implementasi. Adapun teknik pengumpulan data dilakukan dengan studi pustaka, observasi dan wawancara. Penelitian ini sudah menghasilkan perancangan Sistem Pengaduan Mahasiswa Terhadap Prodi Berbasis *Website*. Produk juga sudah divalidasi oleh para pakar ahli media. Sistem ini mendapat respon positif dari ahli media yaitu 92% yang menyatakan setuju. Hal ini membuktikan bahwa sistem pengaduan ini layak digunakan untuk kalangan mahasiswa Pendidikan Teknologi Informasi sebagai salah satu media pengaduan.

جامعة الرانيري
A R - R A N I R Y

KATA PENGANTAR



Alhamdulillah, puji serta syukur penulis panjatkan kehadiran Allah SWT, Yang sudah memberikan kesehatan serta kekuatan sehingga penulis bisa menyelesaikan skripsi ini. Shalawat beriringkan salam kepada Nabi Muhammad SAW, yang sudah membawa umat manusia dari alam jahiliyah kepada alam yang penuh dengan ilmu pengetahuan seperti saat ini. Adapun Judul skripsi di Penelitian ini adalah ***“Rancang Bangun Sistem Pengaduan Mahasiswa Terhadap Prodi Berbasis Website”***.

Skripsi ini ialah tahap akhir penulis untuk memperoleh gelar Sarjana Pendidikan, pada Fakultas Tarbiyah dan Keguruan Program Studi Pendidikan Teknologi Informasi di Universitas Islam Negeri Ar-Raniry. Dalam usaha penyusunan skripsi ini, penulis banyak sekali menghadapi kesulitan dalam teknik penulisan maupun penguasaan bahan. Walaupun demikian, penulis tidak putus asa dalam berusaha dan berdoa. Dengan adanya dukungan dari berbagai pihak, oleh karena itu pada kesempatan ini penulis mengucapkan ribuan terima kasih kepada:

1. Ayahanda M. Fuad dan Ibunda Tercinta Sumiati dan juga semua keluarga yang selalu mengirimkan doa untuk penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Yusran, M.Pd sebagai Ketua Program Studi Pendidikan Teknologi Informasi Fakultas Tarbiyah Dan Keguruan UIN Ar-Raniry, serta sebagai

pembimbing I yang sudah banyak membantu dalam penulisan dan pemahaman materi.

3. Kepada Ibu Mira Maisura, M.Sc sebagai pembimbing II yang sudah meluangkan waktu dan tenaga untuk membimbing didalam pembuatan skripsi ini
4. Bapak/Ibu Dosen Program Studi Pendidikan Teknologi Informasi yang mengajarkan, mendidik serta membagi ilmu pengetahuan selama ini.
5. Terimakasih kepada teman-teman dan sahabat yang terus memberikan motivasi dan semangat kepada penulis dalam menyelesaikan skripsi ini.

Penulis berserah diri kepada Allah karena tidak ada yang terjadi tanpa kehendak-Nya. Segala usaha telah dilakukan untuk menyempurnakan skripsi ini. Namun, penulis juga sadar dalam penulisan skripsi ini masih banyak dijumpai kekurangan. Oleh sebab itu, penulis berharap saran yang dapat digunakan sebagai masukan guna perbaikan di masa yang akan datang. Semoga Allah SWT meridhoi penulisan ini dan selalu memberikan rahmat dan hidayah-Nya kepada kita semua. Aamiin.

Banda Aceh, Juli 2021

Penulis,

Ilham Adzmi Pinim

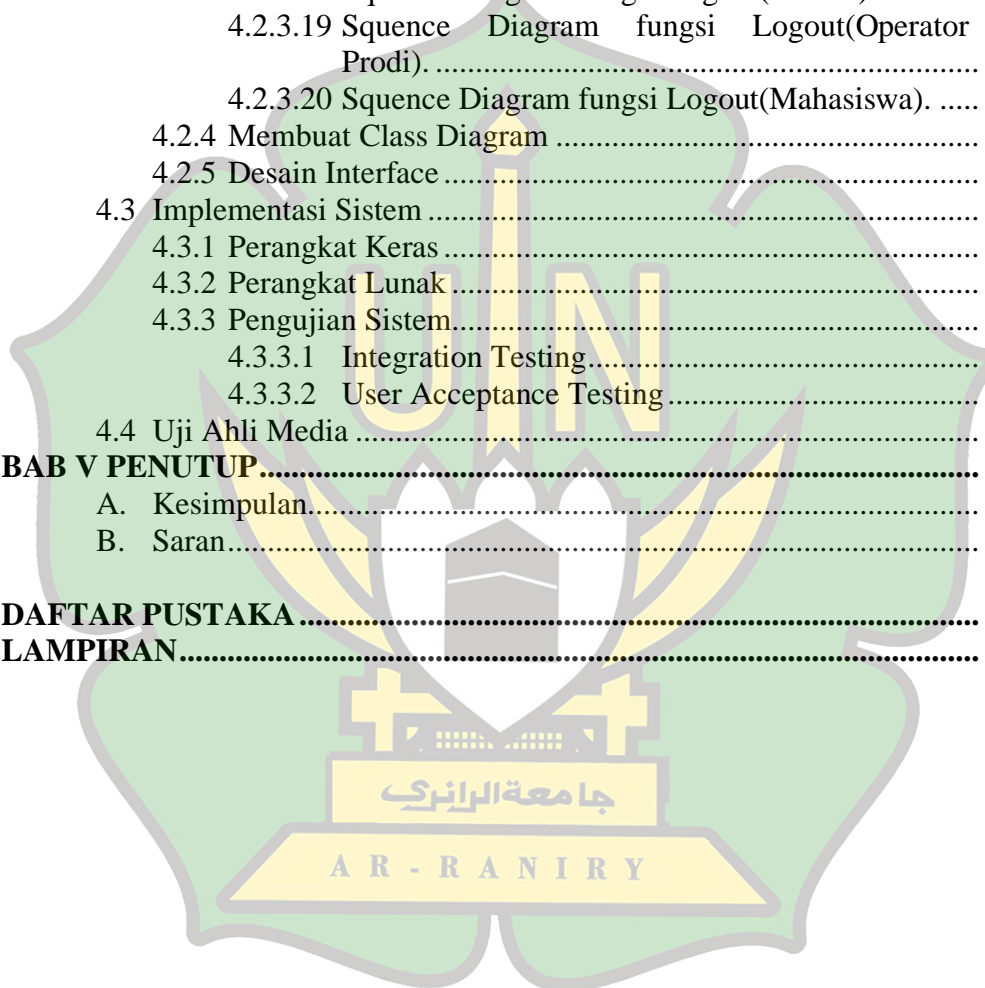
NIM. 160212099

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	
LEMBAR PENGESAHAN PENGUJI	
LEMBAR PERNYATAAN KEASLIAN KARYA ILMIAH/SKRIPSI	
ABSTRAK	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian/kegunaan penelitian	3
BAB II KAJIAN TEORI	4
2.1 Rancang Bangun	4
2.1.1 Rancang	4
2.1.2 Bangun.....	4
2.2 Arti sistem informasi	5
2.2.1 Sistem.....	5
2.2.2 Sistem Informasi.....	5
2.2.3 Website	6
2.3 Pengaduan (<i>E-Reporting</i>).....	6
2.4 Pemrograman	7
2.4.1 MySQL	7
2.4.2 Basis Data	7
2.4.3 PHP	9
2.4.4 HTML	9
2.4.5 CSS	10
2.4.6 DFD (Data Flow Diagram)	11
2.4.7 Unified Modeling Language (UML)	11
2.4.8 Flowchart	18
2.5 <i>Blackbox testing</i>	19
2.6 Penelitian terdahulu.....	20
2.6.1 Sistem pelayanan pengaduan masyarakat pada divis humas POLRI berbasis web.....	20
2.6.2 Rancang bangun sistem administrasi pelaporan masyarakat pada kantor polisi sektor Bontoala Kota Makassar.....	20
2.6.3 Perancangan Aplikasi Pengaduan Masyarakat Terhadap Lingkungan Di Tingkat Kelurahan.....	21

2.6.4 Analisa Dan Perancangan Sistem Pengaduan Mahasiswa Berbasis Web (Studi Kasus: Universitas Mercu Buana Kranggan)	21
BAB III METODOLOGI PENELITIAN	22
3.1 Metode Pengumpulan Data	22
2.6.5 Studi Pustaka.....	22
2.6.6 Studi Lapangan	22
2.6.7 Studi Literatur Sejenis	23
3.2 Metodologi Pengembangan Sistem.....	23
3.3 Kerangka Berfikir Penelitian.....	27
BAB IV HASIL DAN PEMBAHASAN.....	28
4.1 Perencanaan syarat-syarat (<i>Requirement Planning</i>).	28
4.1.1 Analisa sistem yang sedang berjalan	28
4.1.2 Identifikasi Masalah.....	28
4.1.3 Analisa kebutuhan sistem	29
4.1.4 Tujuan Pengembangan Sistem.....	29
4.1.5 Analisa proses sistem yang di usulkan	30
4.2 Proses Desain (<i>Workshop</i> Desain RAD).....	31
4.2.1 Membuat Use Case Diagram	31
4.2.1.1 Use Case Scenario	32
4.2.2 Membuat Activity Diagram.....	37
4.2.3 Membuat Squence Diagram.....	46
4.2.3.1 Squence Diagram fungsi login (Admin).....	46
4.2.3.2 Squence Diagram fungsi login (Operator Prodi)...	47
4.2.3.3 Squence Diagram fungsi login (Mahasiswa).....	48
4.2.3.4 Squence Diagram fungsi lihat dan hapus akun mahasiswa(Admin).....	49
4.2.3.5 Squence Diagram fungsi tambah dan edit akun mahasiswa(Admin).....	50
4.2.3.6 Squence Diagram fungsi lihat dan hapus data pengaduan(Admin).....	51
4.2.3.7 Squence Diagram fungsi tanggapi pengaduan mahasiswa(Admin).....	52
4.2.3.8 Squence Diagram fungsi lihat dan hapus tanggapan(Admin).....	53
4.2.3.9 Squence Diagram fungsi lihat dan hapus akun petugas di sistem(Admin).....	54
4.2.3.10 Squence Diagram fungsi tambah dan edit akun petugas(Admin).....	55
4.2.3.11 Squence Diagram fungsi lihat laporan(Admin)....	56
4.2.3.12 Squence Diagram fungsi cetak laporan(Admin) ..	57
4.2.3.13 Squence Diagram fungsi lihat dan hapus pengaduan(Operator Prodi).	58

4.2.3.14	Squence Diagram fungsi memberikan tanggapan terhadap pengaduan mahasiswa(Operator Prodi)..	59
4.2.3.15	Squence Diagram fungsi lihat dan hapus tanggapan(Operator Prodi).	60
4.2.3.16	Squence Diagram fungsi Tambah Laporan(Mahasiswa).	61
4.2.3.17	Squence Diagram fungsi lihat dan hapus laporan(Mahasiswa).....	62
4.2.3.18	Squence Diagram fungsi Logout(Admin).	63
4.2.3.19	Squence Diagram fungsi Logout(Operator Prodi).	64
4.2.3.20	Squence Diagram fungsi Logout(Mahasiswa).	64
4.2.4	Membuat Class Diagram	65
4.2.5	Desain Interface	68
4.3	Implementasi Sistem	70
4.3.1	Perangkat Keras	70
4.3.2	Perangkat Lunak	71
4.3.3	Pengujian Sistem.....	71
4.3.3.1	Integration Testing.....	72
4.3.3.2	User Acceptance Testing.....	75
4.4	Uji Ahli Media	76
BAB V	PENUTUP.....	79
A.	Kesimpulan.....	79
B.	Saran.....	79
DAFTAR PUSTAKA	81
LAMPIRAN	83



DAFTAR GAMBAR

Gambar 2.1	DBMS	9
Gambar 3.1	Pengembangan Sistem Model RAD.....	25
Gambar 3.2	Kerangka berfikir penelitian.....	27
Gambar 4.1	Sistem yang sedang berjalan	28
Gambar 4.2	Proses sistem yang di usulkan	30
Gambar 4.3	Activity Diagram fungsi Login (Admin, Petugas Prodi dan Mahasiswa).....	37
Gambar 4.4	Activity Diagram fungsi Register,hapus dan edit akun Mahasiswa(Admin)	38
Gambar 4.5	Activity Diagram Lihat dan tanggapi daftar pengaduan dari Mahasiswa (Admin dan Operator Prodi).....	39
Gambar 4.6	Activity Diagram Lihat dan tanggapi daftar pengaduan dari Mahasiswa (Admin dan Operator Prodi).....	40
Gambar 4.7	Activity Diagram Lihat, hapus, dan tambah akun admin atau operator prodi(Admin).....	41
Gambar 4.8	Activity Diagram Lihat dan cetak laporan pengaduan (Admin)	42
Gambar 4.9	Activity Diagram input pengaduan (Mahasiswa).....	43
Gambar 4.10	Activity Diagram Lihat daftar laporan yang telah di tanggapi(Mahasiswa).....	44
Gambar 4.11	Activity Diagram Logout (Admin, Operator Prodi dan Mahasiswa).....	45
Gambar 4.12	Squence Diagram fungsi login (Admin).....	46
Gambar 4.13	Squence Diagram fungsi login (Admin).....	47
Gambar 4.14	Squence Diagram fungsi login (Mahasiswa).....	48
Gambar 4.15	Squence Diagram fungsi lihat dan hapus akun mahasiswa(Admin).....	49
Gambar 4.16	Squence Diagram fungsi tambah dan edit akun mahasiswa(Admin).....	50
Gambar 4.17	Squence Diagram fungsi lihat dan hapus data pengaduan(Admin).....	51
Gambar 4.18	Squence Diagram fungsi tanggapi pengaduan mahasiswa(Admin).....	52
Gambar 4.19	Squence Diagram fungsi lihat dan hapus tanggapan(Admin)..	53
Gambar 4.20	Squence Diagram fungsi lihat dan hapus akun petugas di sistem(Admin)	54
Gambar 4.21	Squence Diagram fungsi tambah dan edit akun petugas(Admin).	55
Gambar 4.22	Squence Diagram fungsi lihat laporan(Admin).....	56
Gambar 4.23	Squence Diagram fungsi cetak laporan(Admin).	57
Gambar 4.24	Squence Diagram fungsi lihat dan hapus pengaduan(Operator Prodi).	58

Gambar 4.25	Squence Diagram fungsi memberikan tanggapan terhadap pengaduan mahasiswa(Operator Prodi).....	59
Gambar 4.26	Squence Diagram fungsi lihat dan hapus tanggapan(Operator Prodi).....	60
Gambar 4.27	Squence Diagram fungsi Tambah Laporan(Mahasiswa).	61
Gambar 4.28	Squence Diagram fungsi lihat dan hapus laporan(Mahasiswa).	62
Gambar 4.29	Squence Diagram fungsi Logout(Admin).	63
Gambar 4.30	Squence Diagram fungsi Logout(Operator Prodi).	64
Gambar 4.31	Squence Diagram fungsi Logout(Mahasiswa).	64
Gambar 4.32	Class Diagram.	65
Gambar 4.33	Tampilan login (admin, Operator, mahasiswa)	68
Gambar 4.34	Tampilan beranda admin	69
Gambar 4.35	Tampilan beranda operator prodi	69
Gambar 4.36	Tampilan beranda untuk mahasiswa	70



DAFTAR TABEL

Tabel 2.1	Simbol DFD	11
Tabel 2.2	Simbol <i>Flowchart</i>	19
Tabel 3.1	Tabel Uji Ahli Media	27
Tabel 4.1	<i>Actor Use Case</i>	31
Tabel 4.2	Tabel interaksi actor di sistem	32
Tabel 4.3	<i>Use Case Scenario Login</i>	32
Tabel 4.4	<i>Use Case Scenario Register</i>	33
Tabel 4.5	<i>Use Case Scenario</i> lihat data pelanggan.....	34
Tabel 4.6	<i>Use Case Scenario</i> lihat data Respon	34
Tabel 4.7	<i>Use Case Scenario input</i> /lihat dan update data user.....	35
Tabel 4.8	<i>Use Case Scenario</i> Lihat dan cetak laporan pengaduan	35
Tabel 4.9	<i>Use Case Scenario</i> Input Pengaduan	36
Tabel 4.10	<i>Use Case Scenario</i> lihat daftar laporan yang telah di tanggap ..	36
Tabel 4.11	<i>Use Case Scenario</i> Logout	37
Tabel 4.12	Tabel Petugas	66
Tabel 4.13	Tabel mahasiswa	66
Tabel 4.14	Tabel pengaduan	67
Tabel 4.15	Tabel tanggapan	67
Tabel 4.16	Tabel Integration testing (Mahasiswa)	72
Tabel 4.17	Tabel Integration testing (Admin).....	74
Tabel 4.18	<i>Integration sistem</i> (Operator Prodi).....	75
Tabel 4.19	<i>User acceptance testing</i>	75
Tabel 4.20	Tabel Uji media 1	76
Tabel 4.21	Tabel Uji media 2	77
Tabel 4.22	Tabel uji media 3	77

جامعة الرانيري

A R - R A N I R Y

DAFTAR LAMPIRAN

Lampiran 1 Dokumentasi Penelitian.....	83
Lampiran 2 Kuisisioner Ahli Media	85
Lampiran 3 Kuisisioner Integration Test.....	88
Lampiran 4 Tangkapan Layar tampilan website yang telah di buat.	92



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan dan kebutuhan terhadap situs *web* saat ini sangat berperan penting di dalam segala aspek yang berlaku, di antaranya seperti *Facebook* yang merupakan situs *web* yang menjembatani semua orang di seluruh dunia melalui *internet*, ada juga *website* yang bisa melakukan tatap muka secara jarak jauh seperti *Zoom*, *Hangout*, dll. Banyak juga *website* yang menyajikan berbagai macam informasi di seluruh dunia seperti *Detik.com* yang menyajikan tentang berita terkini Indonesia, ada juga *website* yang di gunakan sebagai alat *marketing* atau berjualan seperti *Shopee.com* dan *Tokopedia*. Di segala aspek *website* sangat di butuhkan saat ini, Termasuk di dalam dunia perkuliahan di Universitas, di Universitas pada umumnya memiliki berbagai macam websitenya termasuk di dalam sistem pengaduannya, sistem pengaduan di Universitas pada saat ini menggunakan cara manual dan juga memakai pengaduan dengan berbasis teknologi seperti melalui *Website* yang terhubung langsung ke Universitas.

Pada Universitas Islam Negeri Ar-Raniry memakai sistem pengaduan secara manual dan juga sudah ada sistem pengaduan berbasis *Website* juga, seperti *SIAKAD* dan juga ada *web* khusus pelaporan mahasiswa terhadap Universitas di laman <https://pengaduan.ar-raniry.ac.id>. Tetapi untuk sistem pengaduan di setiap prodinya masih menggunakan sistem manual terkhususnya di prodi Pendidikan Teknologi Informasi.

Prodi Pendidikan Teknologi Informasi adalah sebuah prodi yang berada di fakultas Tarbiyah dan Keguruan, di Universitas Islam Negeri Ar-Raniry Banda Aceh, mahasiswa dari prodi ini berasal dari beberapa daerah dan juga di kala pandemi saat ini kebanyakan mahasiswanya di rumahkan dan tidak bisa berhadir di kampus jika ada masalah akademik maupun non akademik, agar proses pengaduan masalah dari mahasiswa ke bagian staf prodi bisa lancar, cepat dan mudah di butuhkan sebuah *website* yang bisa menjembatannya.

Contoh beberapa kasus yang penulis amati dari respon beberapa mahasiswa yaitu sulitnya untuk menghubungi pihak prodi untuk melakukan pelaporan terkendala jarak dan waktu yang tidak tepat, seperti jika mahasiswa ingin melaporkan masalah dari rumah atau kampung halaman ke kampus itu cukup sulit jika harus manual, serta untuk pengarsipan laporan yang masuk ke prodi masih bersifat manual.

Maka dari latar belakang itu pada penelitian kali ini peneliti akan membahas mengenai **“Rancang Bangun Sistem Pengaduan Mahasiswa terhadap Prodi PTI berbasis *Website*.”** Di harapkan dengan pembuatan aplikasi *web* ini bisa memudahkan interaksi pengaduan mahasiswa PTI terhadap prodi dengan akses yang lebih mudah dan efektif.

1.2 Rumusan Masalah

Bagaimana merancang dan mengembangkan sistem pengaduan pada Prodi Pendidikan Teknologi Informasi berbasis *Website*?

1.3 Batasan Masalah

Supaya penelitian ini berjalan dengan semestinya dan lebih terarah, maka ada perlu pembatasan masalah, yaitu :

- a. Dalam tugas akhir ini pembuatan aplikasi *web* pengaduan hanya berlingkup di Prodi Pendidikan teknologi Informasi
- b. PHP adalah bahasa pemograman yang akan di pakai
- c. Untuk dibagian database atau penyimpanan data akan menggunakan aplikasi *MySQL* .

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian kali ini, antara lain:

- a. Membuat aplikasi *web* Pengaduan Mahasiswa PTI terhadap Prodi
- b. Memudahkan proses pengolahan masalah mahasiswa oleh staf Prodi
- c. Memudahkan proses pengarsipan masalah mahasiswa terhadap Prodi.

1.5 Manfaat Penelitian/kegunaan penelitian

Manfaat penelitian kali ini di antaranya :

- a) Bagi penulis, sebagai ajang pembelajaran lebih luas lagi di dalam pembuatan aplikasi berbasis website
- b) Bagi staf prodi, memudahkan di dalam proses penyelesaian laporan mahasiswa
- c) Bagi mahasiswa, memudahkan di dalam pengaduan masalah ke bagain staf Prodi.

BAB II

KAJIAN TEORI

2.1 Rancang Bangun

2.1.1 Rancang

Perancangan adalah tahapan persiapan dari membangun sebuah sistem dengan lebih teratur dan terarah, di dalam perancangan terdapat banyak tahapan di antaranya adalah di mulai dengan pembuatan konsep website yang di buat, bagaimana desain website yang di inginkan dan bagaimana cara kerja dari website tersebut. Perancangan ditujukan sebagai penjelasan bagaimana website ini akan di bangun dengan segala komponen dan struktur di dalmnya sehingga website yang akan di bangun sesuai dengan keinginan awal[1]. Perancangan sistem adalah bagaimana pemilihan tahapan atau langkah-langkah yang akan di ambil dalam sebuah perancangan pembangunan sebuah sistem yang baru, tujuan dari perancangan sistem adalah sebagai bentuk pemenuhan kebutuhan website yang akan di bangun[2]. Perancangan sistem memerlukan sebuah kemampuan merancang di antaranya adalah pemilihan alat-alat yang akan membantu di dalam pembangunan sistem tersebut[3].

2.1.2 Bangun

Bangun atau pembangunan sistem ialah sebuah proses penciptaan sebuah sistem yang baru ataupun memperbaiki dan mengganti sistem yang sebelumnya telah ada baik secara sebagian ataupun keseluruhannya. Rancang Bangun ialah penggabungan antara perncanaaan, penggambaran, serta pembuatan gambaran

awal dan settingan dari beberapa bagian yang dipisah kedalam suatu kesatuan yang berfungsi dan utuh[4].

2.2 Arti sistem informasi

2.2.1 Sistem

Sistem itu adalah sebuah kegiatan pengamatan sistem yang telah berjalan, memiliki tujuan untuk meniali sejauh mana tingkat keefektifitasan sistem tersebut, serta setelahnya mendata fitur-fitur yang sekiranya perlu di buat untuk sistem yang akan di buat[5]. Sistem terdapat dua kelompok pendekatan, yaitu kelompok yang menegaskan pada bagian prosedur, serta menegaskan pada komponennya. Pendekatan ini menegaskan tahapan penggambaran sistem untuk sebuah jaringan kerja prosedur yang berkaitan, bersama, berkumpul bertujuan menjalankan langkah ataupun untuk memecahkan suatu masalah tertentu. Sedangkan, pendekatan sistem yang menegaskan di bagian atau komponen menggambarkan sistem berarti elemen yang berkegiatan untuk mencapai sebuah tujuan khusus[6].

2.2.2 Sistem Informasi

Sistem informasi ialah sekumpulan instruksi atau perintah yang disusun secara sistematis dan teratur, diperlukan suatu inputan, setelahnya akan menghasilkan suatu keluaran yang dapat membantu di dalam pengambilan keputusan sebuah manajemen atau operasi dalam organisasi ataupun perusahaan[7]. Sistem informasi yang berbasis komputer biasanya di artikan dengan sistem informasi. Sistem informasi melingkupi beberapa komponen di antaranya, komputer, teknologi informasi, manusia dan tahapan kerja[8]. Sistem

Informasi adalah sebuah rangkaian informasi yang terdapat di dalamnya beberapa bagian yang berkaitan dan bergantung satu sama lain. Hubungan dalam sistem informasi tersebut, adalah arus jalannya sebuah informasi dari beberapa tingkat sebuah organisasi untuk mendapatkan informasi yang berguna bagi semua pihak.

2.2.3 Website

Website ialah suatu beberapa halaman pada sebuah alamat *website* di dalam internet ditujukan untuk tujuan khusus serta berkaitan satu sama lain secara meluas dan bisa di akses siapa saja melalui *homepage* atau halaman depan dengan menggunakan sebuah *browser* memakai *URL Website*. *Website* ialah seluruh halaman *website* yang tersedia dari sebuah domain yang menampung informasi[5]. *Website* ialah sekumpulan halaman-halaman *web* yang saling terhubung dan terdapat informasi di dalam halaman tersebut dan terhubung dengan *internet*.

2.3 Pengaduan (E-Reporting)

Sistem pelaporan secara manual dapat memunculkan kesalahan (*error*) serta butuh waktu yang lama (*time-consuming*) di dalam pengumpulannya. Setiap dari data laporan yang dibutuhkan akan di masukan ulang secara manual kedalam komputer dan basis data sehingga dapat berdampak terhadap keterlambatan penyampaian informasi dengan cepat[6]. *Electronic reporting* juga akan memberikan kemampuan serta fleksibilitas bagi lembaga terkait di dalam pengumpulan data yang dibutuhkan, menurunkan persentase kesalahan yang di timbulkan oleh penginputan ulang data secara manual, serta secara signifikan bisa mempersingkat waktu yang dapat dipergunakan untuk proses serta penyajian data.

E-Reporting atau juga dikenal dengan sistem pelaporan juga penyajian data secara elektronik adalah pola penyampaian data elektronik kepada public melalui akses internet[6].

2.4 Pemrograman

2.4.1 MySQL

MySQL merupakan suatu program basis data *client server* yang berbasis console, berupa teks/kode. *MySQL* bersifat *open source* yang menerapkan *SQL* sebagai bahasa untuk mengakses *database* di dalamnya. Standar penting untuk bahasa basisdata ialah *Structured Query Language (SQL)*. *SQL* dijadikan standar sebagai bahasa untuk membuat *database*, menyimpan informasi atau data ke dalam *database*, serta memanggil kembali informasi darinya. Aplikasi di khusukan untuk menampilkan data *SQL*. Seorang programmer akan mengawali dengan membuat sebuah struktur di dalam *SQL* dan setelahnya melanjutkan dengan menuli kode *PHP* untuk menjangkau data tersebut. *PHP* kemudian menerima inputan itu dan merubah isi *database* dan menampilkannya kembali[7].

2.4.2 Basis Data

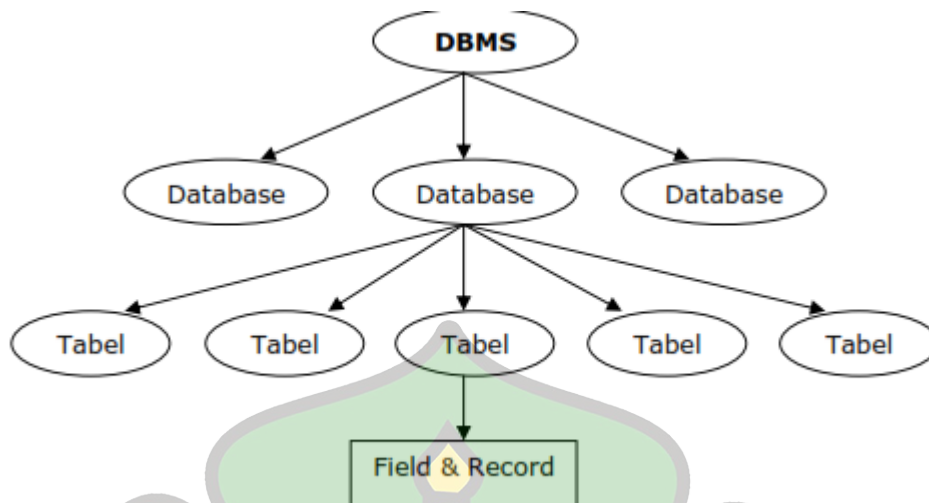
Basis data adalah sekumpulan informasi yang disimpan dalam komputer secara teratur dan tersusun sehingga bisa diperiksa memakai sebuah program komputer guna mendapatkan informasi dari basis data tersebut. Basis Data dipakai untuk penyimpanan informasi dan data yang terhubung satu dengan yang lain dengan baik di dalam sebuah komputer. Untuk pengelolaan basis data dibutuhkan sebuah perangkat lunak yaitu *DBMS (Database Management System)*.

DBMS adalah sebuah sistem perangkat lunak yang memfasilitasi penggunaanya untuk membuat, memelihara, mengontrol, dan mengakses basis data dengan praktis dan mudah. Dengan DBMS, pengguna akan lebih mudah melakukan pengontrolan dan mengubah data yang dibuat. Sedangkan RDBMS atau *Relationship Database Management System* adalah salah satu jenis DBMS yang memfasilitasi adanya hubungan atau *relationship* antar satu tabel dengan tabel lainnya. Selain RDBMS, ada juga jenis DBMS yang lain, seperti *Object Oriented DBMS*, *Hierarchy DBMS*, dsb. Jenis perangkat lunak DBMS yang lazim digunakan dalam aplikasi program di antaranya [9] :

- *PostgreSQL*
- *Microsoft SQL Server*
- *MySQL*
- *Teradata*
- *Interbase*
- *Sybase*
- *Firebird*
- *Oracle*
- *DB2*

Dalam konsep *database*, urutan atau hirarki *database* sangatlah penting.

Urutan atau hirarki *database* digambarkan dalam gambar sbb [9] :



Gambar 2.1 DBMS

2.4.3 PHP

Personal Home Page atau disingkat PHP merupakan salah satu bahasa pemrograman web atau *scripting language* yang didesain untuk *web*. PHP diciptakan pertama kali oleh seseorang yang bernama Rasmus Lerdorf, yang tujuan awalnya diciptakan penghitung jumlah pengunjung di halaman *web*nya. Bahasa pemrograman PHP bisa digabungkan dengan HTML dengan syarat terlebih dahulu memberikan tanda *tag* buka dilanjutkan tanda tanya (<?) setelahnya ditutup dengan tanda tanya dilanjutkan tanda *tag* tutup (?>) [10].

2.4.4 HTML

Hypertext Markup Language atau sering dikenal dengan singkatan HTML merupakan bahasa dasar bagi *web scripting* bersifat *client side* yang dapat menampilkan informasi dalam media teks, grafik, serta *multimedia* dan juga untuk menjadi penghubung antartampilan halaman *web(hyperlink)*. Konsep Dasar HTML pemrograman atau yang sering disebut *web scripting*. Disebut *script*




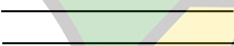
dikarenakan perintah kode program tersebut akan di *interpreter* serta tidak ada kompilasi untuk menjadikannya *executable*. Ada 2 jenis *web scripting* berdasarkan penempatan *interpreter* nya, yaitu yang bersifat *client side* serta *server side*. *Client side* dilakukan oleh *web browser* seperti *Internet Explorer*, *Netscape*, *Opera*, dan *Firefox*. Sebagai contoh bahasa *client side* ialah *HTML*, *CSS*, *Javascript*, *VBscript*, dan *XML*. Dan di bagian *server side* dilakukan oleh *web server* seperti *PWS (Personal Web Server untuk Sistem Operasi Windows 98)*, *IIS (untuk Sistem Operasi Windows 2000/Windows XP)*, *Apache*, *Tomcat*, *Xitami*, dan *ZOPE*. Untuk contoh bahasa *server side* adalah *ASP (.Net)*, *PHP*, *JSP*, *CFM*, dan *CGI/PL*[11].

2.4.5 CSS

Cascading Style Sheet atau sering di singkat *CSS* merupakan sebuah bahasa *stylesheet* yang dipakai untuk pengaturan tampilan sebuah *website*, baik tata letaknya, jenis huruf, warna, serta semua yang berkaitan dengan tampilan. Biasanya *CSS* digunakan untuk mendesain halaman *web* yang ditulis dengan *HTML* maupun *XHTML*. Ada dua jenis cara yang bisa dipakai untuk menggunakan *CSS* pada *web*. Cara yang pertama ialah dengan menuliskan *CSS* langsung didalam satu file *HTML* yang di bhat (*internal / inline style sheet*). Sedangkan yang kedua ialah dengan cara memanggil *CSS* tersebut dari file *CSS* tersendiri (*external style sheet*)[12].

2.4.6 DFD (Data Flow Diagram)

Data Flow Diagram atau DFD merupakan sebuah gambaran dari arus sistem yang telah ada atau sistem yang baru akan dibuat yang kemudian dikembangkan secara logika tanpa melihat lingkungan fisik dimana data tersebut akan mengalir. DFD merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh professional sistem kepada pemakai maupun pembuat program. Adapun simbol yang digunakan untuk membuat DFD yaitu[4]:

Simbol	Nama	Keterangan
	Eksternal <i>entity</i>	Menunjukkan sumber data masukan dan keluaran data dan juga menunjukkan bagian luar sistem
	Proses	Menunjukkan alur pengerjaan proses informasi dalam batas sistem
	Item data	Menunjukkan aliran proses dari data
	Data <i>Storage</i>	Digunakan untuk penyimpanan sebuah arus sistem data berbentuk file sebagai arsip,dll.


Tabel 2.1 Simbol DFD

2.4.7 Unified Modeling Language (UML)

Unified Modelling Language(UML)adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti/lunak.UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat

membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti *C++*, *Java*, *C#* atau *VB.NET*. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: *Grady Booch OOD (Object Oriented Design)*, *Jim Rumbaugh OMT (Object Modeling Technique)*, *Ivar Jacobson OOSE (Object-Oriented Software Engineering)*[13].

1. *UseCase scenario* 

UseCase scenario merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan urutan aksi/langkah yang dilakukan aktor ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal. *Usecase scenario* dijelaskan secara tekstual dalam beberapa format tergantung kebutuhannya, yaitu singkat (*brief*), informal (*casual*), atau lengkap (*fully dressed*), yang bisa dijelaskan dalam bentuk tabel dengan 1 kolom atau 2 kolom. Pada format singkat, penjelasan diberikan cukup paragraf yang mengacu

hanya pada skenario yang berhasil. Pada format informal, penjelasan diberikan dalam beberapa paragraf yang mencakup semua skenario, baik yang berhasil maupun gagal. Sedangkan, pada format lengkap, penjelasan dibuat secara detil disertai dengan bagian-bagian pendukung yang penting. Format terakhir ini yang banyak digunakan di dalam praktik. Bagian-bagian penting tersebut adalah:

- Aktor primer (*primary actor*), yaitu aktor yang menginisiasi layanan sistem untuk mencapai tujuan dari aktor tersebut. Jumlah aktor primer dimungkinkan lebih dari 1.
- Prakondisi (*preconditions*), yaitu kondisi spesifik yang harus terpenuhi sebelum sebuah *Use Case* bisa diinisiasi atau dieksekusi oleh aktor primer. Jumlah prakondisi bisa lebih dari 1 keadaan.
- Alur utama (*main or basic flow*), yaitu jalur interaksi yang mengarahkan pada skenario yang berhasil sehingga tujuan aktor bisa terpenuhi. Jalur ini hanya terdiri dari 1 jalur saja.
- Alur alternatif (*alternative flows*), yaitu jalur alternatif dari interaksi yang terjadi antar aktor dengan sistem yang mencakup pencabangan (pilihan) maupun skenario yang gagal sehingga tujuan aktor tidak terpenuhi. Jalur ini bisa terdiri dari lebih dari 1 jalur kemungkinan.
- Kondisi akhir (*post conditions*), yaitu kondisi spesifik yang harus terjadi ketika *Use Cas* berhasil dijalankan atau dieksekusi secara lengkap, sebagai representasi dari tujuan yang ingin dicapai oleh aktor primer. Jumlah kondisi akhir bisa lebih dari 1 keadaan[14].

2. Usecase Diagram

Sebuah *Usecase* diagram menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem. Diagram ini bisa menjadi gambaran yang bagus untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sistem.

Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan *Usecase*. Aktor adalah segala sesuatu yang berinteraksi langsung dengan sistem, bisa merupakan orang (yang ditunjukkan dengan perannya dan bukan namanya/personalnya) atau sistem komputer yang lain.. *Usecase* dinotasikan dengan simbol elips dengan nama kata kerja aktif di bagian dalam yang menyatakan aktivitas dari perspektif aktor. Setiap aktor dimungkinkan untuk berinteraksi dengan sistem dalam banyak *Usecase*. Sebaliknya, setiap *Usecase* bisa dijalankan oleh lebih dari satu aktor.

Antar aktor maupun antar *Usecase* bisa memiliki relasi, masing-masing dengan spesifikasi yang berbeda. Sebuah *Usecase*, disebut dengan *base Usecase*, bisa memiliki relasi dengan 1 atau lebih *Usecase* yang lain, disebut dengan *supplier Usecase*, dalam bentuk *extend* dan *include*. Relasi *extend* menyatakan bahwa fungsionalitas dari *base Usecase* bisa diperluas oleh *supplier Usecase*, jika dibutuhkan, di dalam aksekusi alur alternatif yang ada pada *Usecase scenario* dari *base Usecase*. Sedangkan, relasi *include* menyatakan bahwa fungsionalitas dari *base Usecase* selalu hanya bisa dipenuhi dengan bantuan dari *supplier Usecase* di dalam eksekusi alur utama yang ada pada *Usecase scenario* dari *base Usecase*. Dalam hal ini, relasi *include* dan *extend* tidak menjelaskan urutan eksekusi apapun

antara *base Usecase* dan *supplier Usecase*, baik dalam alur utama maupun alternatif yang dijelaskan dalam *Usecase scenario* dari *base Usecase*. Selanjutnya, sebuah aktor, disebut aktor induk, bisa memiliki relasi inheritance dengan aktor yang lain, disebut aktor turunan, yang menyatakan bahwa sebuah aktor merupakan turunan dari aktor yang lain. Aktor turunan akan memiliki hak akses terhadap fungsionalitas sistem yang lebih luas dibandingkan dengan aktor induk [14].

3. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan

join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal [13].

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek/objek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut. Proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan ikon khusus untuk objek boundary, controller dan persistent entity [13].

5. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)

2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

Sesuai dengan perkembangan *class model*, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.





Hubungan Antar *Class* :



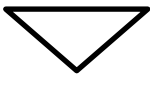

- Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.

- Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
- Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
- Hubungan dinamis, yaitu rangkaian pesan (message) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian [13].

2.4.8 Flowchart

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma. Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Berikut simbol-simbol flowchart yaitu[4]:

Simbol	Nama	Keterangan
	Terminal	Menyatakan sebagai sebuah awalan dan akhiran dari sebuah aliran proses
	Proses	Untuk menandakan suatu proses
	Input dan Output	Untuk menunjukkan bagian keluaran dan juga masukan di dalam suatu proses
	Storage Data	Menunjukkan sebuah informasi yang disimpan didalam sebuah

		penyimpanan data umum
	Operasi Manual	Menunjukkan sebuah proses yang tidak dilakukan dengan menggunakan komputer
	<i>Decision</i> atau Logika	Menunjukkan sebuah kondisi diman ada dua pilihan bisa jadi iya atau tidak
	<i>Offline Storage</i>	Menunjukkan bahwa data yang akan ada di simbol ini di simpan ke penyimpanan khusus.
	<i>Page connector</i>	Menunjukkan simbol sebagai penghubung antara satu halaman proses dengan halama lainnya

Tabel 2.2 Simbol Flowchart

2.5 *Blackbox testing*

Black-Box Testing merupakan pengujian perangkat lunak yang merupakan tes fungsionalitas dari aplikasi yang tidak mengacu pada struktur internal atau tidak membutuhkan pengetahuan khusus pada kode program aplikasi dan pengetahuan pengguna. Ujicoba *Black-Box* dilakukan untuk menemukan kesalahan dalam beberapa kategori yaitu:

1. Fungsi-fungsi yang hilang atau salah
2. Kesalahan desain antarmuka (*interface*) atau tampilan
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan performa dan
5. Kesalahan inisialisasi dan terminasi [15].

2.6 Penelitian terdahulu

Penelitian terdahulu menjadi salah satu acuan dalam melakukan penelitian sehingga terdapat banyak teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dalam membuat penelitian kami tidak menemukan penelitian yang sama judul seperti kami tapi kami mengangkat penelitian tersebut untuk referensi dalam memperkaya teori dalam penelitian ini.

2.6.1 Sistem pelayanan pengaduan masyarakat pada divis humas POLRI berbasis web

Pada penelitian ini terdapat 3 orang peneliti yaitu Fajar Masya, Elvina dan Fitria Maria Simanjuntak. Penelitian ini bertempat di Divisi Humas POLRI pada tahun 2012. Di penelitian ini mereka melakukan perancangan serta membangun sebuah sistem pengaduan berbasis web yang di tujukan untuk masyarakat ramai kepada kepolisian. Mereka menerapkan model perancangan dan pembangunannya menggunakan model UML, dan juga model *Waterfall*. Mereka memakai bahasa pemograman PHP untuk membangun aplikasinya dan juga memakai MySQL sebagai *databasenya*[16].

2.6.2 Rancang bangun sistem administrasi pelaporan masyarakat pada kantor polisi sektor Bontoala Kota Makassar.

Penelitian ini dilaksanakan oleh Saudara Sudianto di tahun 2018 bertempat di Kantor Polisi Sektor Bontoala Kota Makassar. Penelitian ini memakai pendekatan berdasar ilmu pengetahuan dan teknologi, dengan menggunakan sumber data melalui observasi langsung di Kantor Polisi Sektor

Bontola, dan juga memakai 2 metode yaitu observasi dan wawancara, analisis data yang di pakai adalah analisi data kualitatif, dan menggunakan metode perancangan sistem *Waterfall*[4].

2.6.3 Perancangan Aplikasi Pengaduan Masyarakat Terhadap Lingkungan Di Tingkat Kelurahan

Di penelitian ini ada 3 orang peneliti yaitu Imam Syafei, Mia Kamayani, Estu Sinduningrum. Penelitian ini bertempat di di Kelurahan Kukusan, Beji. Depok, Jawa Barat pada tahun 2019. Di penelitian ini mereka menggunakan metode pengembangan aplikasinya menggunakan metode *Extreme Programming* [17].

2.6.4 Analisa Dan Perancangan Sistem Pengaduan Mahasiswa Berbasis Web (Studi Kasus: Universitas Mercu Buana Kranggan)

Pada penelitian ini ada 2 orang ppeneliti yaitu Melati Naomi dan Handrie Noprisson, penelitian ini di lakukan di Universitas Mercu Buana Kranggan pada tahun 2019. Analisis data yang di gunakan di dalam penelitian ini adalah dengan metode PIECES, teknik pengumpulan data menggunakan observasi dan survei. Perancangan sistem nya menggunakan sistem UML.[6]

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Di dalam proses penelitian ini, dibutuhkan beberapa informasi dan data yang lengkap untuk bahan pendukung kebenaran pembahasan dan materi. Karena sebab itu, sebelum memulai menyusun skripsi ini dilakukan penelitian awal untuk mencari informasi yang bersangkutan. Ada 3 teknik untuk pengumpulan data yang dilakukan, diantaranya sebagai berikut :

2.6.5 Studi Pustaka

Peneliti melaksanakan studi pustaka dengan cara banyak membaca dan memahami beberapa buku sebagai sumber yang berhubungan dengan perancangan sistem, pemrograman *web*, dan beberapa referensi yang berkaitan dengan sistem pelaporan yang dibahas didalam skripsi ini.

2.6.6 Studi Lapangan

Pada studi lapangan dilakukan 2 metode yaitu dengan Observasi/pengamatan dan juga wawancara langsung dengan mahasiswa dan pihak prodi.

1. Observasi/ pengamatan

Observasi dilakukan untuk mengetahui secara langsung bagaimana prosedur pengaduan mahasiswa terhadap prodi yang sedang berjalan saat ini dan apa saja permasalahan yang kerap ada berhubungan dengan pengaduan

mahasiswa. Hal yang di perhatikan antara lain adalah bagaimana alur proses pengaduan mahasiswa terhadap prodi dan teknologi informasi yang di gunakan.

Observasi dilakukan pada :

Tempat : Kantor Prodi Pendidikan Teknologi Informasi, UIN Ar-Raniry, Banda Aceh.

Waktu : Senin, 7 Juni 2021

2. Interview/ Wawancara

Metode ini dilakkan dengan dengan cara mewawacarai langsung pihak mahasiswa dan juga pihak prodi, hal ini dilakukan supaya mendapat infromasi lebih lagi untuk penelitian ini baik dari subjek mahasiswa dan pihak prodi. Pada wawancara ini diajukan beberapa pertanyaan kepada pihak mahasiswa dan pihak prodi seperti bagaimana sistem yang sedang berjalan danapa saja masalah yang di hadapi ketika menggunakan sistem pengaduan yang biasa.

Wawancara dilakukan pada :

Tempat : Kantor Prodi Pendidikan Teknologi Informasi,
UIN Ar- Raniry, Banda Aceh.

Waktu : Senin, 7 Juni 2021

2.6.7 Studi Literatur Sejenis

Pada tahap ini peneliti membaca beberapa penelitian serupa untuk membantu referensi di dalam penelitian ini.

3.2 Metodologi Pengembangan Sistem

Metode yang dipakai didalam penelitian ini ialah dengan metode penelitian pengembangan (*Research & Development*).Metode penelitian dan

pengembangan *Research & Development* merupakan metode penelitian yang bisa digunakan guna menciptakan produk serta menguji keefektifan produk yang diciptakan[18].

Dipilihnya metode ini dikarenakan menurut peneliti metode ini cocok di pakai sesuai dengan judul yang akan dibahas yaitu perancangan sebuah produk yaitu sistem pelaporan Berbasis *Website*, di anggap cocok karna langkah-langkah di dalam metode ini mencakup langkah-langkah yang sesuai dengan perancangan sebuah sistem yaitu di mulai dengan tahap pembuatan konsep *web* nya seperti apa seterusnya bagaimana desain *website* itu sendiri dan setelah melakukan perancangan desain di lanjutkan dengan tahap pembuatan *website* dengan sistem peimplementasian koding, dan setelah *website* selesai maka akan di lakukan tahap *testing* untuk mengetahui kegunaan dari sistem yang dibangun. Dengan menggunakan metode R&D (*Research & Development*) dan metode pengembangan sistemnya yang menggunakan model RAD, maka penulis akan membangun suatu sistem pelaporan berbasis *web* di lingkungan prodi Pendidikan teknologi Informasi UIN Ar-Raniry.

1. Metode Model Pengembangan RAD

Rapid Application Development (RAD) merupakan metode pengembangan sistem informasi dengan waktu singkat, sehingga dinilai tepat digunakan dalam pembangunan sistem pengaduan berbasis *web* tersebut. RAD menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja), sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (requirement) pengguna dan selanjutnya

disingkirkan. Dalam pengembangan sistem informasi normal, memerlukan waktu minimal 180 hari, namun dengan menggunakan metode RAD, sistem dapat diselesaikan dalam waktu 30-90 hari. Metode RAD memiliki 3 tahapan yaitu :



Gambar 3.1 Pengembangan Sistem Model RAD

a. Rencana Kebutuhan (*Requirement Planning*)

User dan *analyst* melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

b. Proses Desain Sistem (*Design System*)

Pada tahap ini keaktifan *user* yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Seorang *user* dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain[19].

c. Implementasi (*Implementation*)

Tahapan ini adalah tahapan programmer yang mengembangkan desain suatu program yang telah disetujui oleh *user* dan *analyst*. Sebelum diaplikasikan

pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini user biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut[19].

d. *Testing* (percobaan)

- *Blackbox Testing*

Di *Blackbox Testing* akan di lakukan uji kelayakan fitur-fiutr di dalam sistem yang telah di buat , pengujian ini di lakukan untuk mengetahui error-error yang mungkin masih ada dan bisa di perbaiki. Penguji di tahapan ini adalah mahasiswa dan petugas prodi.

- Uji Ahli Media

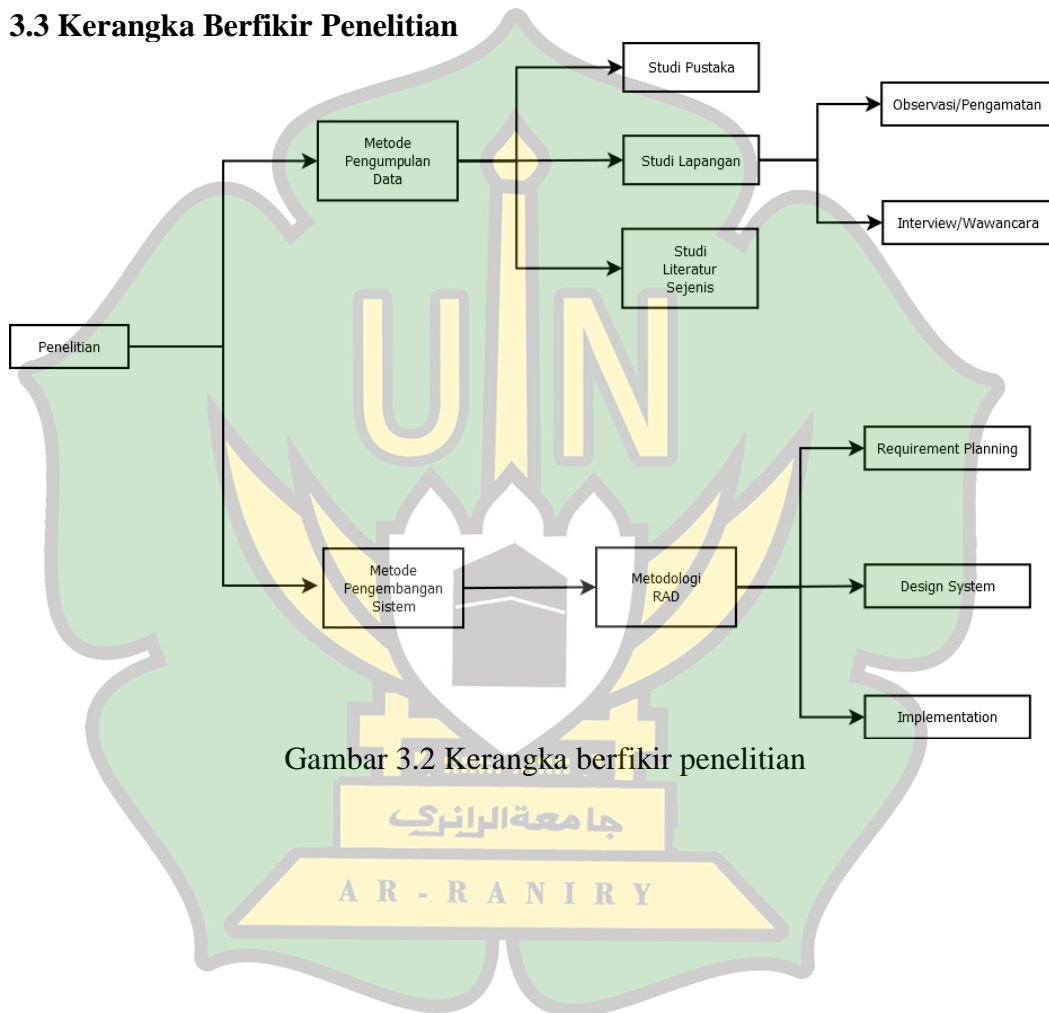
Test produk dilakukan untuk mengetahui kelemahan-kelemahan produk yang dikembangkan sebagai dasar untuk melakukan perbaikan terhadap produk Sistem Pelaporan berbasis web ini. Pada testing ini akan diuji kepada beberapa pakar ahli media. Kisi-kisi instrumen pengujian pada ahli media terlampir dalam tabel berikut :

NO	Aspek yang Dinilai	Altenatif Pilihan				
		1 SS	2 TS	3 R	4 S	5 SS
1	Pemilihan tampilan					
2	Layout/tata letak tombol					
3	Kemenarikan desain					
4	Ketepatan pemilihan huruf					
5	Ketepatan ukuran huruf					
6	Kebergunaan untuk Pordi					
7	Pengaturan jarak					
8	Kejelasan gambar					

9	Ukuran gambar					
10	Kebergunaan untuk mahasiswa					
Jumlah						
Jumlah rata-rata						
Jumlah Persentase						

Tabel 3. 1 Tabel Uji Ahli Media

3.3 Kerangka Berfikir Penelitian



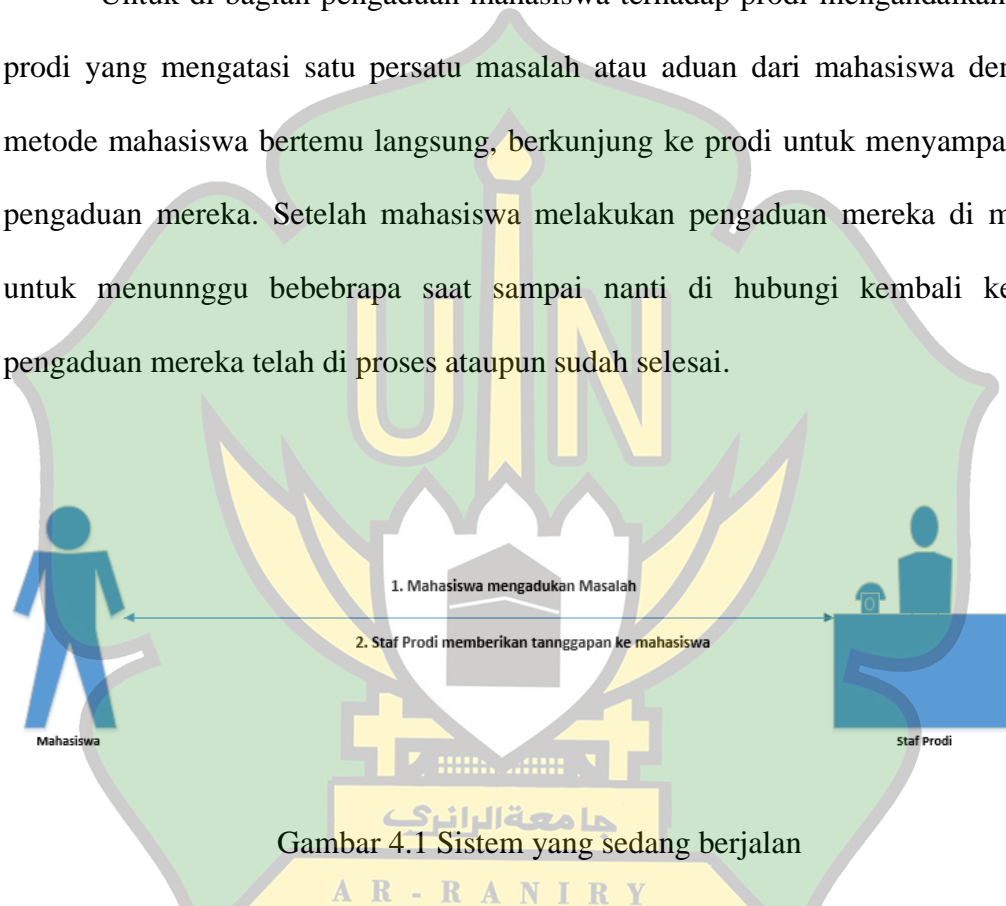
Gambar 3.2 Kerangka berfikir penelitian

BAB IV HASIL DAN PEMBAHASAN

4.1 Perencanaan syarat-syarat (*Requirement Planning*).

4.1.1 Analisa sistem yang sedang berjalan

Untuk di bagian pengaduan mahasiswa terhadap prodi mengandalkan staf prodi yang mengatasi satu persatu masalah atau aduan dari mahasiswa dengan metode mahasiswa bertemu langsung, berkunjung ke prodi untuk menyampaikan pengaduan mereka. Setelah mahasiswa melakukan pengaduan mereka di minta untuk menunggu beberapa saat sampai nanti di hubungi kembali ketika pengaduan mereka telah di proses ataupun sudah selesai.



Gambar 4.1 Sistem yang sedang berjalan

4.1.2 Identifikasi Masalah

Dari analisa di atas di ketahui bahwa sistem pelaporan di prodi ini masih menggunakan sistem manual dan tidak ada sistem yang menggunakan komputersasi sedemikian rupa, sehingga tidak cukup efektif untuk menjadi media interaksi pengaduan mahasiswa kepada prodi, mengingat begitu banyaknya mahasiswa prodi ini yang melaporkan pengaduan. Selain itu pengaduan yang telah

masuk ke prodi tidak ada rekapitulasi atau penyimpanan data laporan yang di adukan oleh mahasiswa sehingga terkadang laporan mahasiswa terlupakan.

Dalam merumuskan solusi bagi permasalahan tersebut di atas, peneliti mengusulkan sebuah sistem yang mampu menjembatani interaksi antara mahasiswa dan juga petugas prodi di dalam hal pengaduan masalah mahasiswa melalui *website* serta mampu menyimpan data pengaduan yang di adukan mahasiswa jikalau nanti data itu di butuhkan kembali bisa di dapat dengan lebih efisien.

4.1.3 Analisa kebutuhan sistem

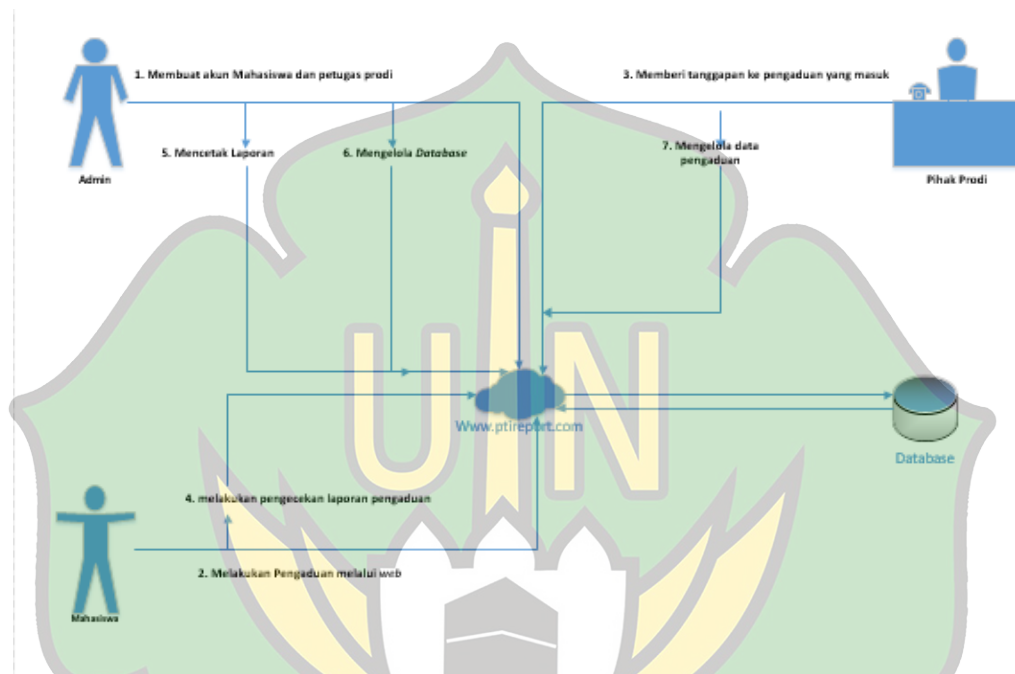
Sistem pengaduan yang sedang berjalan di prodi ini cukup memiliki kelemahan, salah satunya adalah sistem yang belum bisa menjembatani interaksi mahasiswa dengan pihak prodi secara jarak jauh dengan lebih efisien, serta tidak ada perekapan data pengaduan mahasiswa yang ada perlu jikalau data itu di perlukan kembali. Melihat permasalahan di atas, maka di perlukan sebuah sistem yang menjembatani interaksi pengaduan mahasiswa terhadap pihak prodi serta bisa menyimpan data pengaduan yang masuk ke prodi.

4.1.4 Tujuan Pengembangan Sistem

Pengembangan sistem ini bertujuan untuk memudahkan mahasiswa dan juga pihak prodi di dalam hal administrasi khususnya di bagian pengaduan mahasiswa dan penyimpanan data pengaduan yang masuk dari mahasiswa ke pihak prodi dengan lebih efisien. Karena apabila sistem di sebuah organisasi atau

tempat berjalan dengan efisien maka akan lebih mudah untuk proses penyelesaiannya baik dari pihak mahasiswa dan pihak prodinya.

4.1.5 Analisa proses sistem yang di usulkan



Gambar 4.2 Proses sistem yang di usulkan

Dalam proses bisnis yang di usulkan, Admin sebagai pengatur utama websitenya bisa mendaftarkan mahasiswa dan petugas prodi di dalam sebuah database, setelah adminnya mendaftarkan mahasiswa dan petugasnya barulah bisa mahasiswa bisa mengadukan aduan melalui akun masing-masing dengan memasukan username dan password masing-masing yang sudah di buatkan oleh sang admin, begitu pula dengan pihak prodi sudah bisa melkukan interkasi dengan mahasiswa melalui website dengan Login terlebih dahulu dengan username dan password yang telah di buatkan oleh admin, di dalam website Admin memiliki hak akses di antaranya mendaftarkan akun mahasiswa dan pihak prodi, mengelola

data pengaduan di antaranya menghapus, mencetak, mengedit, dll. Sedangkan, untuk petugas prodi memiliki hak akses untuk melihat dan menanggapi aduan mahasiswa yang masuk ke akun petugas masing-masing. Sedangkan untuk mahasiswa memiliki hak akses yaitu melakukan pengaduan di website dengan akun masing-masing dan bisa melihat progres laporan-laporan dari masing-masing mereka apakah laporan sudah di tanggap atau belum.

4.2 Proses Desain (*Workshop* Desain RAD)

4.2.1 Membuat Use Case Diagram

Use Case adalah penjelasan bagaimana interaksi *actor* di dalam sistem pengaduan pada prodi. Seperti di jelaskan pada tabel berikut :

No.	<i>Actor</i>	Deskripsi
1.	Admin	<i>Actor</i> menginput data akun operator prodi dan mahasiswa, mengelola <i>database</i> dan seluruh sistem.
2.	Operator Prodi	<i>Actor</i> yang melakukan penyelesaian pengaduan masalah oleh mahasiswa dan berinteraksi dengan mahasiswa di dalam sistem, mengelola laporan aduan yang masuk ke sistem.
3.	Mahasiswa	<i>Actor</i> yang melakukan pengaduan di dalam sistem dan mengelola daftar laporan yang telah di respon ataupun yang belum.

Tabel 4.1 *Actor Use Case*

Selanjutnya, adalah tabel yang akan menjelaskan interaksi antara *actor* dengan sistem.

No.	<i>Use Case Name</i>	<i>Description</i>	<i>Actor</i>
1.	<i>Login</i>	Kegiatan memasukan <i>username</i> dan <i>password</i> untuk masuk kedalam sistem.	Admin, Operator Prodi dan Mahasiswa
2.	<i>Register/Pendaftaran</i> akun Mahasiswa	Kegiatan pembuatan akun pengaduan Mahasiswa di dalam sistem	Admin
3.	Lihat data pengaduan	Kegiatan melihat dan meng <i>update</i> , merespon aduan dari mahasiswa di dalam sistem	Admin dan Operator Prodi
4.	Lihat data respon/tanggapan	Kegiatan melihat respon atau tanggapan pihak prodi terhadap aduan	Admin dan Operator Prodi

	dari prodi	mahasiswa di dalam sistem	
5.	Input/lihat/update dan tambah data <i>user</i>	Kegiatan melihat/mengubah/menambahkan data pengguna baik itu data pengguna operator prodi ataupun mahasiswa	Admin
6.	Lihat/Cetak laporan pengaduan	Kegiatan melihat atau mencetak laporan pengaduan dari mahasiswa	Admin
7.	Input pengaduan	Kegiatan memasukan laporan pengaduan	Mahasiswa
8.	Lihat daftar laporan yang telah di tanggap	Kegiatan melihat dan melihat laporan yang telah di tanggap sesuai dengan akun masing-masing mahasiswa yang saling berbeda satu sama lain	Mahasiswa
9.	<i>Logout</i>	Kegiatan mengakhiri sistem atau keluar dari sistem	Admin, Operator Prodi dan Mahasiswa

Tabel 4.2 Tabel interaksi aktor di sistem

4.2.1.1 Use Case Scenario

Use Case Name	<i>Login</i>	
Use Case ID	1	
Actor	Admin, Operator Prodi dan Mahasiswa	
Description	Kegiatan memasukan <i>username</i> dan <i>password</i> untuk masuk dan mengakses sistem	
Precondition	Admin, Operator Prodi dan Mahasiswa membuka sistem	
Typical Course of Events	Actor Action	System Response
	1. Masukan <i>username</i> dan <i>password</i>	2. Cek <i>username</i> dan <i>password</i>
		3. Menampilkan halaman <i>website</i> sesuai dengan akun.
Alternate courses	Jika no.2 tidak benar maka sistem akan menampilkan peringatan kesalahan	
Conclusion	Admin, Operator Prodi dan Mahasiswa masuk kedalam sistem	
Post condition	Admin, Operator Prodi dan Mahasiswa bisa mengakses sistem	

Tabel 4.3 Use Case Scenario Login

Use Case Name	Register/Pendaftaran akun Mahasiswa	
Use Case ID	2	
Actor	Admin	
Description	Kegiatan pembuatan akun pengaduan Mahasiswa di dalam sistem	
Precondition	Admin masuk ke dalam sistem	
Typical Course of Events	Actor Action	System Response
	1. Memilih menu Registrasi	2. Menampilkan daftar akun mahasiswa yang sudah ada
	3. Meng <i>klik</i> tombol tambah Mahasiswa	4. Menampilkan form tambah data Mahasiswa
	5. Masukkan NIM, Nama, <i>username</i> , <i>password</i> dan nomer telpon.	6. Menghubungkan dengan <i>database</i> Mahasiswa
	7. Klik “simpan” sebagai tanda penambahan data mahasiswa telah sukses tersimpan di dalam <i>database</i>	
Alternate courses	Jika ada kesamaan <i>username</i> data pada <i>database</i> dengan data yang baru di masukkan sistem tidak akan menyimpan data yang sama	
Conclusion	Admin menambahkan/mengelola akun mahasiswa	
Post condition	Data akun mahasiswa sudah tersimpan di dalam <i>database</i> .	

Tabel 4.4 Use Case Scenario Register

Use Case Name	Lihat data pengaduan	
Use Case ID	3	
Actor	Admin dan Operator Prodi	
Description	Kegiatan melihat dan meng <i>update</i> , merespon aduan dari mahasiswa di dalam sistem	
Precondition	Admin dan Operator Prodi masuk kedalam sistem	
Typical Course of Events	Actor Action	System Response
	1. Memilih menu Pengaduan	2. Menampilkan daftar pengaduan mahasiswa yang sudah ada
	3. Meng <i>klik</i> “More” untuk melihat detail laporan.	4. Menghubungkan dengan <i>database</i> pengaduan
	5. Memberikan tanggapan di pengaduan mahasiswa yang telah tampil dan klik tombol “Kirim”	6. Menghubungkan dengan <i>database</i> Pengaduan
		7. Menampilkan pesan bahwa tanggapan telah

		di kirim.
Conclusion	Admin dan Operator Prodi melihat dan menggapai aduan mahasiswa	
Post condition	Pengaduan mahasiswa telah berhasil di beri tanggapan	

Tabel 4.5 Use Case Scenario lihat data pelanggan

Use Case Name	Lihat data respon/tanggapan dari prodi	
Use Case ID	4	
Actor	Admin dan Operator Prodi	
Description	Kegiatan melihat respon atau tanggapan pihak prodi terhadap aduan mahasiswa di dalam sistem	
Precondition	Admin dan Operator Prodi masuk ke dalam sistem	
Typical Course of Events	Actor Action	System Response
	1. Memilih menu Respon	2. Menampilkan daftar aduan mahasiswa yang telah di tanggapi oleh operator prodi
	3. Meng klik “More” untuk melihat detail laporan	4. Menghubungkan dengan <i>database</i>
		5. Menampilkan detail laporan pengaduan yang telah di tanggapi
Conclusion	Admin dan Operator Prodi melihat dan menggapai aduan mahasiswa	
Post condition	-	

Tabel 4.6 Use Case Scenario lihat data Respon

Use Case Name	Input/lihat/update data user	
Use Case ID	5	
Actor	Admin	
Description	Kegiatan melihat/mengubah/menambahkan data pengguna baik itu data pengguna operator prodi ataupun mahasiswa	
Precondition	Admin masuk kedalam sistem	
Typical Course of Events	Actor Action	System Response
	1. Memilih menu <i>User</i>	2. Menampilkan daftar <i>User</i> yang sudah ada
	3. Meng klik “Edit” untuk melihat dan mengubah data <i>user</i>	4. Menghubungkan dengan <i>database</i>
	5. Meng klik “Hapus” untuk menghapus <i>user</i>	6. Menampilkan pertanyaan apakah yakin untuk menghapus?
	7. Klik “OK” untuk	8. Menampilkan pesan

	menghapus atau klik “Cancel” untuk membatalkan perintah hapus	bahwa tindakan hapus akun sudah/tidak terjadi.
	9. Klik tanda tambah untuk menambahkan <i>user</i>	10. Tampilkan <i>form</i> penambahan <i>user</i>
	11. Masukkan Nama, <i>username</i> , <i>password</i> , nomer telpon dan level akses akun yang akan di buat	12. Menghubungkan dengan <i>database</i>
	13. Klik “Simpan”	14. Tampil pesan bahwa akun telah sukses di buat
Conclusion	Admin dan Operator Prodi melihat dan meng <i>update</i> data <i>user</i>	
Post condition	Data <i>user</i> telah di lihat/di <i>update</i> /dihapus atau telah di tambahkan	

Tabel 4.7 Use Case Scenario input/lihat dan update data user

Use Case Name	Lihat/Cetak laporan pengaduan	
Use Case ID	6	
Actor	Admin	
Description	Kegiatan melihat/mengubah/menambahkan data pengguna baik itu data pengguna operator prodi ataupun mahasiswa	
Precondition	Admin masuk kedalam sistem	
Typical Course of Events	Actor Action	System Response
	1. Memilih menu Laporan	2. Menampilkan daftar pengaduan yang sudah di tanggapi
	3. Meng <i>klik</i> “More” untuk melihat detail ppengaduan	4. Menghubungkan dengan <i>database</i>
	5. Meng <i>klik</i> tanda <i>printer</i>	6. Menampilkan <i>preview</i> hasil <i>print</i> laporan
	7. Klik simpan atau <i>print</i>	8. Mencetak/menyimpan laporan.
Conclusion	Admin melihat/dan mencetak laporan	
Post condition	Data laporan telah di lihat/dicetak dengan berhasil	

Tabel 4.8 Use Case Scenario Lihat dan cetak laporan pengaduan

Use Case Name	Input pengaduan	
Use Case ID	7	
Actor	Mahasiswa	
Description	Kegiatan memasukan laporan pengaduan	
Precondition	Mahasiswa masuk kedalam sistem	
Typical Course of Events	Actor Action	System Response
	1. Mengetikan laporan boleh disertakan dengan gambar	2. Menampilkan daftar laporan yang telah di tanggapi
	3. Meng klik “Kirim” untuk mengirim pengaduan	4. Sistem menampilkan pesan bahwa pegaduan akan segera di proses
Conclusion	Mahasiswa melakukan pengaduan dan melihat daftar pengaduan yang telah di tanggapi	
Post condition	Pengaduan telah selesai di lakukan	

Tabel 4.9 Use Case Scenario Input Pengaduan

Use Case Name	Lihat daftar laporan yang telah di tanggapi	
Use Case ID	8	
Actor	Mahasiswa	
Description	Kegiatan melihat dan melihat laporan yang telah di tanggapi sesuai dengan akun masing-masing mahasiswa yang saling berbeda satu sama lain	
Precondition	Mahasiswa masuk kedalam sistem	
Typical Course of Events	Actor Action	System Response
	1. Meng klik “More” untuk melihat detail laporan yang telah di tanggapi	2. Menampilkan detail laporan pengaduan
	3. Meng klik “Hapus” untuk menghapus daftar pengaduan yang telah di proses	4. Sistem menampilkan pesan apakah yakin untuk menghapus daftar pengaduan
	5. Klik “OK” untuk menghapus dan klik “Cancel” untuk membatalkan perintah hapus.	6. Sistem membatalkan/melakukan penghapusan data laporan yang telah di proses
Conclusion	Mahasiswa melihat daftar pengaduan yang telah di tanggapi	
Post condition	-	

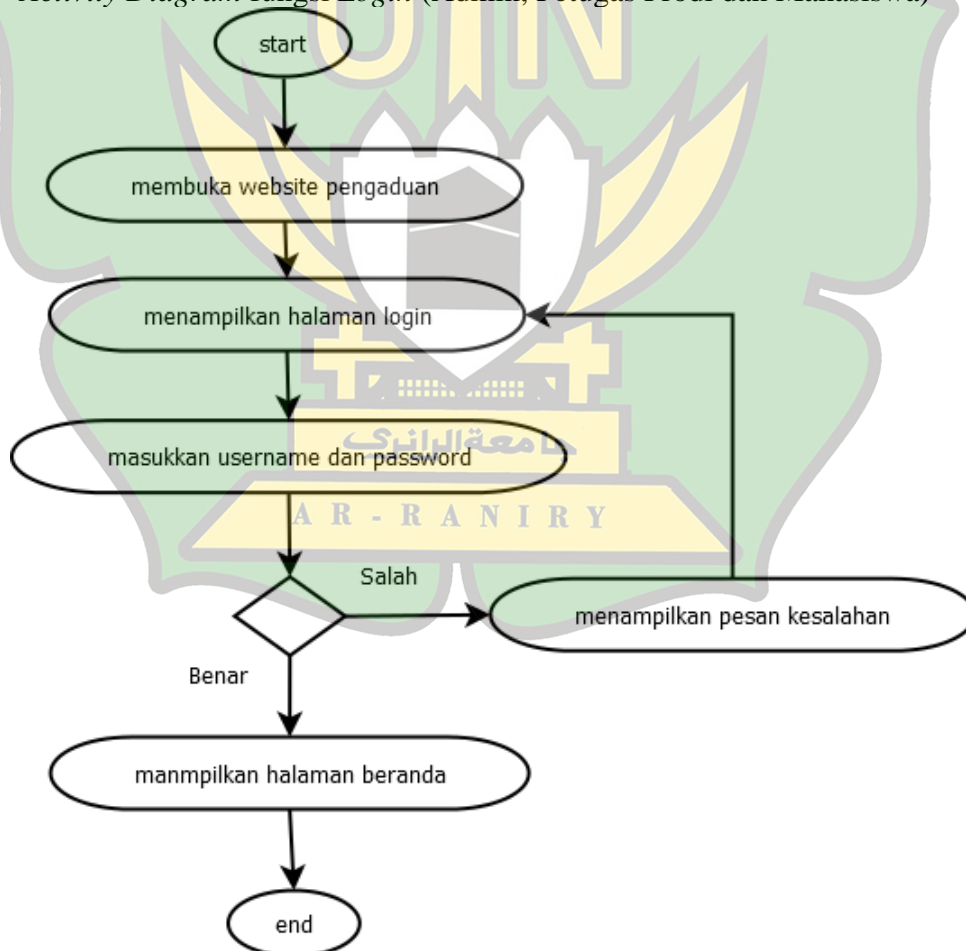
Tabel 4.10 Use Case Scenario lihat daftar laporan yang telah di tanggapi

Use Case Name	<i>Logout</i>	
Use Case ID	9	
Actor	Admin, Operator Prodi dan Mahasiswa	
Description	Kegiatan keluar dari sistem	
Precondition	Admin, Operator Prodi dan Mahasiswa masuk ke sistem	
Typical Course of Events	Actor Action	System Response
	1. Memilih menu <i>Logout</i>	2. Menghubungkan dengan <i>database</i>
		3. Menampilkan halaman <i>Login</i>
Alternate courses	-	
Conclusion	Admin, Operator Prodi dan Mahasiswa keluar dari sistem	
Post condition	Keluar sistem	

Tabel 4.11 Use Case Scenario Logout

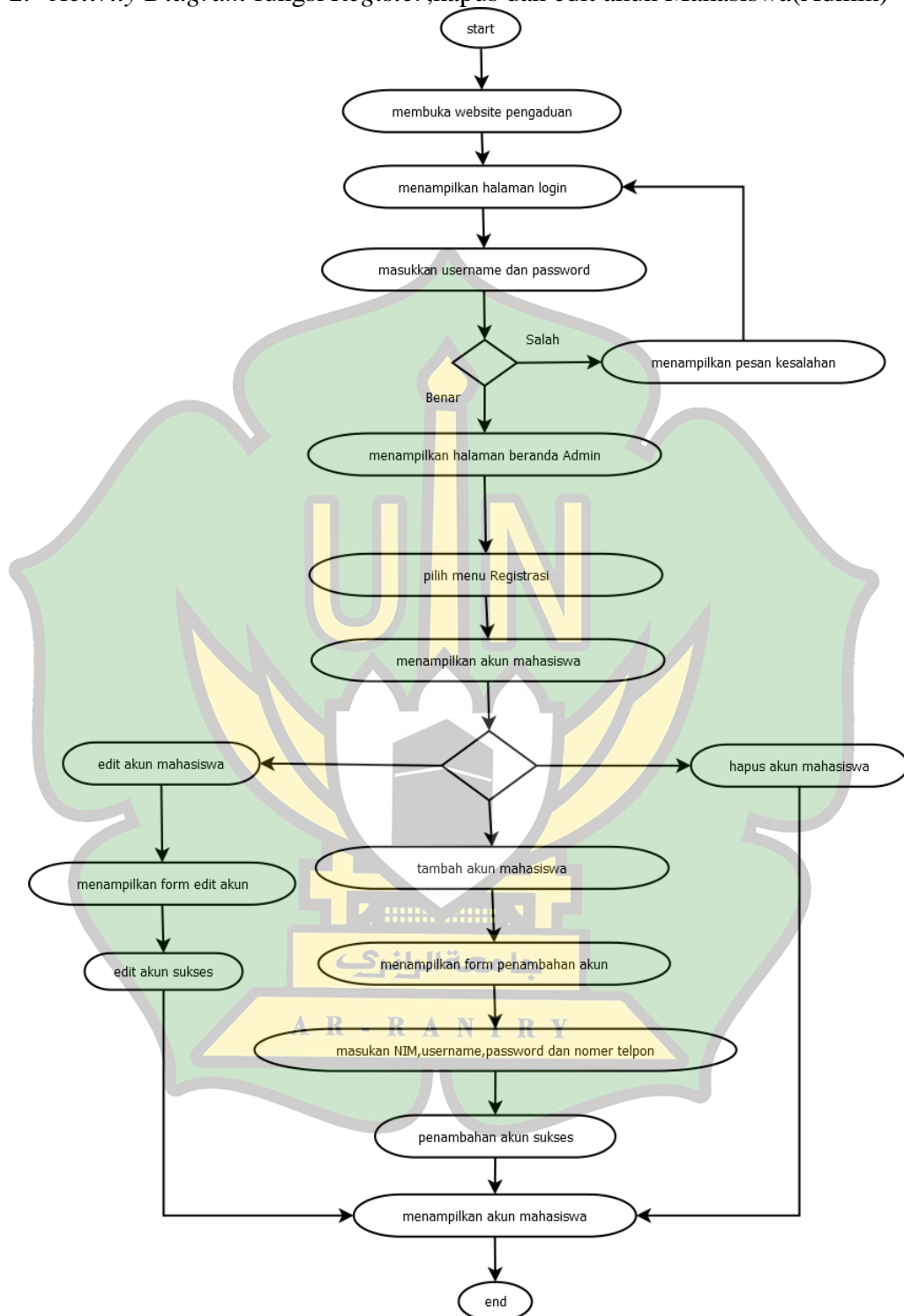
4.2.2 Membuat Activity Diagram

1. Activity Diagram fungsi Login (Admin, Petugas Prodi dan Mahasiswa)



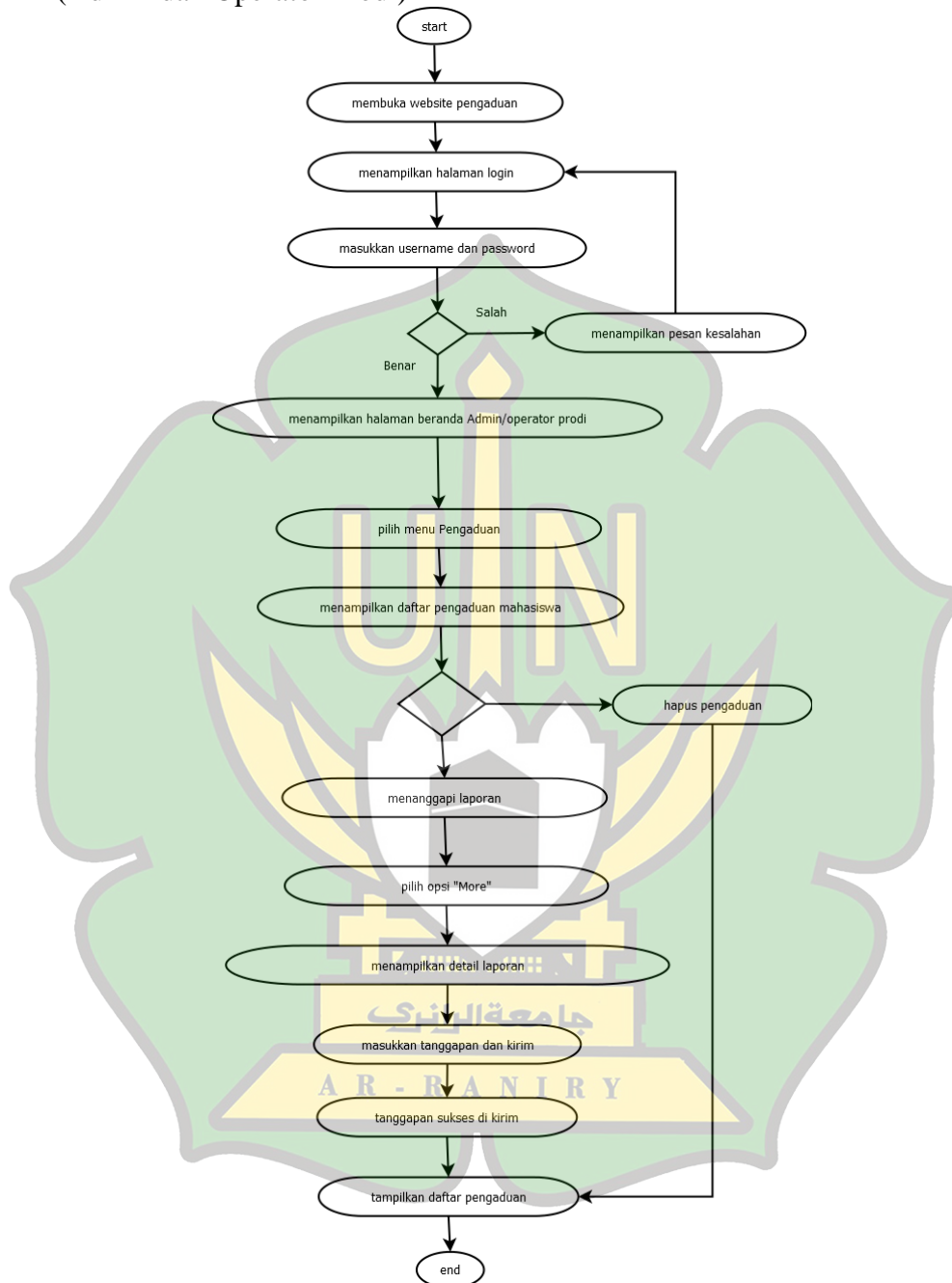
Gambar 4.3 Activity Diagram fungsi Login (Admin, Petugas Prodi dan Mahasiswa)

2. Activity Diagram fungsi Register, hapus dan edit akun Mahasiswa(Admin)



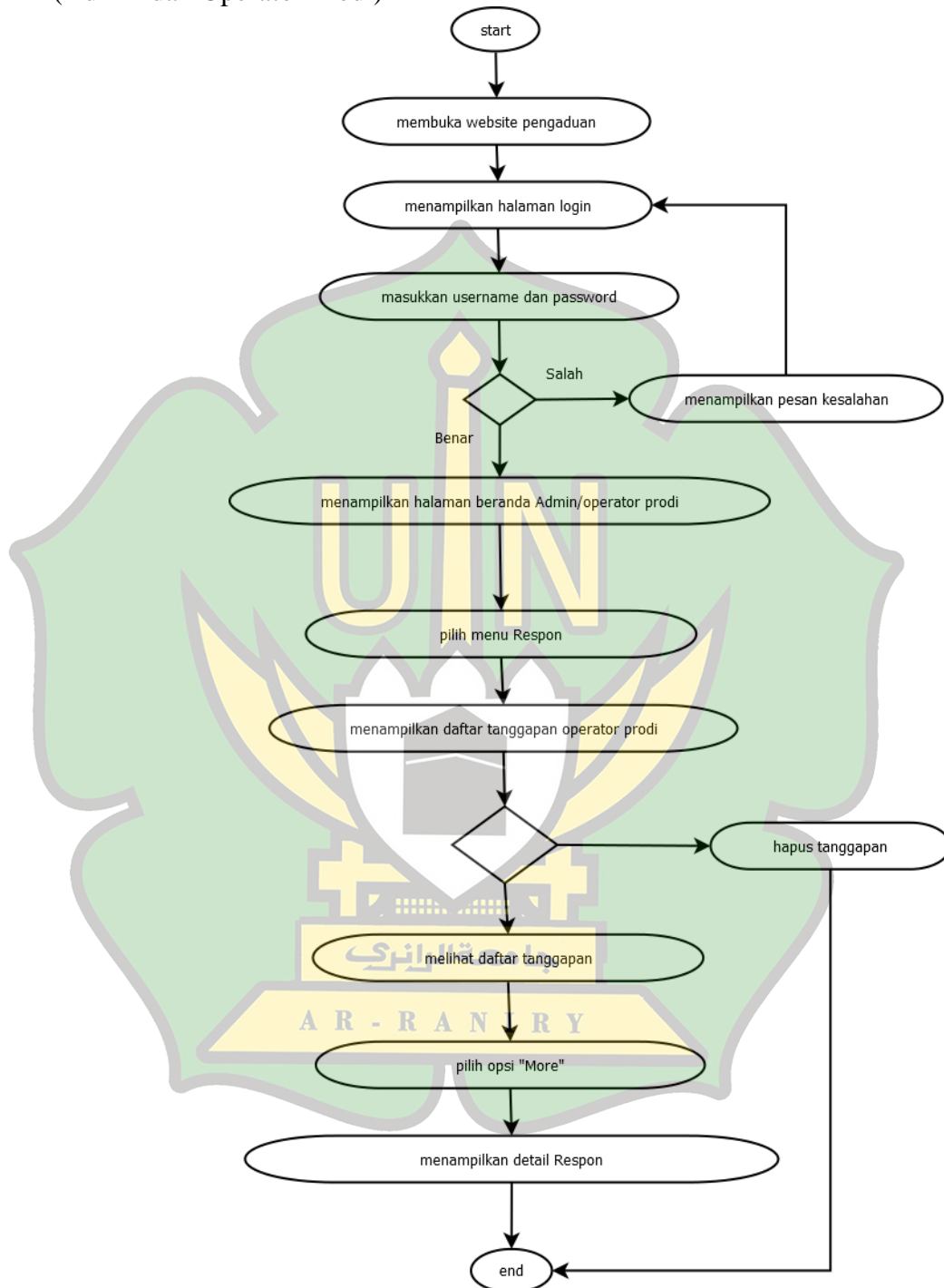
Gambar 4.4 Activity Diagram fungsi Register, hapus dan edit akun Mahasiswa(Admin)

3. *Activity Diagram* Lihat dan tanggap daftar pengaduan dari Mahasiswa (Admin dan Operator Prodi)



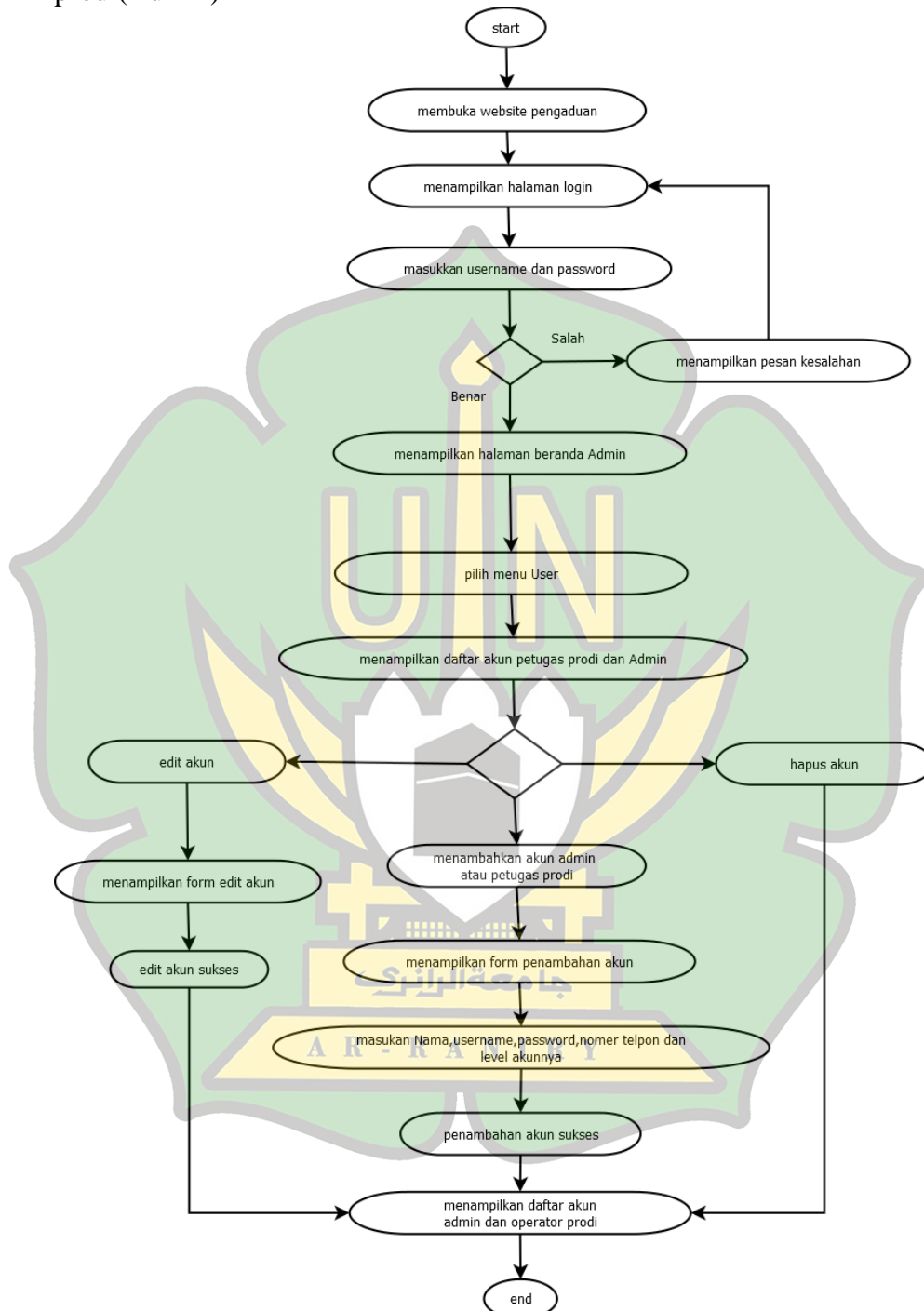
Gambar 4.5 *Activity Diagram* Lihat dan tanggap daftar pengaduan dari Mahasiswa (Admin dan Operator Prodi).

4. *Activity Diagram* Lihat dan tanggapı daftar pengaduan dari Mahasiswa (Admin dan Operator Prodi)



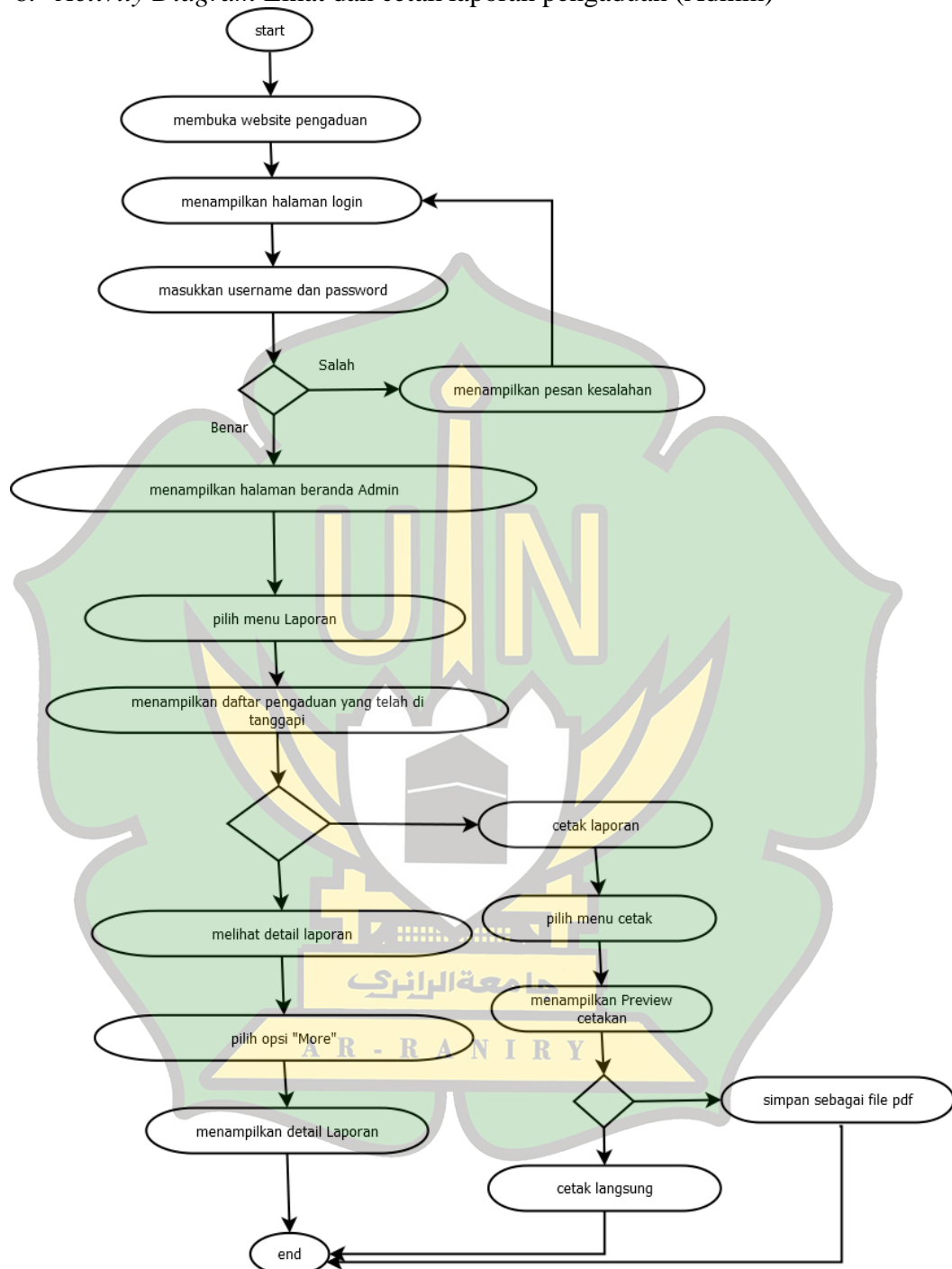
Gambar 4.6 Activity Diagram Lihat dan tanggapı daftar pengaduan dari Mahasiswa (Admin dan Operator Prodi)

5. *Activity Diagram* Lihat, hapus, dan tambah akun admin atau operator prodi(Admin)



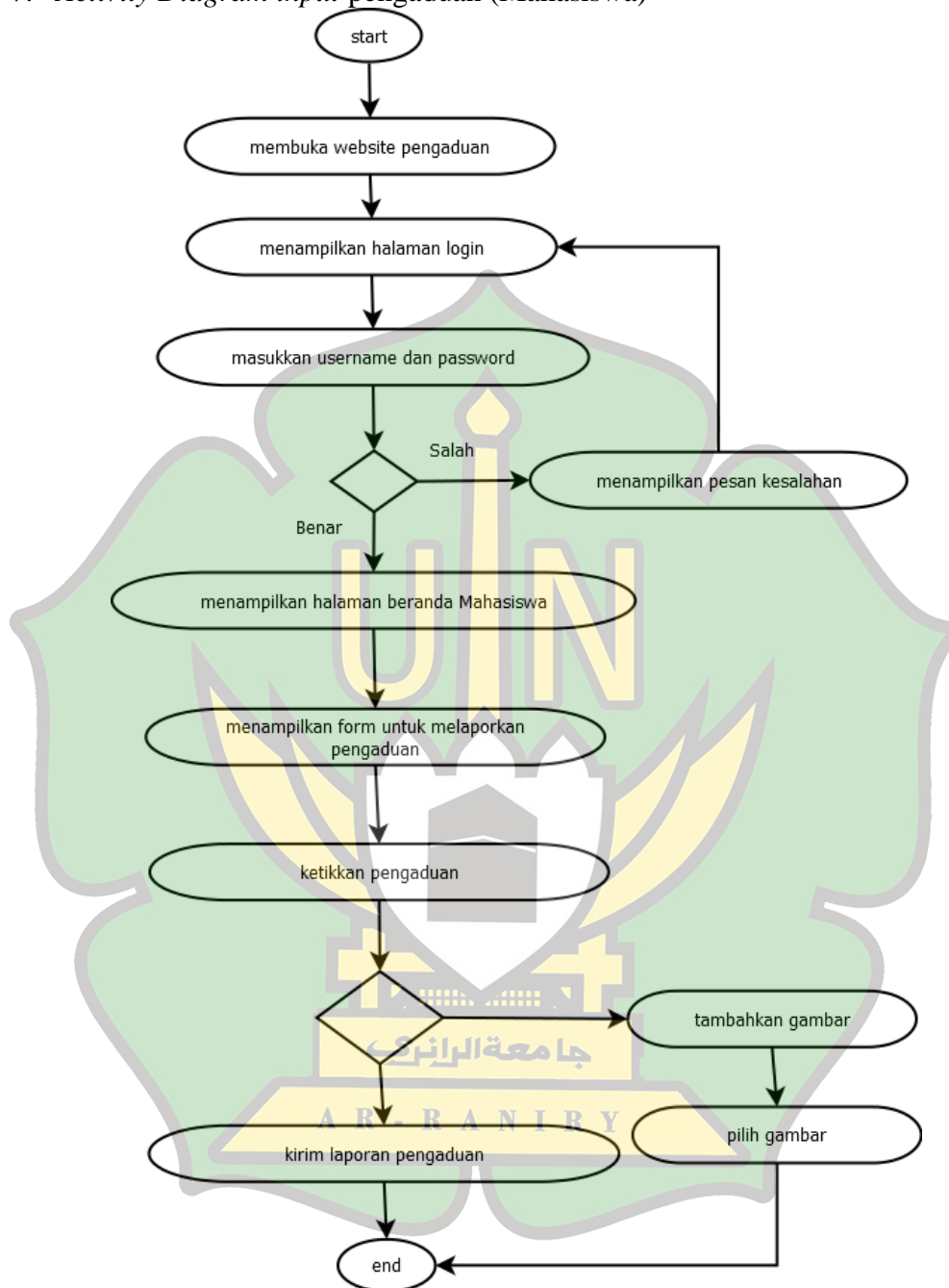
Gambar 4.7 *Activity Diagram* Lihat, hapus, dan tambah akun admin atau operator prodi(Admin)

6. *Activity Diagram* Lihat dan cetak laporan pengaduan (Admin)



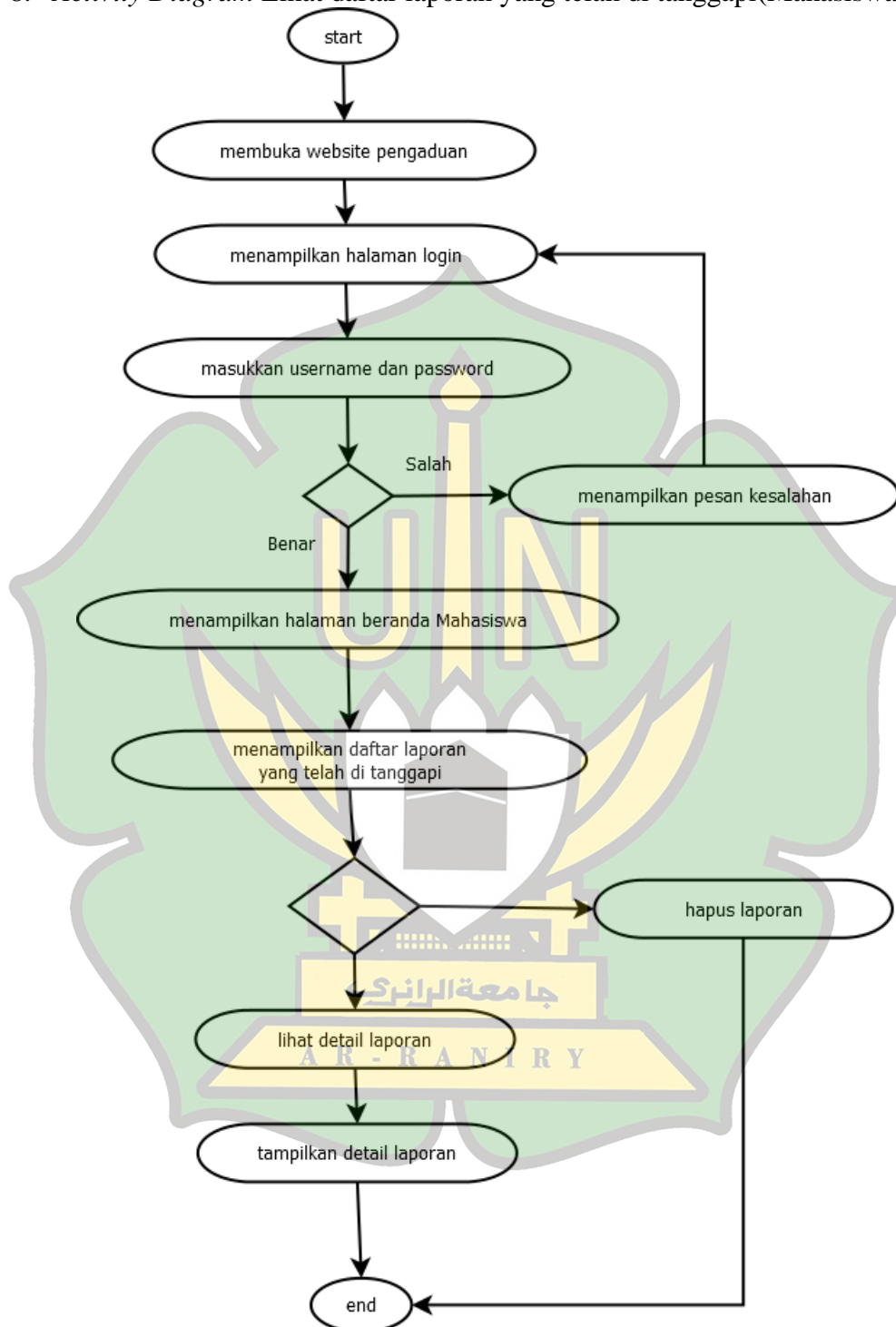
Gambar 4.8 *Activity Diagram* Lihat dan cetak laporan pengaduan (Admin)

7. Activity Diagram input pengaduan (Mahasiswa)



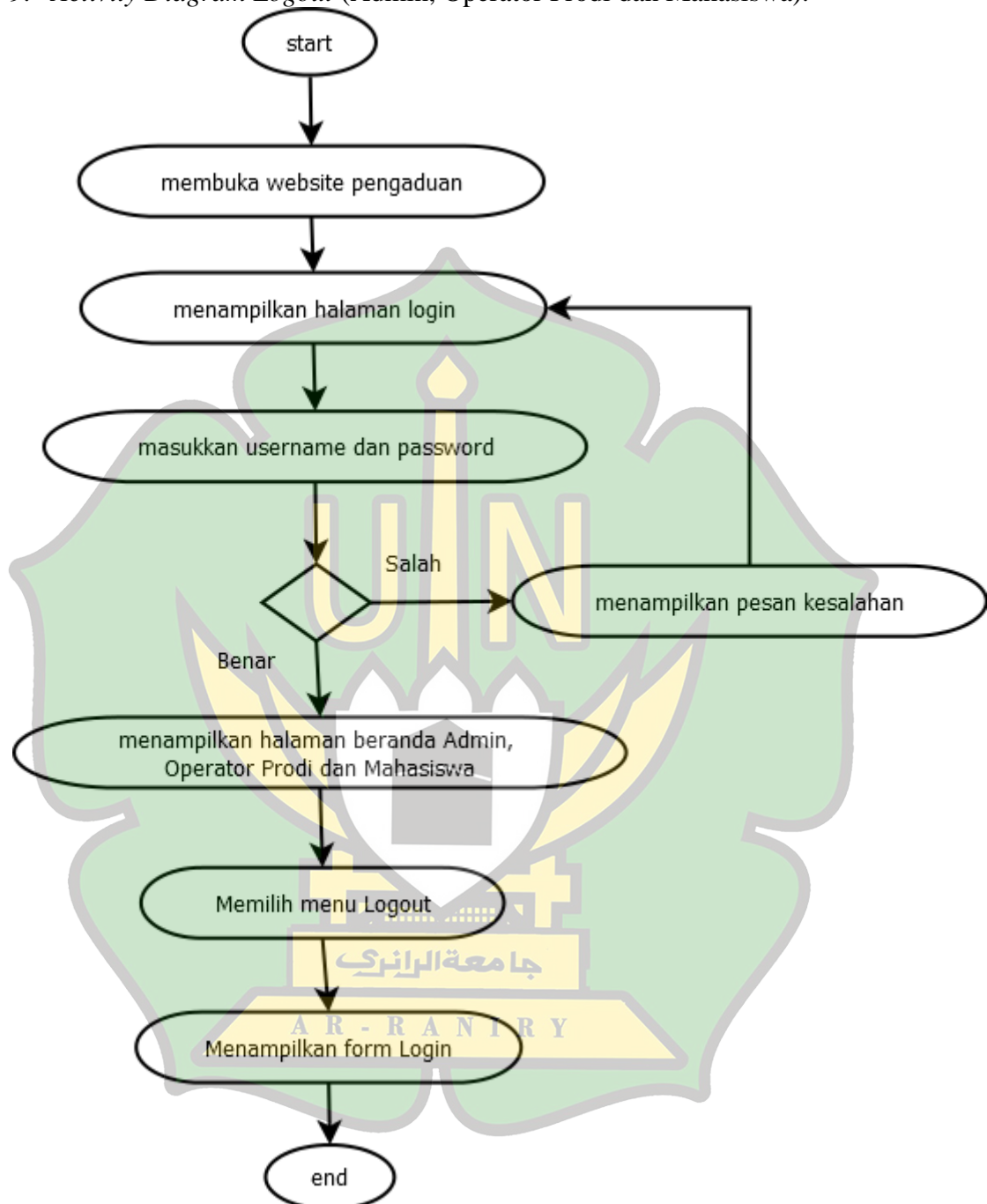
Gambar 4.9 Activity Diagram input pengaduan (Mahasiswa)

8. Activity Diagram Lihat daftar laporan yang telah di tanggapi(Mahasiswa)



Gambar 4.10 Activity Diagram Lihat daftar laporan yang telah di tanggapi(Mahasiswa)

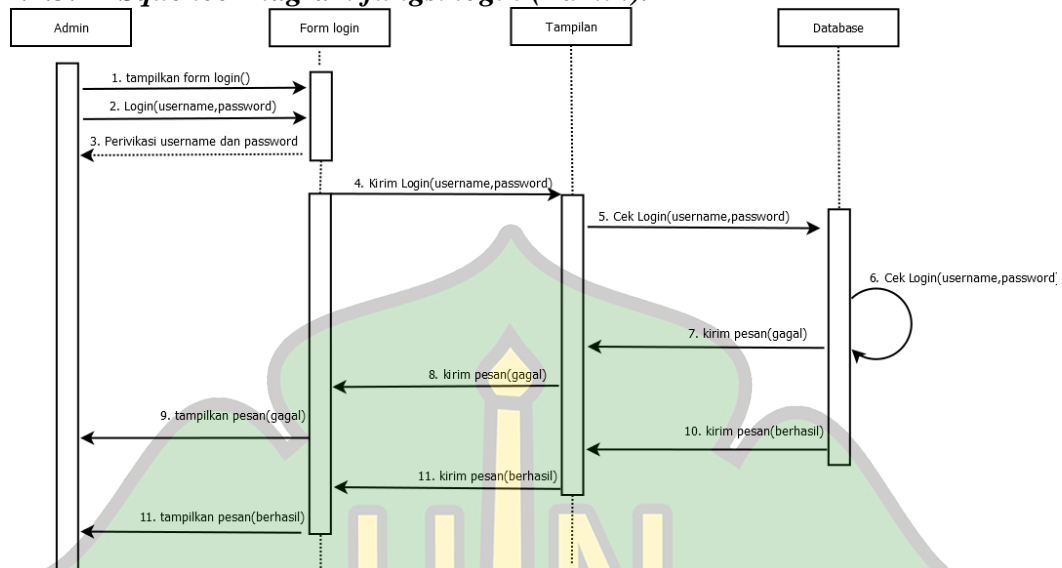
9. Activity Diagram Logout (Admin, Operator Prodi dan Mahasiswa).



Gambar 4.11 Activity Diagram Logout (Admin, Operator Prodi dan Mahasiswa).

4.2.3 Membuat Squence Diagram

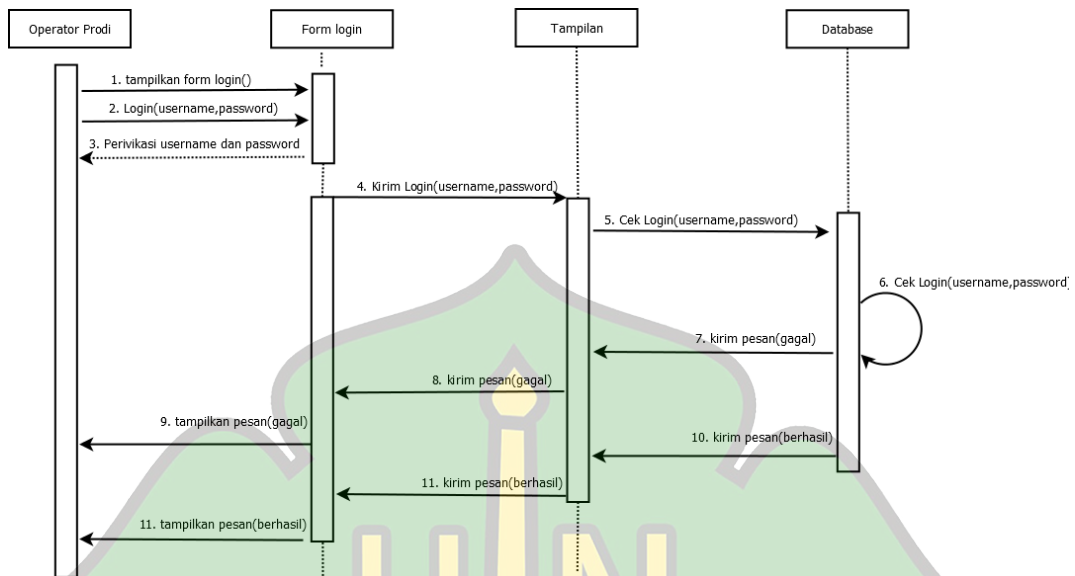
4.2.3.1 Squence Diagram fungsi login (Admin).



Gambar 4.12 Squence Diagram fungsi login (Admin).

Keterangan gambar : Pada saat admin melakukan *login*, maka sistem akan meminta admin untuk memasukkan *username* dan *password*, ketika sistem membaca *username* dan *password*, sistem akan mengecek *username* dan *password* yang di masukkan oleh admin di dalam *database*. jika benar admin akan langsung di alihkan kehalaman beranda admin, jika salah maka akan muncul pesan kesalahan *username* dan *password* dan admin di minta memasukkan ulang *username* dan *password*.

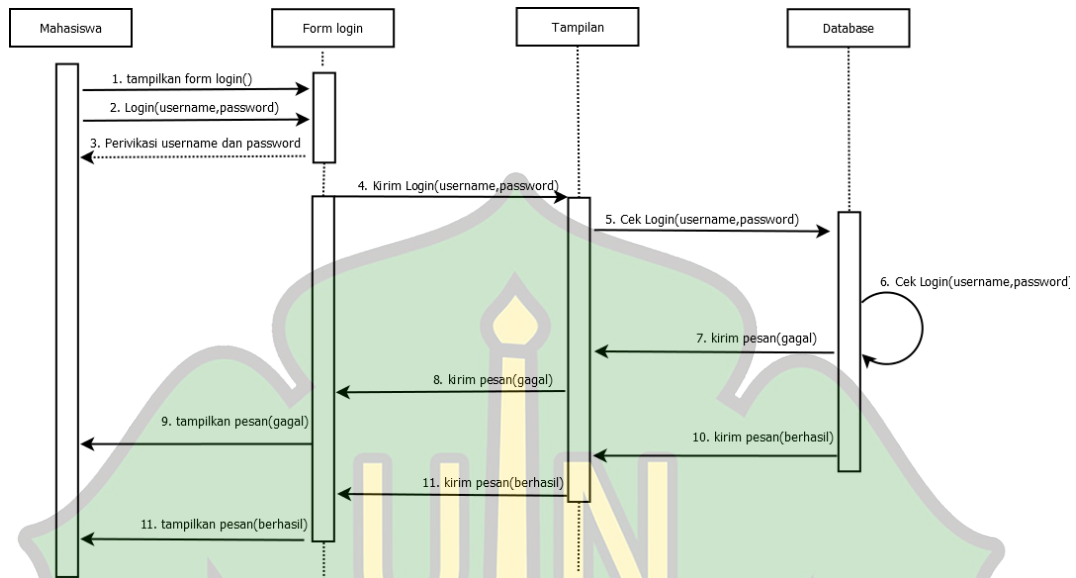
4.2.3.2 Sequence Diagram fungsi login (Operator Prodi).



Gambar 4.13 Squence Diagram fungsi login (Admin).

Keterangan gambar : Pada saat operator prodi melakukan *login*, maka sistem akan meminta admin untuk memasukkan *username* dan *password*, ketika sistem membaca *username* dan *password*, sistem akan mengecek *username* dan *password* yang di masukan oleh admin di dalam *database*. jika benar admin akan langsung di alihkan kehalaman beranda admin, jika salah maka akan muncul pesan kesalahan *username* dan *password* dan admin di minta memasukkan ulang *username* dan *password*.

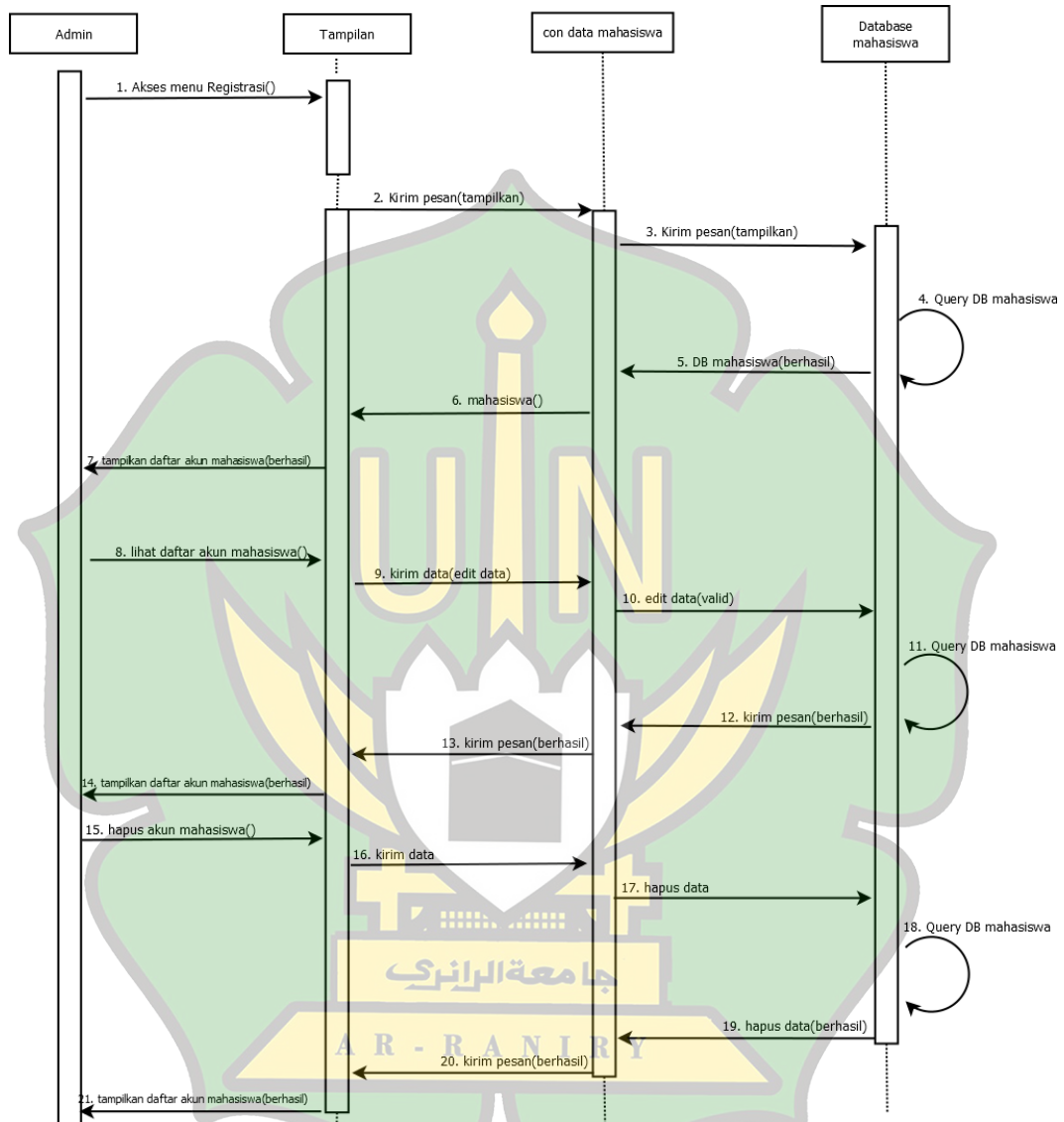
4.2.3.3 Sequence Diagram fungsi login (Mahasiswa).



Gambar 4. 14 .Squence Diagram fungsi login (Mahasiswa).

Keterangan gambar : Pada saat mahasiswa melakukan *login*, maka sistem akan meminta admin untuk memasukkan *username* dan *password*, ketika sistem membaca *username* dan *password*, sistem akan mengecek *username* dan *password* yang di masukkan oleh admin di dalam *database*. jika benar admin akan langsung di alihkan kehalaman beranda admin, jika salah maka akan muncul pesan kesalahan *username* dan *password* dan admin di minta memasukkan ulang *username* dan *password*.

4.2.3.4 Sequence Diagram fungsi lihat dan hapus akun mahasiswa(Admin).

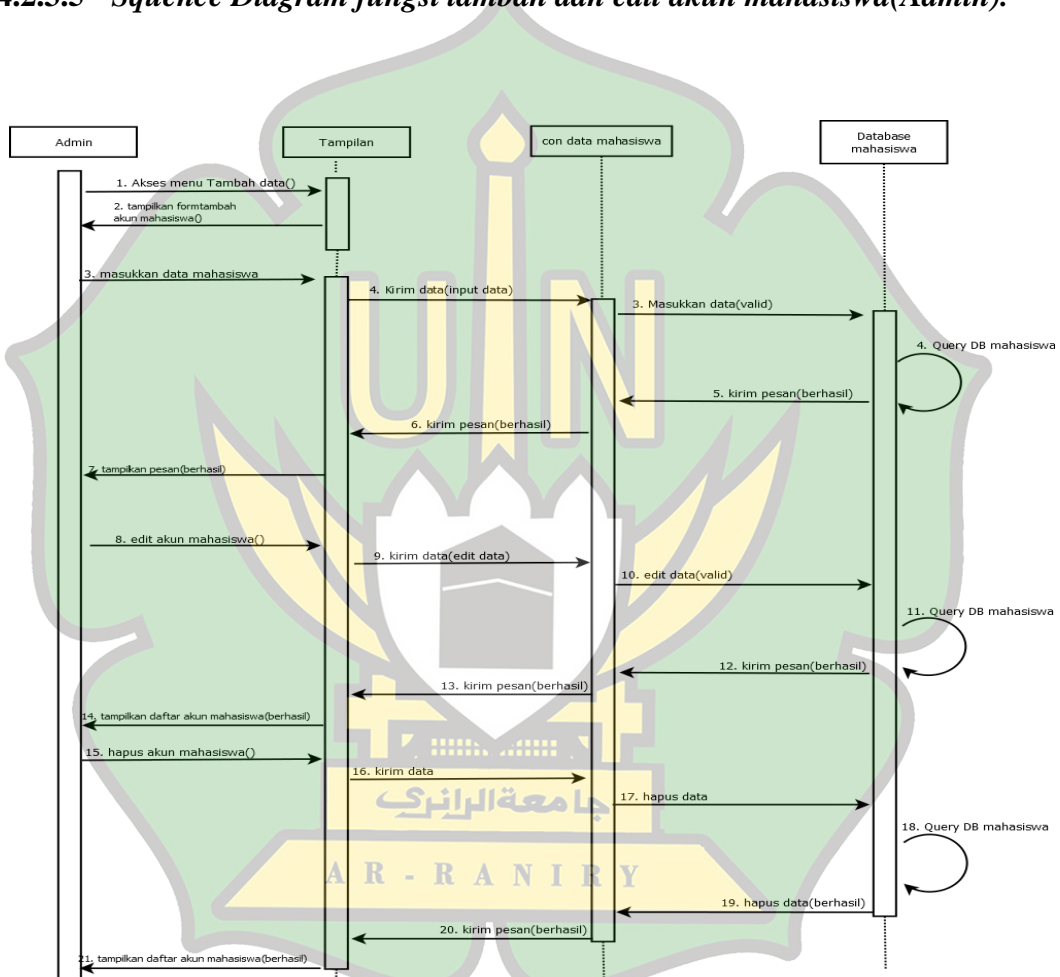


Gambar 4.15 Sequence Diagram fungsi lihat dan hapus akun mahasiswa(Admin).

Keterangan gambar : Pada saat admin ingin melihat daftar akun mahasiswa,yang pertama di lakukan adalah admin meminta kepada sistem untuk menampilkan seluruh data akun mahasiswa yang telah terdata dan setelahnya sistem akan mencari semua data akun mahasiswa di dalam database mahasiswa

dan menampilkannya. Apabila admin ingin menghapus data akun mahasiswa tersebut maka admin akan memilih opsi hapus pada sistem dan sistem akan menghapus data akun mahasiswa yang di pilih di dalam database mahasiswa.

4.2.3.5 *Sequence Diagram fungsi tambah dan edit akun mahasiswa(Admin).*

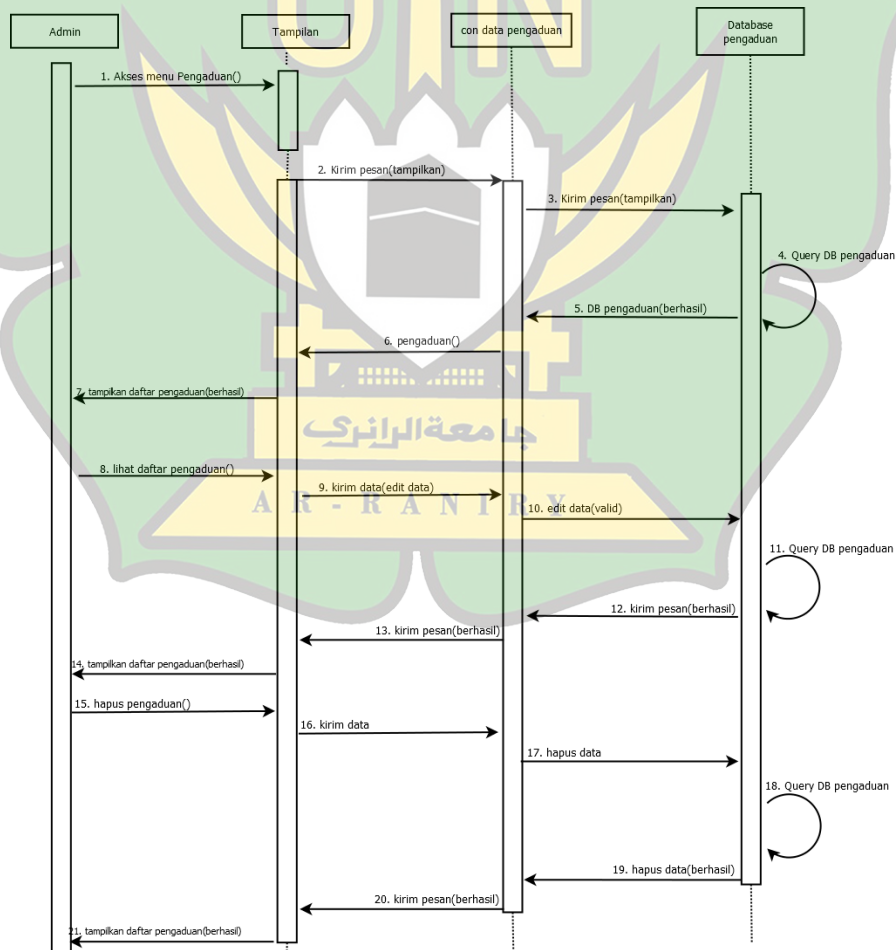


Gambar 4.16 Squence Diagram fungsi tambah dan edit akun mahasiswa(Admin).

Keterangan gambar : pada saat admin ingin menambahkan akun mahasiswa, admin akan memilih menu tambah akun mahasiswa, setelahnya sistem akan menampilkan form penambahan akun ke admin, admin memasukkan data-data mahasiswa yang akan di buatkan, sistem akan memasukkan data ke

dalam database mahasiswa dan mengirimkan pesan bahwa penambahan akun telah berhasil dilakukan. Dan apabila admin ingin mengubah data yang telah dibuat sebelumnya, maka admin akan memilih opsi edit di dalam sistem, sistem akan menampilkan detail akun petugas dan admin bisa mengubah data di dalamnya dan ketika admin telah selesai menyimpan, admin akan memilih menu simpan dan sistem akan menyimpan data yang telah berubah kembali ke database petugas.

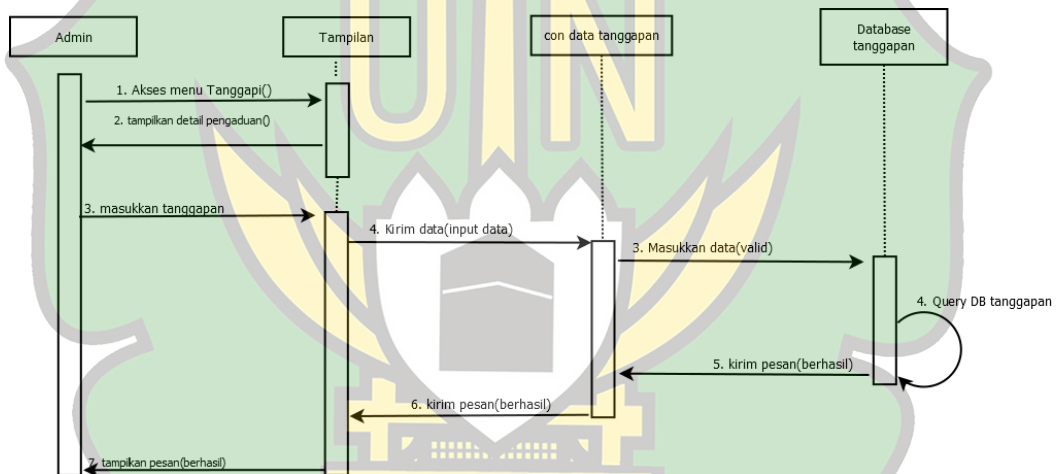
4.2.3.6 Sequence Diagram fungsi lihat dan hapus data pengaduan(Admin).



Gambar 4.17 Sequence Diagram fungsi lihat dan hapus data pengaduan(Admin).

Keterangan gambar : pada saat admin ingin melihat daftar pengaduan yang masuk admin akan memilih menu pengaduan, dan sistem akan memanggil seluruh data tentang pengaduan di dalam database pengaduan dan menampilkannya kepada admin, apabila admin ingin menghapus data pengaduan tersebut maka admin akan memilih opsi hapus pada sistem dan sistem akan menghapus data pengaduan yang di pilih di dalam database pengaduan.

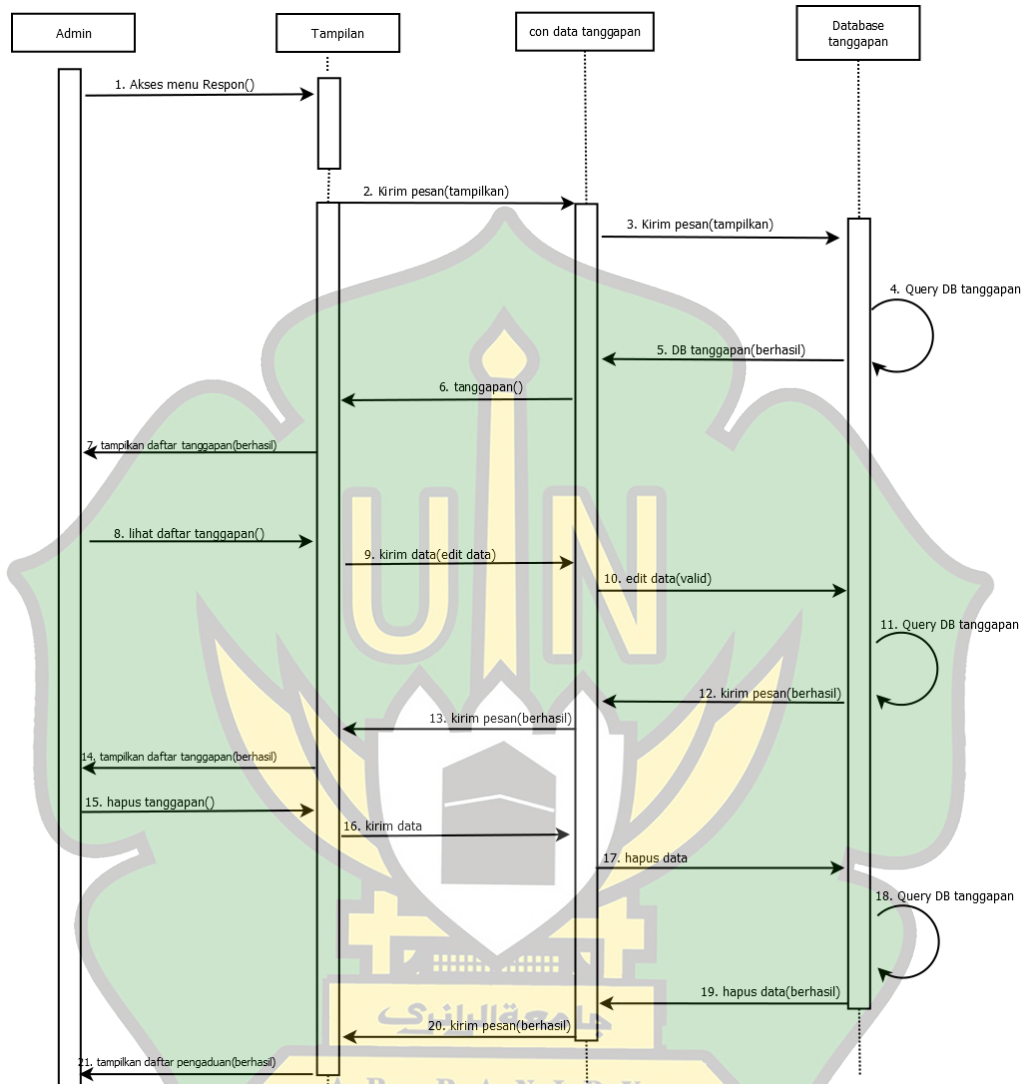
4.2.3.7 Sequence Diagram fungsi tanggapi pengaduan mahasiswa(Admin).



Gambar 4.18 Sequence Diagram fungsi tanggapi pengaduan mahasiswa(Admin).

Keterangan gambar : pada saat admin ingin menanggapi salah satu pengaduan, admin akan memilih menu pengaduan dan melihat pengaduan tersebut terlebih dahulu, setelah itu admin akan memilih menu menanggapi pengaduan yang ada pada sistem dan sistem akan menampilkan form untuk memberikan tanggapan terhadap pengaduan, setelah selesai memberikan tanggapan admin akan menekan tombol kirim pada sistem dan sistem akan menyimpan data tanggapannya dalam *database* tanggapan.

4.2.3.8 Sequence Diagram fungsi lihat dan hapus tanggapan(Admin).

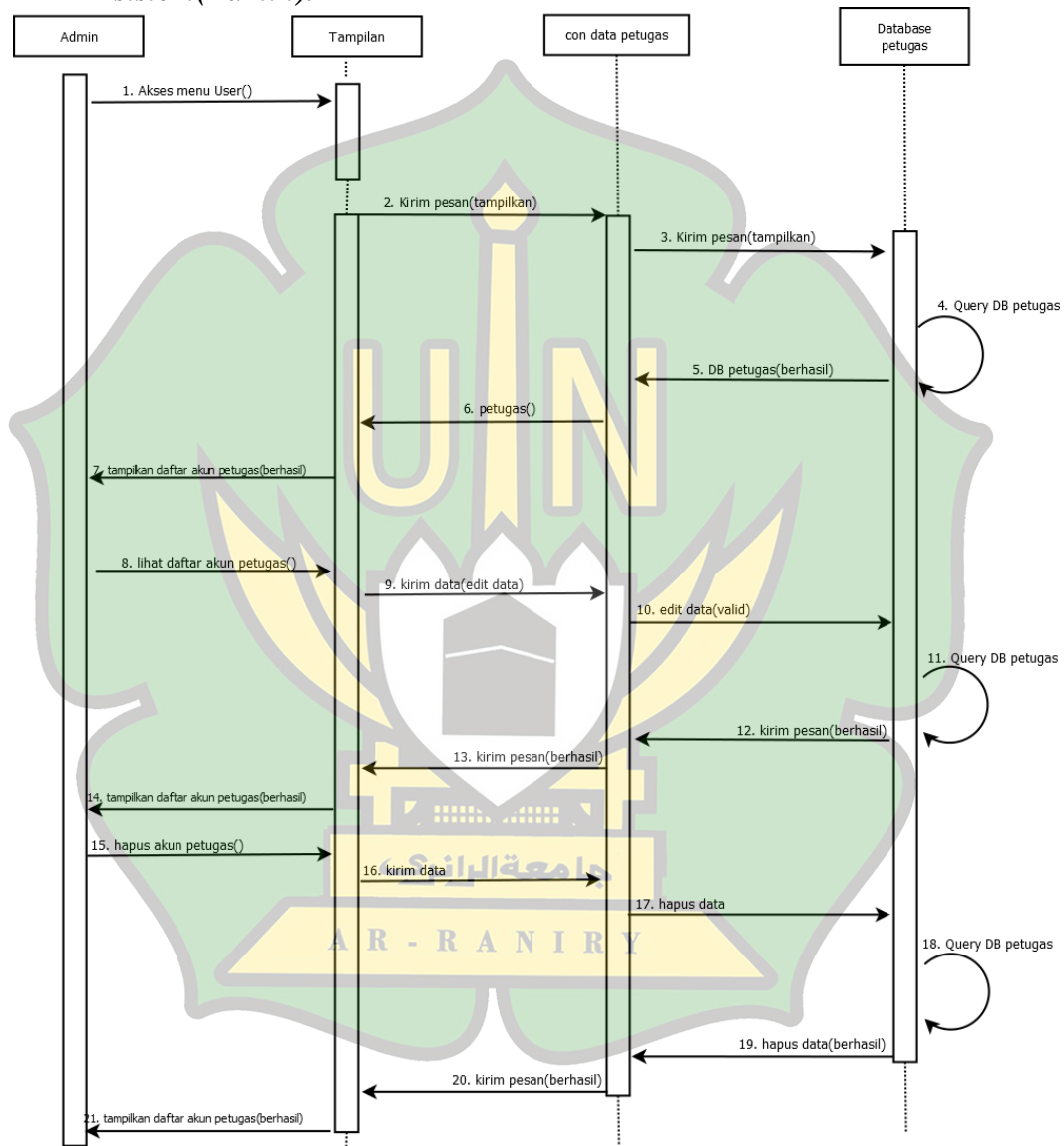


Gambar 4.19 Sequence Diagram fungsi lihat dan hapus tanggapan(Admin).

Keterangan gambar : pada saat admin ingin melihat daftar tanggapan operator prodi yang masuk, admin akan memilih menu Respon, dan sistem akan memanggil seluruh data tentang tanggapan di dalam database tanggapan dan menampilkannya kepada admin, apabila admin ingin menghapus data tanggapan

tersebut maka admin akan memilih opsi hapus pada sistem dan sistem akan menghapus data tanggapan yang di pilih di dalam database tanggapan.

4.2.3.9 Squence Diagram fungsi lihat dan hapus akun petugas di sistem(Admin).

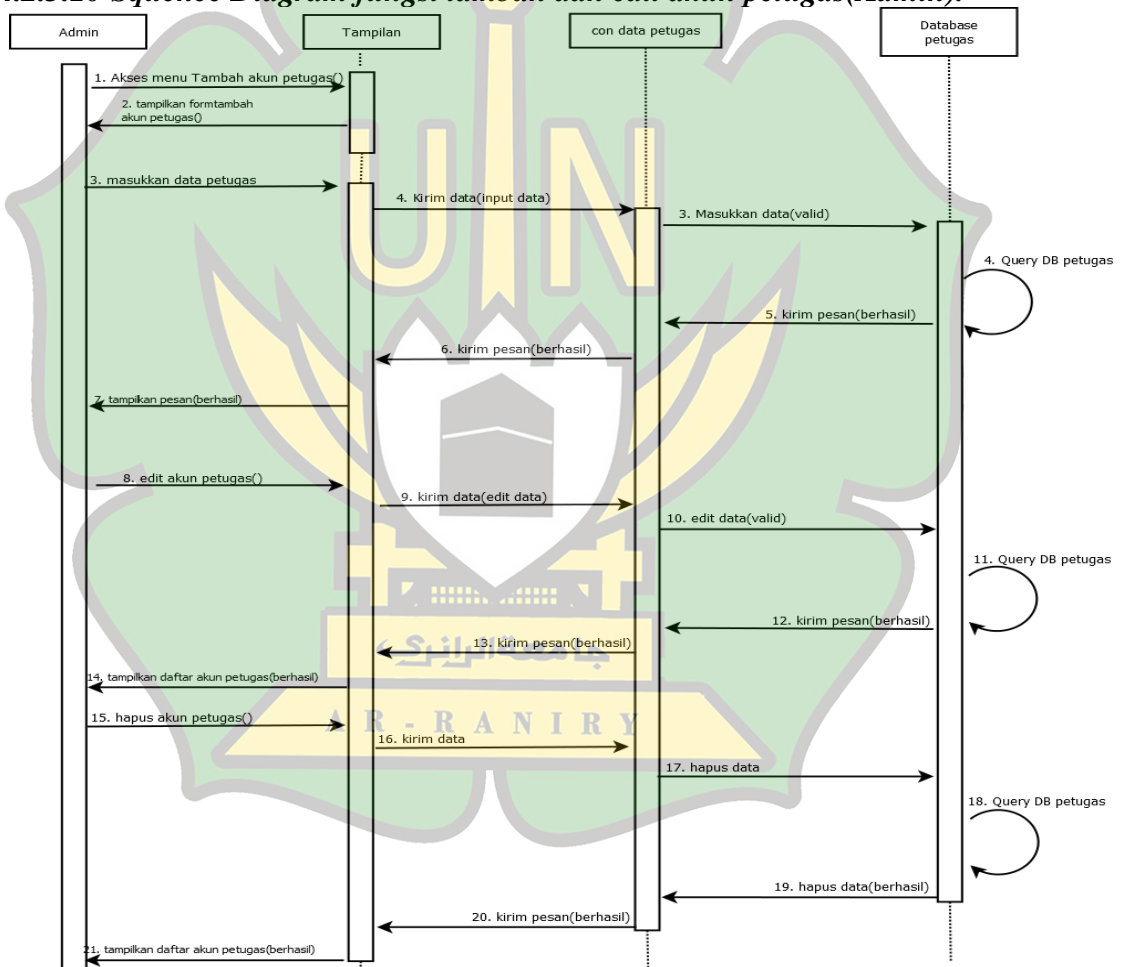


Gambar 4.20 .Squence Diagram fungsi lihat dan hapus akun petugas di sistem(Admin)

Keterangan gambar : Pada saat admin ingin melihat daftar akun petugas(Operator Prodi dan Admin),yang pertama di lakukan adalah admin

meminta kepada sistem untuk menampilkan seluruh data akun petugas yang telah terdata dan setelahnya sistem akan mencari semua data akun petugas di dalam database petugas dan menampilkannya. Apabila admin ingin menghapus data akun petugas tersebut maka admin akan memilih opsi hapus pada sistem dan sistem akan menghapus data akun petugas yang di pilih di dalam *database* petugas.

4.2.3.10 Sequence Diagram fungsi tambah dan edit akun petugas(Admin).

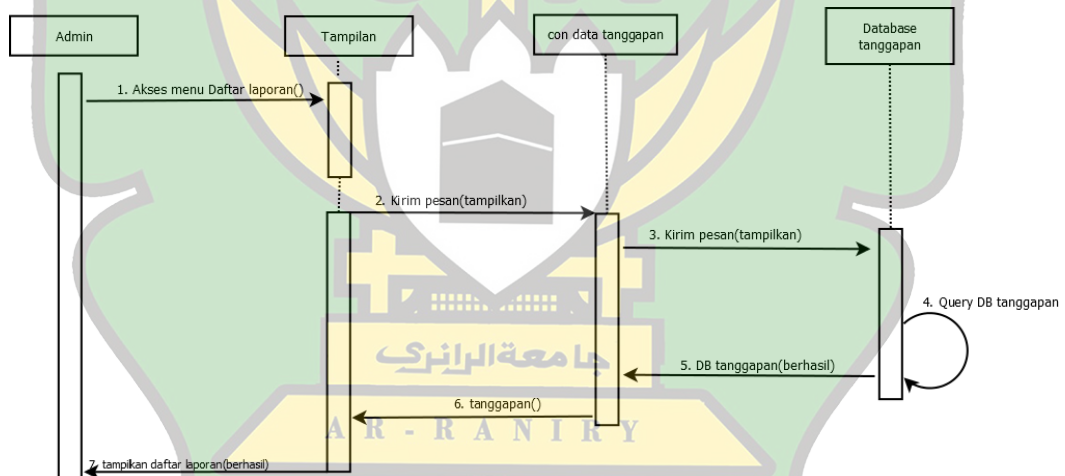


Gambar 4.21 Squence Diagram fungsi tambah dan edit akun petugas(Admin).

Keterangan gambar : pada saat admin ingin menambahkan akun petugas, admin akan memilih menu tambah akun petugas, setelahnya sistem akan

menampilkan form penambahan akun ke admin, admin memasukkan data-data petugas yang akan di buat, sistem akan memasukkan data ke dalam database petugas dan mengirimkan pesan bahwa penambahan akun telah berhasil di lakukan. Dan apabila admin ingin mengubah data yang telah di buat sebelumnya, maka admin akan memilih opsi edit di dalam sistem, sistem akan menampilkan detail akun petugas dan admin bisa mengubah data di dalamnya dan ketika admin telah selesai menyimpan, admin akan memilih menu simpan dan sistem akan menyimpan data yang telah berubah kembali ke database petugas.

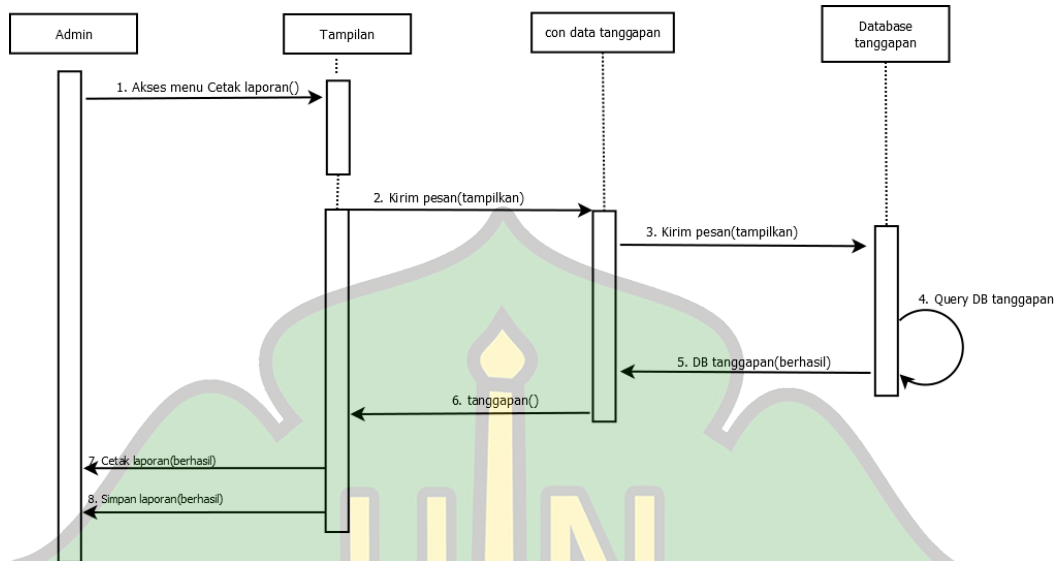
4.2.3.11 Sequence Diagram fungsi lihat laporan(Admin).



Gambar 4.22 Sequence Diagram fungsi lihat laporan(Admin).

Keterangan gambar : Pada saat admin ingin melihat Laporan, yang pertama di lakukan adalah admin meminta kepada sistem untuk menampilkan seluruh data pengaduan yang telah di tanggapi yang telah terdata dan setelahnya sistem akan mencari semua data pengaduan yang telah di tanggapi di dalam database tanggapan dan menampilkannya.

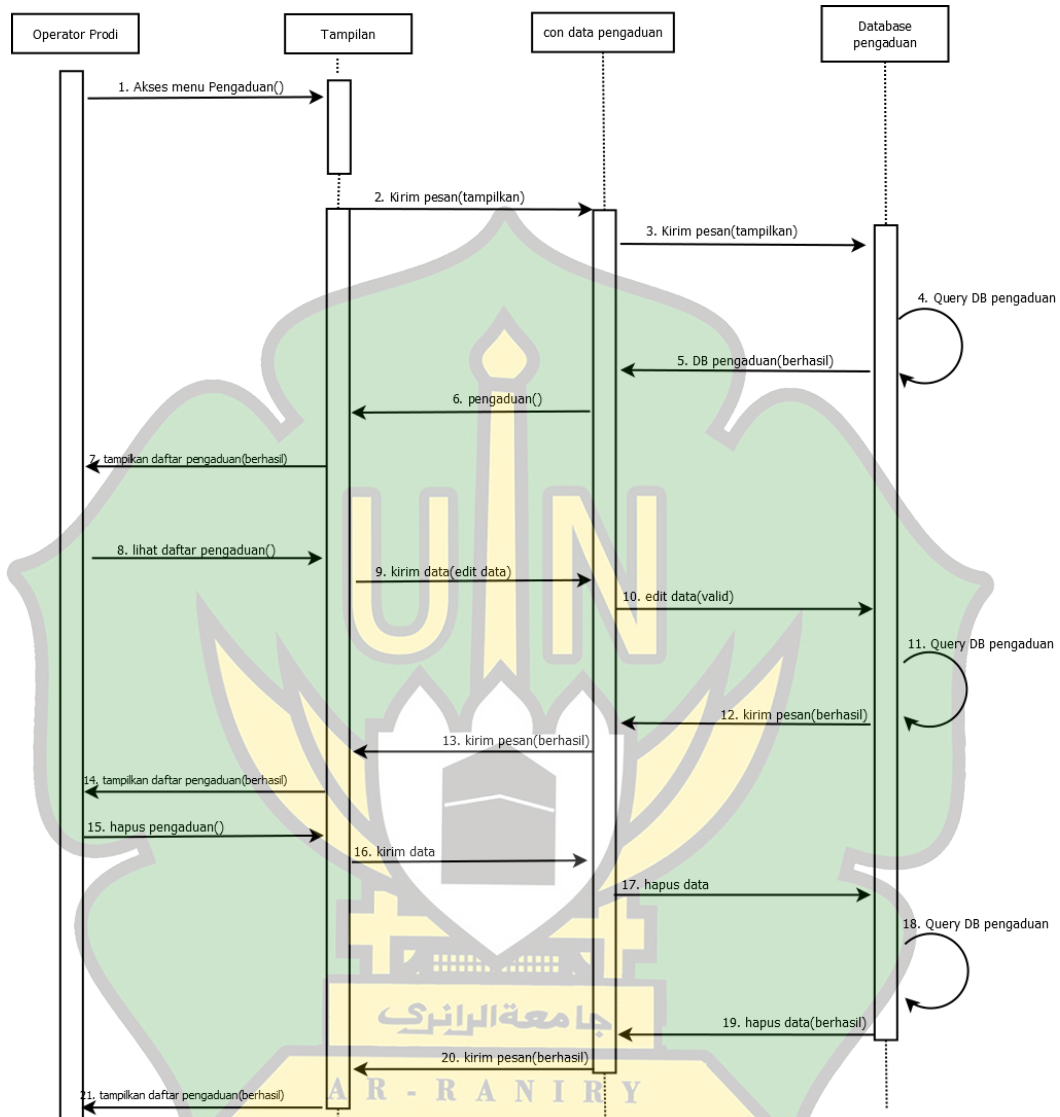
4.2.3.12 Sequence Diagram fungsi cetak laporan(Admin).



Gambar 4.23 Sequence Diagram fungsi cetak laporan(Admin).

Keterangan gambar : Pada saat admin ingin mencetak Laporan, yang pertama dilakukan adalah admin meminta kepada sistem untuk menampilkan seluruh data pengaduan yang telah di tanggapi yang telah terdata dan setelahnya sistem akan mencari semua data pengaduan yang telah di tanggapi di dalam database tanggapan dan menampilkannya, setelah di tampilkan admin bisa memilih opsi cetak yang ada di dalam sistem dan sistem akan mengambil seluruh data pengaduan yang ada di database tanggapan dan di sajikan dalam sebuah file berisikan tabel tentang segala data pengaduan yang telah di tanggapi dan admin bis mencetak langsung ataupun menyimpannya dalam bentuk *file*.

4.2.3.13 Squence Diagram fungsi lihat dan hapus pengaduan(Operator Prodi).

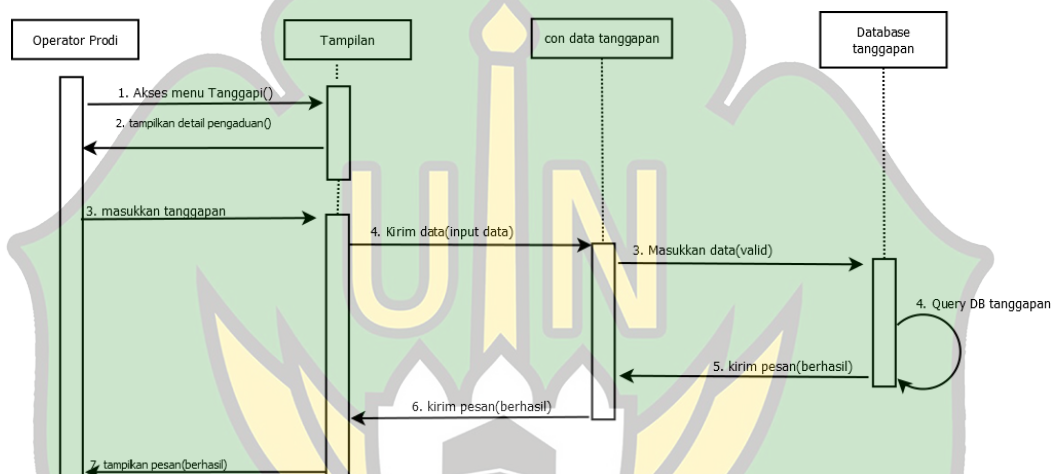


Gambar 4.24 Squence Diagram fungsi lihat dan hapus pengaduan(Operator Prodi).

Keterangan gambar : Pada saat Operator Prodi ingin melihat daftar pengaduan yang masuk, operator prodi akan memilih menu pengaduan, dan sistem akan memanggil seluruh data tentang pengaduan di dalam database pengaduan dan menampilkannya kepada operator prodi, apabila operator prodi

ingin menghapus data pengaduan tersebut maka operator prodi akan memilih opsi hapus pada sistem dan sistem akan menghapus data pengaduan yang di pilih di dalam database pengaduan.

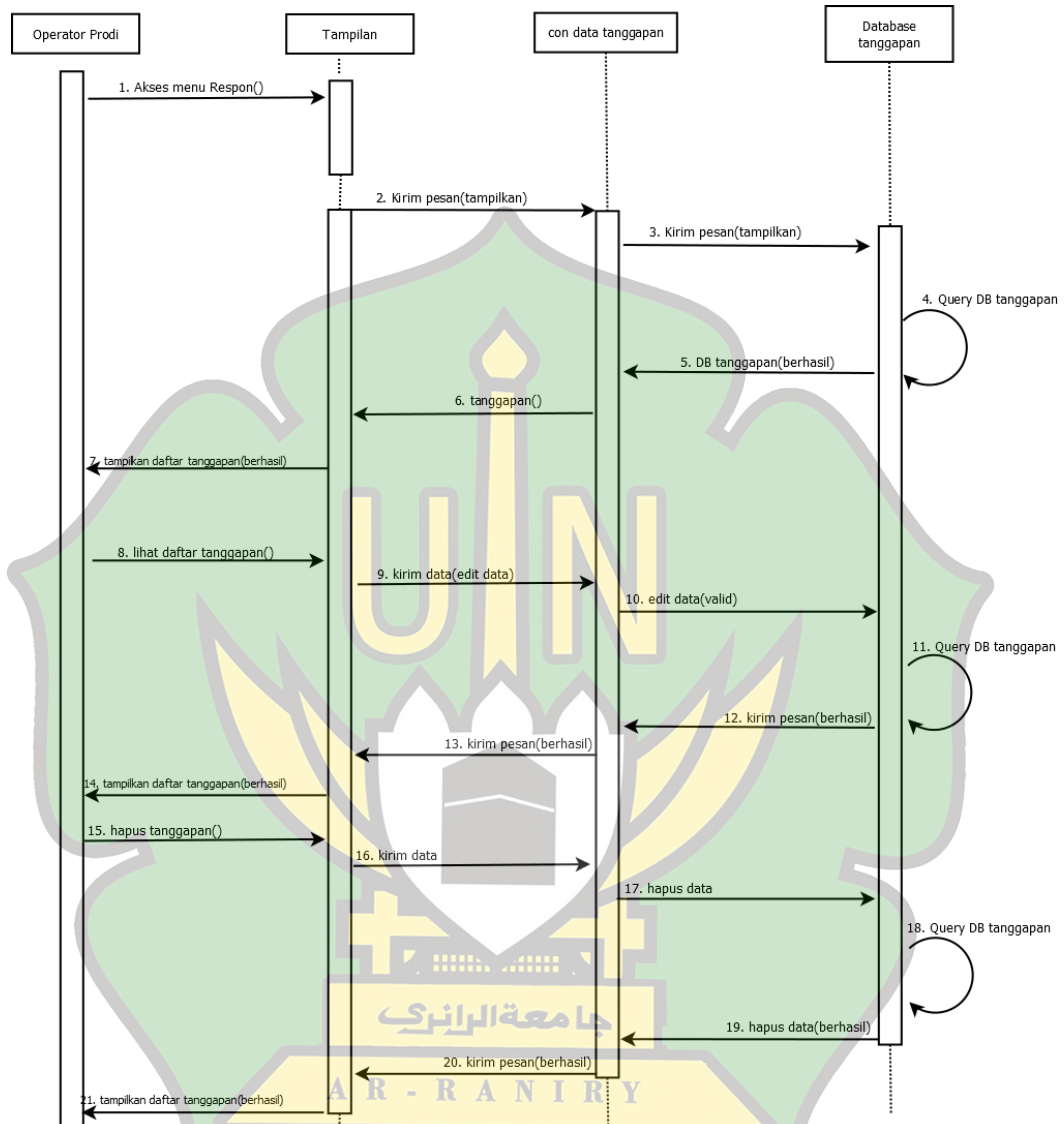
4.2.3.14 Squence Diagram fungsi memberikan tanggapan terhadap pengaduan mahasiswa(Operator Prodi).



Gambar 4.25 Squence Diagram fungsi memberikan tanggapan terhadap pengaduan mahasiswa(Operator Prodi).

Keterangan gambar : pada saat Operator Prodi ingin menanggapi salah satu pengaduan, Operator Prodi akan memilih menu pengaduan dan melihat pengaduan tersebut terlebih dahulu, setelah itu Operator Prodi akan memilih menu menanggapi pengaduan yang ada pada sistem dan sistem akan menampilkan form untuk memberikan tanggapan terhadap pengaduan, setelah selesai memberikan tanggapan Operator Prodi akan menekan tombol kirim pada sistem dan sistem akan menyimpan data tanggapannya dalam *database* tanggapan.

4.2.3.15 Squence Diagram fungsi lihat dan hapus tanggapan(Operator Prodi).

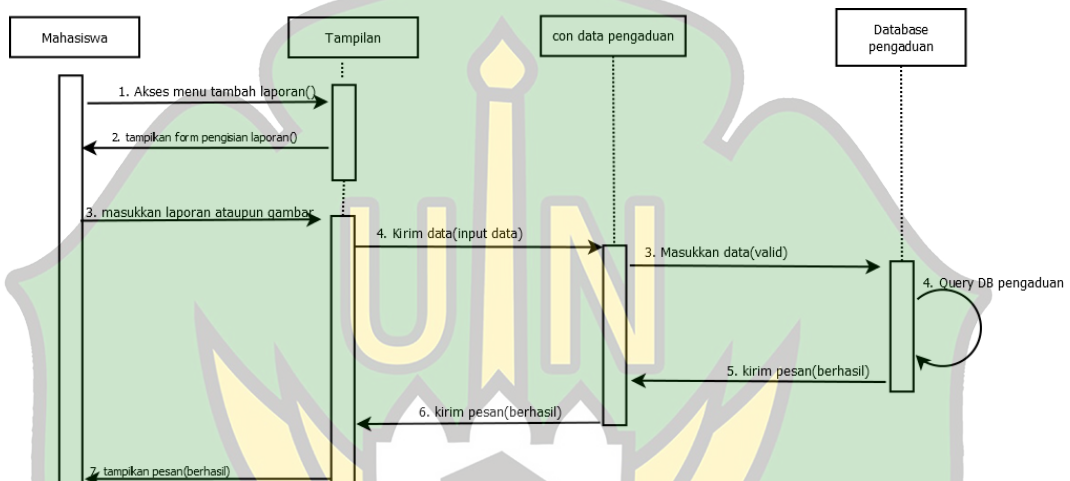


Gambar 4.26 Squence Diagram fungsi lihat dan hapus tanggapan(Operator Prodi).

Keterangan gambar : Pada saat Operator Prodi ingin melihat daftar tanggapan yang masuk, operator prodi akan memilih menu laporan, dan sistem akan memanggil seluruh data tentang laporan di dalam database tanggapan dan menampilkannya kepada operator prodi, apabila operator prodi ingin menghapus

data tanggapan tersebut maka operator prodi akan memilih opsi hapus pada sistem dan sistem akan menghapus data tanggapan yang di pilih di dalam database tanggapan.

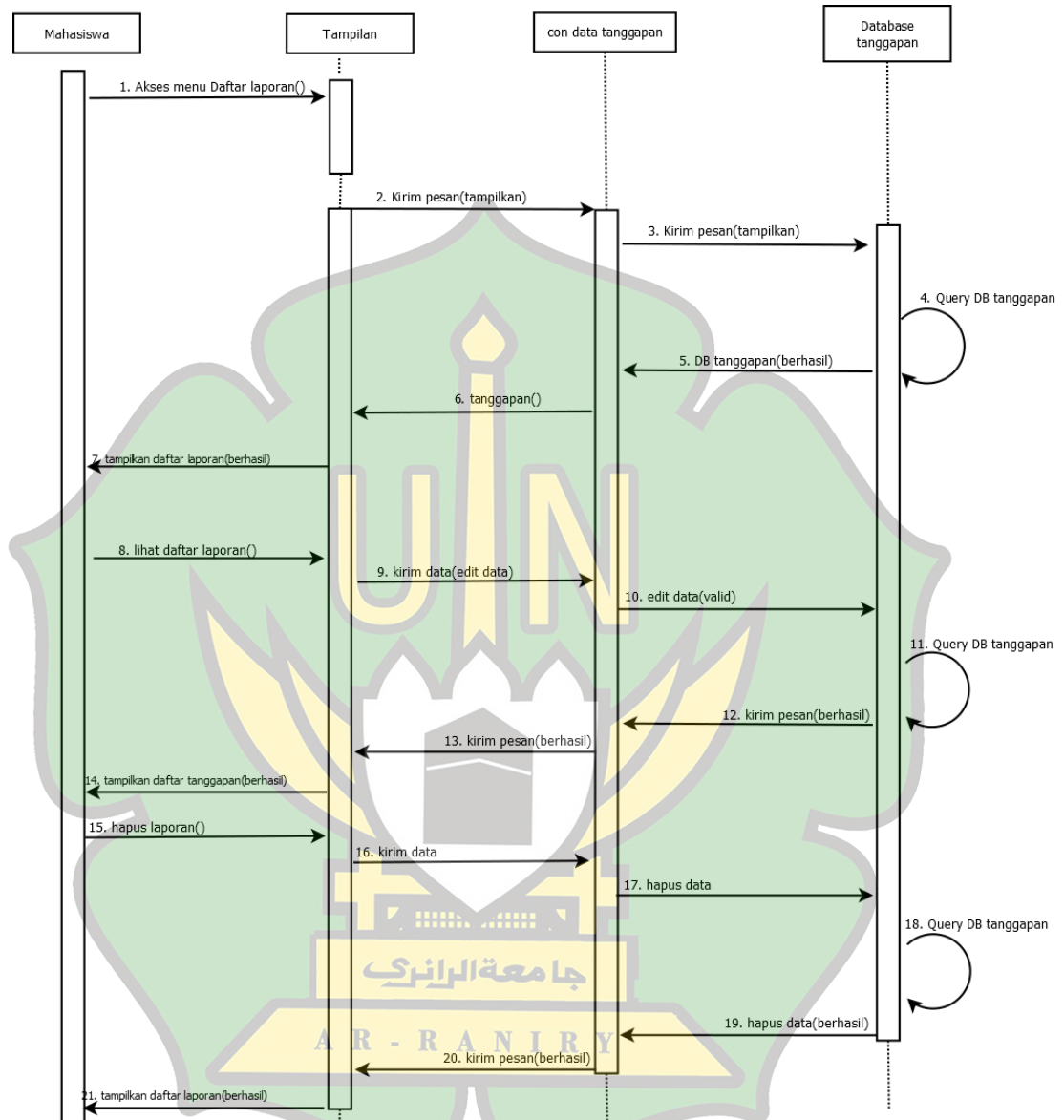
4.2.3.16 Squence Diagram fungsi Tambah Laporan(Mahasiswa).



Gambar 4.27 Squence Diagram fungsi Tambah Laporan(Mahasiswa).

Keterangan gambar : Pada saat Mahasiswa ingin memberikan pengaduan, mahasiswa akan memilih menu tambah laporan, sistem akan menampilkan form pengisian laporan, mahasiswa memasukkan laporan dan di tambah gambar jika perlu, sistem akan memsukkan pengaduan ke dalam database pengaduan dan sistem akan menampilkan pesan bahwa pengaduan telah berhasil di kirim dan akan segera di tanggapi.

4.2.3.17 Squence Diagram fungsi lihat dan hapus laporan(Mahasiswa).

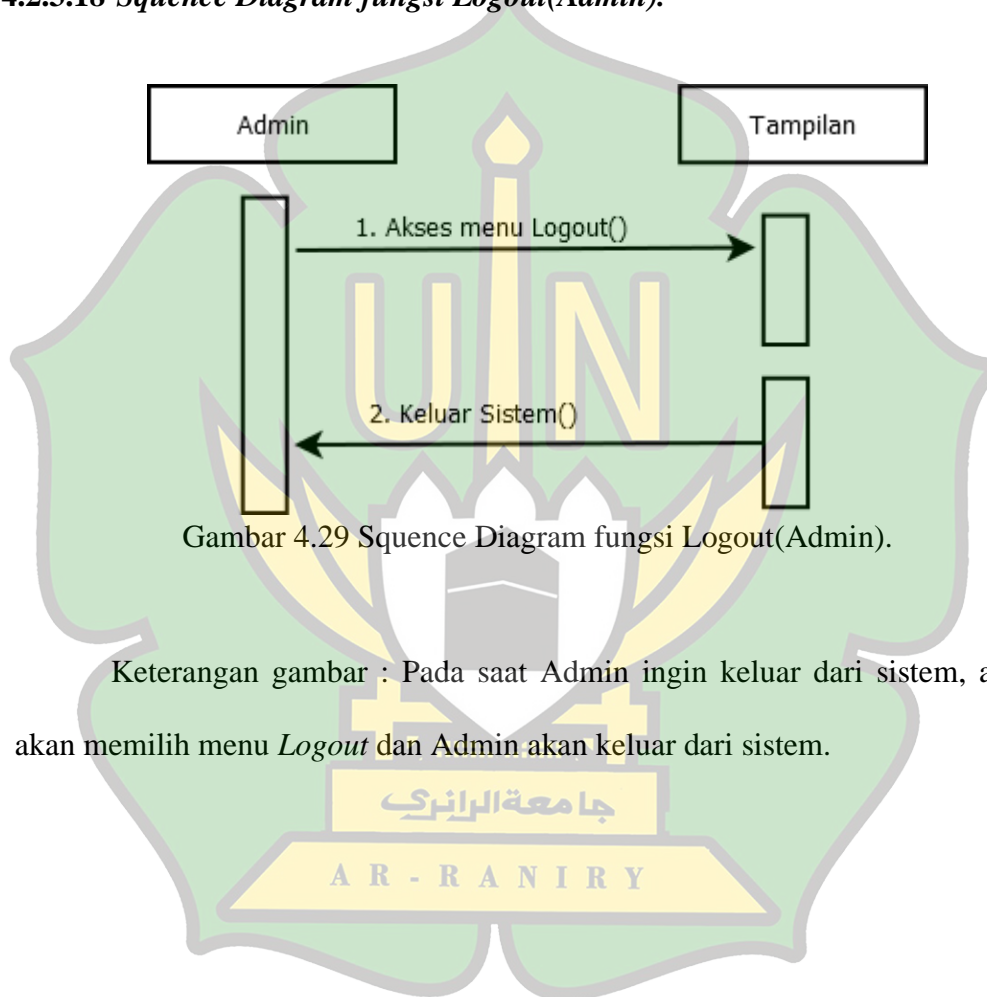


Gambar 4. 28 Squence Diagram fungsi lihat dan hapus laporan(Mahasiswa).

Keterangan gambar : Pada saat Mahasiswa ingin melihat daftar pengaduan yang telah di tanggapi, Mahasiswa akan memilih menu daftar laporan, dan sistem akan memanggil seluruh data tentang laporan di dalam database tanggapan dan menampilkannya kepada mahasiswa, apabila mahasiswa

ingin menghapus data pengaduan tersebut maka mahasiswa akan memilih opsi hapus pada sistem dan sistem akan menghapus data pengaduan yang di pilih di dalam database pengaduan.

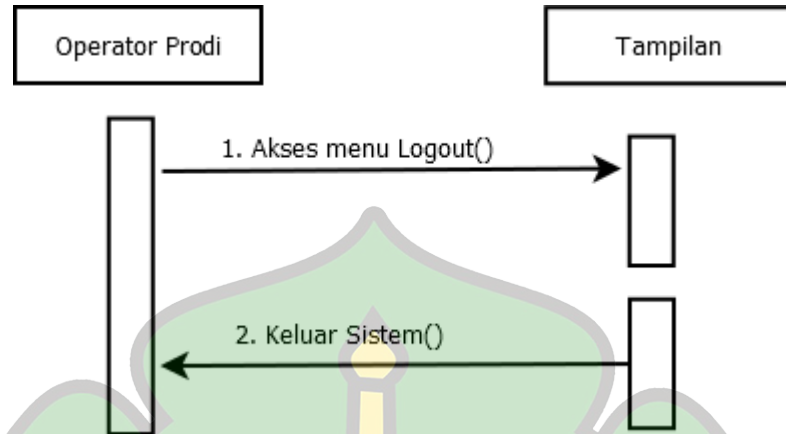
4.2.3.18 Sequence Diagram fungsi Logout(Admin).



Gambar 4.29 Sequence Diagram fungsi Logout(Admin).

Keterangan gambar : Pada saat Admin ingin keluar dari sistem, admin akan memilih menu *Logout* dan Admin akan keluar dari sistem.

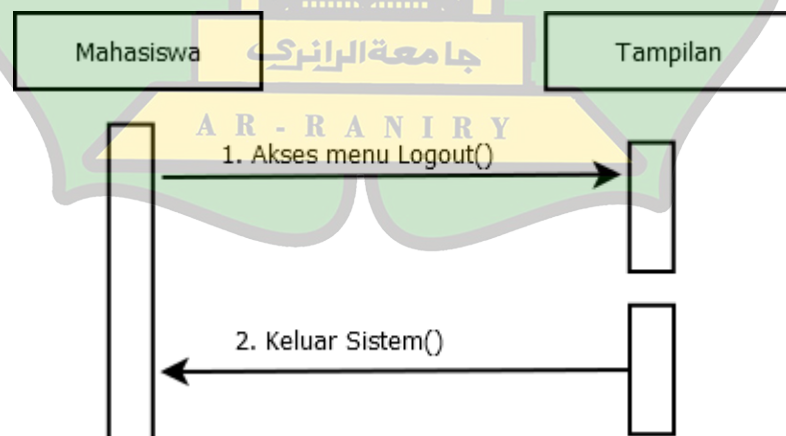
4.2.3.19 Sequence Diagram fungsi Logout(Operator Prodi).



Gambar 4.30 Squence Diagram fungsi Logout(Operator Prodi).

Keterangan gambar : Pada saat Operator prodi ingin keluar dari sistem, Operator Prodi akan memilih menu *Logout* dan Operator Prodi akan keluar dari sistem.

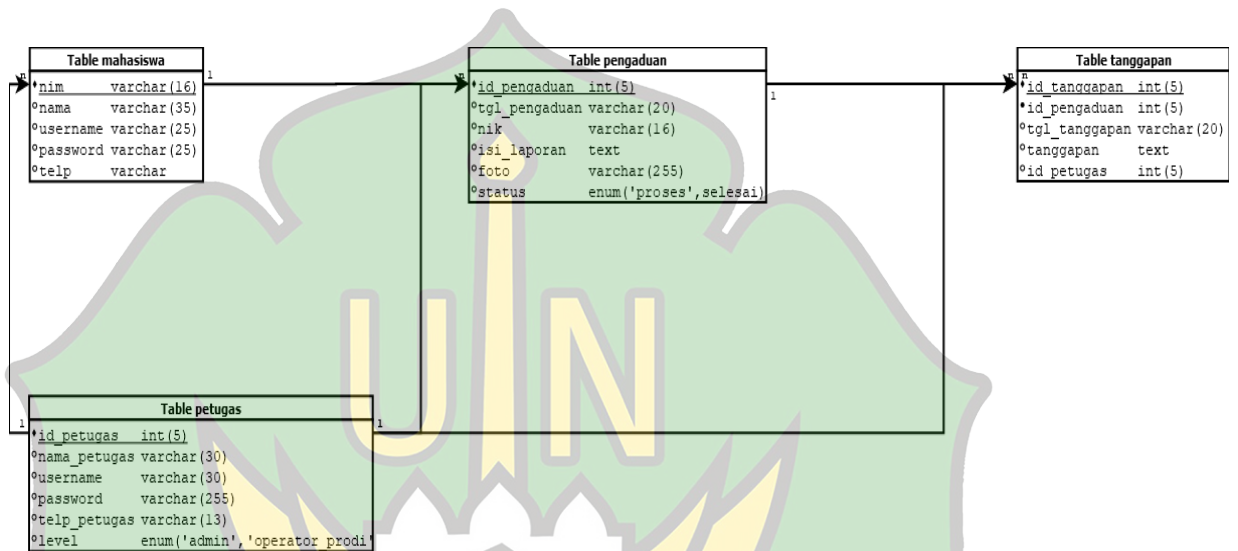
4.2.3.20 Sequence Diagram fungsi Logout(Mahasiswa).



Gambar 4.31 .Squence Diagram fungsi Logout(Mahasiswa).

Keterangan gambar : Pada saat Mahasiswa ingin keluar dari sistem, mahasiswa akan memilih menu *Logout* dan mahasiswa akan keluar dari sistem.

4.2.4 Membuat Class Diagram



Gambar 4.32 Class Diagram.

Spesifikasi tabel :

1. Tabel petugas

Nama : petugas

Type File : Tabel petugas

Primary Key : id_petugas

Foreign Key :-

Nama Field	Tipe data	Ukuran	Keterangan
id_petugas	varchar	30	Id admin dan operator prodi
nama_petugas	varchar	30	Nama admin dan operator prodi
Username	varchar	30	username
Password	varchar	255	Password admin dan operator prodi

telp_petugas	varchar	13	Nomer telpon admin dan petugas prodi
level	enum('admin','operator_prodi')		Level user petugas di dalam sistem yaitu admin dan operator prodi

Tabel 4.12 Tabel Petugas

2. Tabel mahasiswa

Nama : mahasiswa

Type File : Tabel mahasiswa

Primary Key : nim

Foreign Key : -

Nama Field	Type data	Ukuran	Keterangan
nim	varchar	16	Nim mahasiswa untuk membedakan satu akun dengan akun yang lainnya
nama	varchar	35	Nama mahasiswa
username	varchar	30	Username mahasiswa
password	varchar	255	Password mahasiswa
telp	varchar	13	Nomer telpon mahasiswa

Tabel 4.13 Tabel mahasiswa

3. Tabel pengaduan

Nama : pengaduan

Type File : Tabel pengaduan

Primary Key : id_pengaduan

Foreign Key : nim

Nama Field	Tipe data	Ukuran	Keterangan
id_pengaduan	Int	5	Nomer id pengaduan
tgl_pengaduan	Varchar	20	Tanggal pengaduan di lakukan
Nim	Varchar	16	nim pelapor
Isi_laporan	text		isi pengaduan
Foto	Varchar	255	Gambar pendukung pengaduan
Status	enum('proses','selesai')		Status dari pengaduan apakah masih di proses atau sudah selesai

Tabel 4. 14 Tabel pengaduan

4. Tabel tanggapan

Nama : tanggapan

Type File : Tabel tanggapan

Primary Key : id_tanggapan

Foreign Key : id_pengaduan,id_petugas

Nama Field	Tipe data	Ukuran	Keterangan
id_tanggapan	int	5	Nomer id tanggapan
id_pengaduan	int	5	Nomer id pengaduan
tgl_tanggapan	varchar	20	Tanggal pemberian tanggapan di lakukan
tanggapan	text		isi tanggapan
id_petugas	int	5	id petugas yang menanggapi

Tabel 4.15 Tabel tanggapan

4.2.5 Desain Interface

Interface design yang dimaksud adalah tampilan untuk merancang Sistem pelaporan berbasis web agar pasti bagaimana tata letak dari tampilan sistem. Berikut merupakan *interface* untuk Sistem pelaporan berbasis web yang akan di rancang:

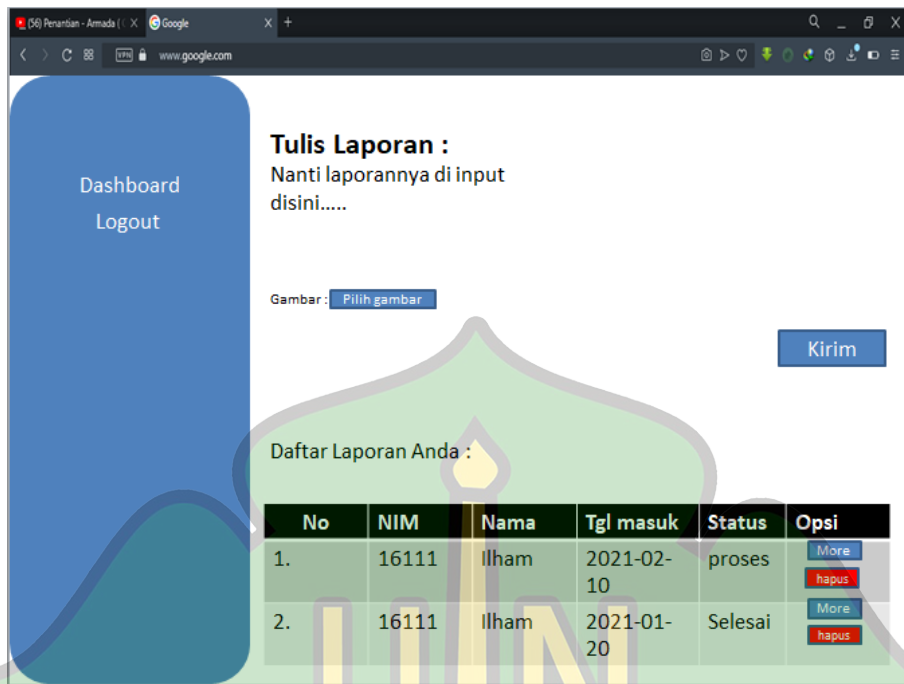


Gambar 4. 33 Tampilan *login* (admin, Operator, mahasiswa)



Gambar 4.34 Tampilan beranda admin

Gambar 4.35 Tampilan beranda operator prodi



Gambar 4.36 Tampilan beranda untuk mahasiswa

4.3 Implementasi Sistem

Setelah desain sistem telah dilakukan, maka tahap selanjutnya adalah mengimplementasikan hasil rancangan tersebut. Dalam pelaksanaan pembuatan sistemnya dengan menggunakan XAMPP versi 3.2.4. Selain dari itu digunakan juga aplikasi *Sublime Text 3* sebagai perangkat lunak untuk *Text Editor* dan juga *Microsoft Visio 2013* untuk peralatan membuat UML. Dan sarana pendukung lainnya untuk memperlancar agar sistem dapat berjalan dengan sesuai perencanaan.

4.3.1 Perangkat Keras

Perangkat Keras yang di butuhkan untuk menjalankan sistem ini memiliki spesifikasi sebagai berikut:

1. *Server*

- a. *Processor* : setara dengan core i3 ataupun di atasnya
- b. *Memory* : Minimal 256 MB ataupun lebih
- c. *Harddisk* : 30 GB

2. *Client*

- a. *Processor* : setara dengan core i3 ataupun di atasnya
- b. *Memory* : Minimal 256 MB ataupun lebih
- c. *Harddisk* : 30 GB

4.3.2 Perangkat Lunak

Dan untuk di perangkat lunaknya di butuhkan spesifikasi sebagai berikut:

1. *Server*

- a. *Microsoft Windows 2007* ataupun di atasnya
- b. *XAMPP version 3.2.4* yang memiliki fitur *Apache,PHP* dan juga *MySQL*
- c. *Browser* apa saja bisa mendukung.

2. *Client*

- a. *Microsoft Windows 2007* ataupun di atasnya
- b. *Browser* apa saja bisa mendukung.

4.3.3 Pengujian Sistem

Di tahap pengujian sistem digunakan model pengujian *Blackbox testing*.

Yaitu dengan langkah menjalankan sistemnya selanjutnya melihat keluaran ataupun *output* yang di tampilkan oleh sistem tersebut apakah sudah sesuai

dengan yang di harapkan. Hasil pengujiannya di tampilkan dengan menggunakan beberapa *level user* di sistem seperti berikut :

4.3.3.1 Integration Testing

1. Integration Testing (Mahasiswa)

No.	Test Case	Keluaran	Hasil
1.	Masukkan alamat berikut http://localhost/ptireport/ . Pada <i>browser</i> anda.	Tampil halaman <i>Login</i>	Sesuai
2.	Masukkan <i>Username</i> dan <i>Password</i>	<ul style="list-style-type: none"> - Jika login benar maka tampil Beranda mahasiswa - Jika login salah tampil pesan kesalahan 	Sesuai
3.	Tulis pelaporan	<i>Text box</i> tulis laporan akan aktif	Sesuai
4.	Tekan tombol pilih gambar	Tampil <i>File explorer</i> untuk mencari gambar yang di inginkan	Sesuai
5.	Tekan tombol kirim	Tampil pesan mengatakan pengaduan akan segera di tanggapi	Sesuai
6.	Tekan tombol “More”	Tampil detail laporan yang telah di tanggapi	Sesuai
7.	Tekan tombol “Hapus”	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data laporan - Jika Cancel, sistem akan membatalkan penghapusan data 	Sesuai
8.	Pilih menu <i>Logout</i>	Keluar dari sistem	Sesuai

Tabel 4.16 Tabel Integration testing (Mahasiswa)

2. Integration Testing (Admin)

No.	Test Case	Keluaran	Hasil
1.	Masukkan alamat berikut http://localhost/ptireport/ . Pada <i>browser</i> anda.	Tampil halaman <i>Login</i>	Sesuai
2.	Masukkan <i>Username</i> dan <i>Password</i>	<ul style="list-style-type: none"> - Jika login benar maka tampil 	Sesuai

		<p>Beranda Admin</p> <ul style="list-style-type: none"> - Jika login salah tampil pesan kesalahan 	
3.	Pilih menu Registrasi	Tampil daftar mahasiswa yang sudah memiliki akun pelaporan	Sesuai
	3.1 Pilih menu edit pada salah satu akun mahasiswa	Tampil form untuk mengedit data akun Mahasiswa	Sesuai
	3.2 Pilih menu Hapus pada salah satu akun mahasiswa	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data mahasiswa - Jika Cancel, sistem akan membatalkan penghapusan data mahasiswa 	Sesuai
	3.3 pilih menu tambah	Tampil form untuk menambahkan akun mahasiswa yang baru	Sesuai
4.	Pilih menu Pengaduan	Tampil daftar aduan yang sudah selesai maupun yang masih tahap proses	Sesuai
	4.1 Pilih menu “More” pada salah satu data pengaduan	Tampil detail Laporan Pengaduan	Sesuai
	4.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data laporan - Jika Cancel, sistem akan membatalkan penghapusan data 	Sesuai
5.	Pilih menu Respon	Tampil data tanggapan operator prodi terhadap pengaduan yang masuk	Sesuai
	5.1 Pilih menu “More” pada salah satu data pengaduan	Tampil detail Respon	Sesuai
	5.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data Respon - Jika Cancel, sistem akan membatalkan penghapusan data 	Sesuai
6.	Pilih Menu User	Tampil seluruh daftar petugas yang ada di sistem termasuk operator prodi dan admin	Sesuai
	6.1 Pilih menu edit pada salah satu akun	Tampil form untuk mengedit data akun	Sesuai

	6.2 Pilih menu Hapus pada salah satu akun	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data - Jika Cancel, sistem akan membatalkan penghapusan data akun 	Sesuai
	6.3 pilih menu tambah	Tampil form untuk menambahkan akun petugas yang baru	Sesuai
7.	Pilih menu Laporan	Tampil seluruh data pengaduan yang telah di tanggap	Sesuai
	7.1 Pilih menu "More" pada salah satu data pengaduan	Tampil detail Laporan	Sesuai
	7.2 Klik tombol <i>Printer</i>	<ul style="list-style-type: none"> - Tampil <i>preview</i> cetak pada sistem - menyimpan daftar pengaduan 	Sesuai
8.	Pilih menu Logout	Keluar dari sistem	Sesuai

Tabel 4.17 Tabel Integration testing (Admin).

3. Integration Testing (Operator Prodi)

No.	Test Case	Keluaran	Hasil
1.	Masukkan alamat berikut http://localhost/ptireport/ . Pada <i>browser</i> anda.	Tampil halaman <i>Login</i>	Sesuai
2.	Masukkan <i>Username</i> dan <i>Password</i>	<ul style="list-style-type: none"> - Jika login benar maka tampil Beranda Operator prodi - Jika login salah tampil pesan kesalahan 	Sesuai
3.	Pilih menu Pengaduan	Tampil daftar aduan yang sudah selesai maupun yang masih tahap proses	Sesuai
	3.1 Pilih menu "More" pada salah satu data pengaduan	Tampil detail Laporan Pengaduan	Sesuai
	3.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data pengaduan - Jika Cancel, sistem akan membatalkan penghapusan data 	Sesuai

	3.3 tambahkan tanggapan	Text Box tanggapan akan aktif	Sesuai
	3.4 kirim tanggapan	Muncul pesan bahwa tanggapan telah di kirim	Sesuai
4.	Pilih menu Respon	Tampil data tanggapan operator prodi terhadap pengaduan yang masuk	Sesuai
	4.1 Pilih menu "More" pada salah satu data pengaduan	Tampil detail Respon	Sesuai
	4.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data Respon - Jika Cancel, sistem akan membatalkan penghapusan data 	Sesuai
5.	Pilih menu Logout	Keluar dari sistem	Sesuai

Tabel 4.18 Integration sistem (Operator Prodi)

4.3.3.2 User Acceptance Testing

No.	User Acceptance
1.	Tambahkan fitur ubah <i>password</i> pada mahasiswa
2.	Tambahkan notifikasi pengaduan yang sedang di proses
3.	Tambahkan tanda tangan di laporan pengaduan
4.	Gabungkan pendaftaran mahasiswa dengan operator prodi
5.	Warna di sesuaikan dengan tema prodi

Tabel 4.19 User acceptance testing

Hasil User Acceptance Testing didapat dari pengujian yang dilakukan

oleh:

1. Nama Penguji : Mushaddiq

Tanggal : 17 Juni 2021

Jam : 13.00-13.30

2. Nama Penguji : Nurul Fajri

Tanggal : 23 Juni 2021

Jam : 10:30-11.00

3. Nama Penguji : Cut Azhar Fuadi

Tanggal : 28 Juni 2021

Jam : 10.00-11.20

4.4 Uji Ahli Media

Penilaian ahli media bertujuan untuk mengetahui kekurangan dan kelebihan dari sistem pengaduan mahasiswa terhadap prodi berbasis *website* yang telah dikembangkan. Data hasil wawancara disajikan sebagai berikut:

Nama Ahli media : Rahmat Musfika, M.Kom

Tanggal : 23 Juni 2021

NO	Aspek yang Dinilai	Alternatif Pilihan				
		1 SS	2 TS	3 R	4 S	5 SS
1	Pemilihan tampilan				X	
2	Layout/tata letak tombol					X
3	Kemenarikan desain				X	
4	Ketepatan pemilihan huruf					X
5	Ketepatan ukuran huruf			X		
6	Kebergunaan untuk Prodi					X
7	Pengaturan jarak				X	
8	Kejelasan gambar			X		
9	Ukuran gambar			X		
10	Kebergunaan untuk mahasiswa					X
Jumlah						
Jumlah rata-rata						
Jumlah Persentase						

Tabel 4.20 Tabel Uji media 1

Tambahan dari ahli media : Aplikasinya sudah bagus, Cuma tambahkan satu menu di operator prodi untuk membedakan pengaduan yang belum di proses, sedang di proses dan sudah di proses.

Nama Ahli media : Hendri Ahmadian, M.I.M

Tanggal : 23 Juni 2021

NO	Aspek yang Dinilai	Alternatif Pilihan				
		1 SS	2 TS	3 R	4 S	5 SS
1	Pemilihan tampilan					V
2	Layout/tata letak tombol					V
3	Kemenarikan desain					V
4	Ketepatan pemilihan huruf					V
5	Ketepatan ukuran huruf					V
6	Kebergunaan untuk Prodi					V
7	Pengaturan jarak					V
8	Kejelasan gambar				V	
9	Ukuran gambar				V	
10	Kebergunaan untuk mahasiswa					V
Jumlah						
Jumlah rata-rata						
Jumlah Persentase						

Tabel 4.21 Tabel Uji media 2

Tambahan dari ahli media : Tambah menu daftar laporan pada akun mahasiswa.

Nama Ahli media : Bustami, M.Sc

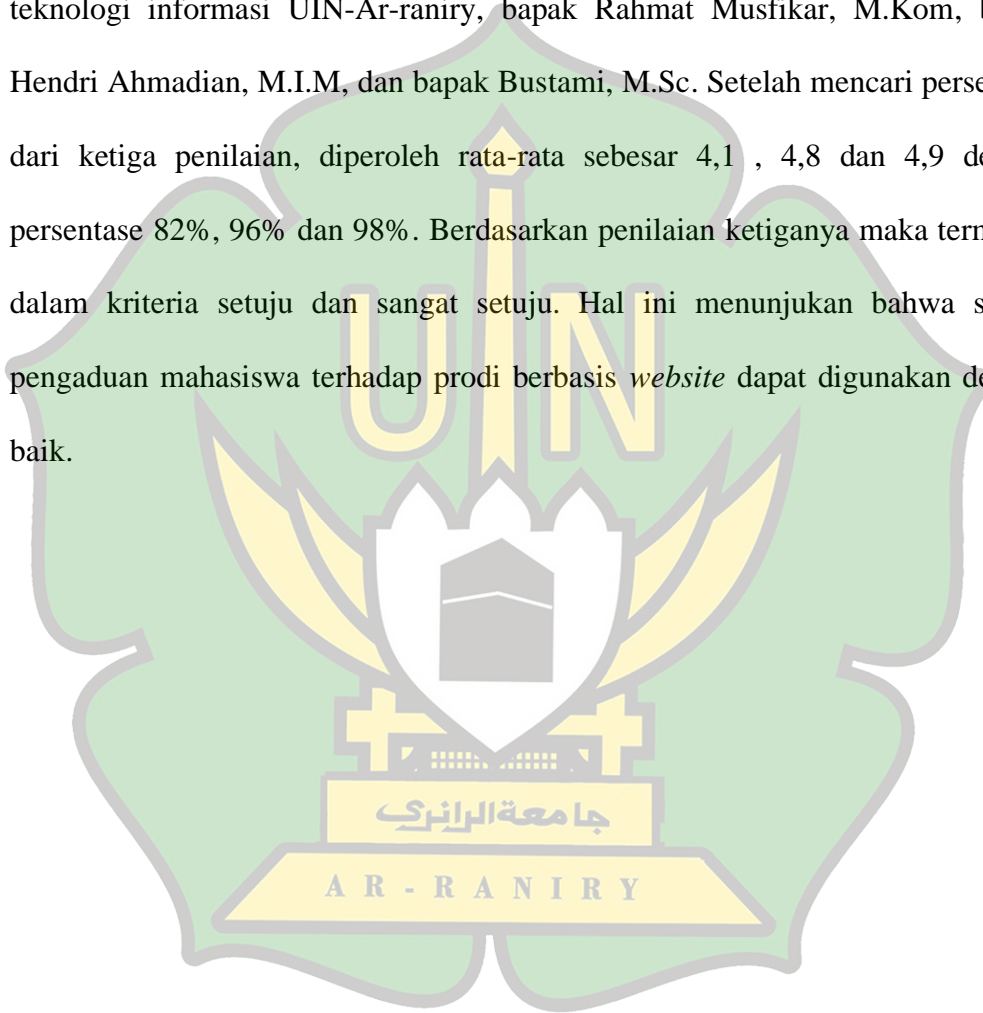
Tanggal : 30 Juni 2021

NO	Aspek yang Dinilai	Alternatif Pilihan				
		1 SS	2 TS	3 R	4 S	5 SS
1	Pemilihan tampilan				V	
2	Layout/tata letak tombol					V
3	Kemenarikan desain					V
4	Ketepatan pemilihan huruf					V
5	Ketepatan ukuran huruf					V
6	Kebergunaan untuk Prodi					V
7	Pengaturan jarak					V
8	Kejelasan gambar					V
9	Ukuran gambar					V
10	Kebergunaan untuk mahasiswa					V
Jumlah						
Jumlah rata-rata						
Jumlah Persentase						

Tabel 4.22 Tabel uji media 3

Tambahan dari ahli media : Percantik background, tambah daftar proses di akun mahasiswa, pendaftaran akun disatukan, buat jangka waktu untuk cetak laporan.

Penilaian ahli media dilakukan dengan 3 pakar, yaitu dosen pendidikan teknologi informasi UIN-Ar-raniry, bapak Rahmat Musfekar, M.Kom, bapak Hendri Ahmadian, M.I.M, dan bapak Bustami, M.Sc. Setelah mencari persentase dari ketiga penilaian, diperoleh rata-rata sebesar 4,1 , 4,8 dan 4,9 dengan persentase 82%, 96% dan 98%. Berdasarkan penilaian ketiganya maka termasuk dalam kriteria setuju dan sangat setuju. Hal ini menunjukkan bahwa sistem pengaduan mahasiswa terhadap prodi berbasis *website* dapat digunakan dengan baik.



BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil perancangan dan penelitian, dapat disimpulkan bahwa:

1. Sistem pengaduan mahasiswa terhadap prodi berbasis *website* dikembangkan berdasarkan model pengembangan *RAD* yang terdiri dari 3 tahapan yaitu: Rencana Kebutuhan (*Requirement Planning*), Proses Desain Sistem (*Design System*), Implementasi .
2. Berdasarkan hasil uji coba kepada mahasiswa, operator prodi dan admin dengan menggunakan uji *testing Blackbox* diketahui semua fungsi di dalam sistem berjalan dengan baik dan layak digunakan.
3. Berdasarkan hasil uji media yang dilakukan oleh 3 penguji ahli, didapatkan hasil persentase 92%, sistem pengaduan mahasiswa terhadap prodi berbasis *website* termasuk kedalam kategori sangat setuju, sehingga dapat ditarik kesimpulan bahwa Sistem pengaduan mahasiswa terhadap prodi berbasis *website* dapat dan layak di pergunakan oleh mahasiswa dan pihak prodi.

B. Saran

Setelah melakukan penelitian ini, terdapat beberapa masukan atau saran-saran yang peneliti kemukakan, diantaranya adalah:

1. Sistem pengaduan yang dikembangkan diharapkan dapat dilanjutkan kembali sampai uji pemakaian atau produksi massal.
2. Sistem pengaduan perlu pengembangan di bagian fitur-fitur dan tampilan.

3. Pengujian hanya dilakukan oleh beberapa subjek dan ahli media, sehingga perlu penambahan responden agar tanggapan dari responden lebih baik.



DAFTAR PUSTAKA

- [1] E. F. Deddi Akbar Riyanto, Setiyawan Asegaf, “Perancangan1Aplikas Sistem1Informasi Goeografis (SIG)1Lokasi Minimarket1Di Kota1Jambi Berbasis Android.,” *J. Ilm. Media SISFO Vol.9 No.II*, vol. II, 2015.
- [2] S. Mulyani, “perancangan1metode1dan analisis,” 2017.
- [3] A. Moeharto, M., Hasan, S. and Ambarita, “Penggunaan Model1E-learning dalam Meningkatkan1Hasil Belajar1Mahasiswa pada Mater1Microprocessor.,” *Indones. J. Inf. Syst.*, vol. II.
- [4] Sudioanto, *Rancang Bangun Sistem Administrasi Pelaporan Masyarakat Pada Kantor Polisi Sektor Bontoala Kota Makassar*. 2018.
- [5] Firmansyah, “Sistem Informasi Pengaduan Warga Berbasis Website (Studi Kasus : Kelurahan Siantan Tengah , Pontianak Utara),” *J. Cendikia*, vol. XIX, no. April, pp. 397–404, 2020, [Online]. Available: <https://jurnal.dcc.ac.id/index.php/JC/article/view/328>.
- [6] M. Naomi, H. Noprisson, F. I. Komputer, U. Mercu, and B. Jakarta, “Analisa Dan Perancangan Sistem Pengaduan Mahasiswa Berbasis Web (Studi Kasus : Universitas Mercu Buana Kranggan) Pendahuluan Landasan Teori,” *JUSIBI (Jurnal Sist. Inf. Dan E-Bisnis*, vol. 1, no. 5, pp. 185–193, 2019.
- [7] H. Ahmadian, I. Dwitawati, and P. T. Informasi, “Akreditasi Berbasis Web Pada Prodi Teknologi Informasi Uin Ar-Raniry,” vol. 3, pp. 88–95, 2019.
- [8] M. Munawar, M. Maisura, and C. M. Nur, “Perancangan Aplikasi Pengolahan Data Buku Dan Pengunjung Perpustakaan Pada Taman Bacaan Masyarakat Ar-Rasyid Aceh Besar,” *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 3, no. 2, p. 122, 2019, doi: 10.22373/cj.v3i2.6340.
- [9] Achmad Solichin. S.Kom, “Pemrograman Web dengan PHP dan MySQL - Achmad Solichin - Google Buku,” *Univ. Budi Luhur*, no. June, p. 85, 2016, [Online]. Available: <https://books.google.co.id/books?id=kcD4BQAAQBAJ&printsec=frontcover&dq=aplikasi+berbasis+web+dengan&hl=id&sa=X&ved=0ahUKEwib-ft80ITYAhVBrI8KHT9GD6QQ6AEIJzAA#v=onepage&q=aplikasi+berbasis+web+dengan&f=false>.
- [10] A. T. Sholeh, E. Gunadhi, and A. D. Supriatna, “Mengamankan Skrip Pada Bahasa Pemrograman PHP Dengan Menggunakan Kriptografi Base64,” *J. Algoritm.*, vol. 10, no. 1, pp. 30–38, 2013, doi: 10.33364/algoritma/v.10-1.30.
- [11] sovia dan Febio, “MEMBANGUN APLIKASI E-LIBRARY MENGGUNAKAN HTML, PHP SCRIPT, DAN MYSQL DATABASE Rini Sovia dan Jimmy Febio,” *Processor*, vol. 6, no. 2, pp. 38–54, 2011.
- [12] T. Sunarya, “Aplikasi Internet Menggunakan Website,” 2015.
- [13] S. Dharwiyanti and R. S. Wahono, “Pengantar Unified Modeling LAnguage (UML),” *IlmuKomputer.com*, pp. 1–13, 2003, [Online]. Available: <http://www.unej.ac.id/pdf/yanti-uml.pdf>.
- [14] T. A. Kurniawan, “Pemodelan Use Case (UML): Evaluasi Terhadap

- beberapa Kesalahan dalam Praktik,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 1, p. 77, 2018, doi: 10.25126/jtiik.201851610.
- [15] D. Febiharsa, I. M. Sudana, and N. Hudallah, “Uji Fungsionalitas (BlackBox Testing) Sistem Informasi Lembaga Sertifikasi Profesi (SILSP) Batik Dengan AppPerfect Web Test Dan Uji Pengguna,” *JOINED J.*, vol. 1, no. 2, pp. 117–126, 2018, [Online]. Available: <http://e-journal.ivet.ac.id/index.php/jiptika/article/view/752>.
- [16] F. Masya, Elvina, and F. M. Simanjuntak, “Sistem Pelayanan Pengaduan Masyarakat pada Divisi Humas Polri Berbasis Web,” *Semin. Nas. Apl. Teknol. Inf. 2012 (SNATI 2012)*, vol. 2012, no. Snati, pp. 1–6, 2012.
- [17] I. Syaifei, M. Kamayani, and E. Sinduningrum, “Perancangan Aplikasi Pengaduan Masyarakat Terhadap Lingkungan Di Tingkat Kelurahan,” *Pros. Semin. Nas. Teknoka*, vol. 4, no. 2502, pp. I111–I116, 2020, doi: 10.22236/teknoka.v4i0.4271.
- [18] Sugiyono, *Metode Penelitian Pendidikan: Pendekatan Kuantitatif, Kualitatif dan R&D*. 2016.
- [19] T. Wahyuningrum and D. Januarita, “Perancangan Web e-Commerce dengan Metode Rapid Application Development (RAD) untuk Produk Unggulan Desa,” vol. 2014, no. November, pp. 81–88, 2014.



LAMPIRAN

Lampiran 1 Dokumentasi Penelitian

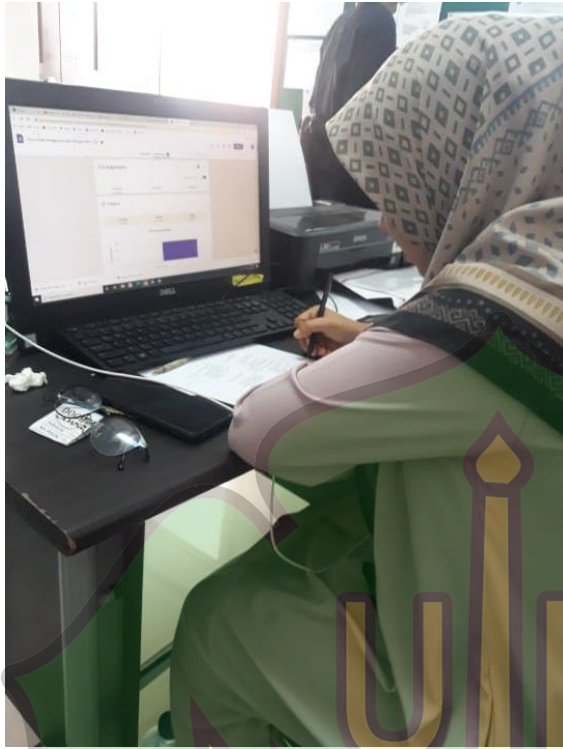
a) Wawancara dngan petugas prodi



b) Uji integritas aplikasi oleh *User*(mahasiswa)



c) Uji integrasi aplikasi dengan petugas prodi



d) Uji media oleh Bapak Hendri Ahmadian



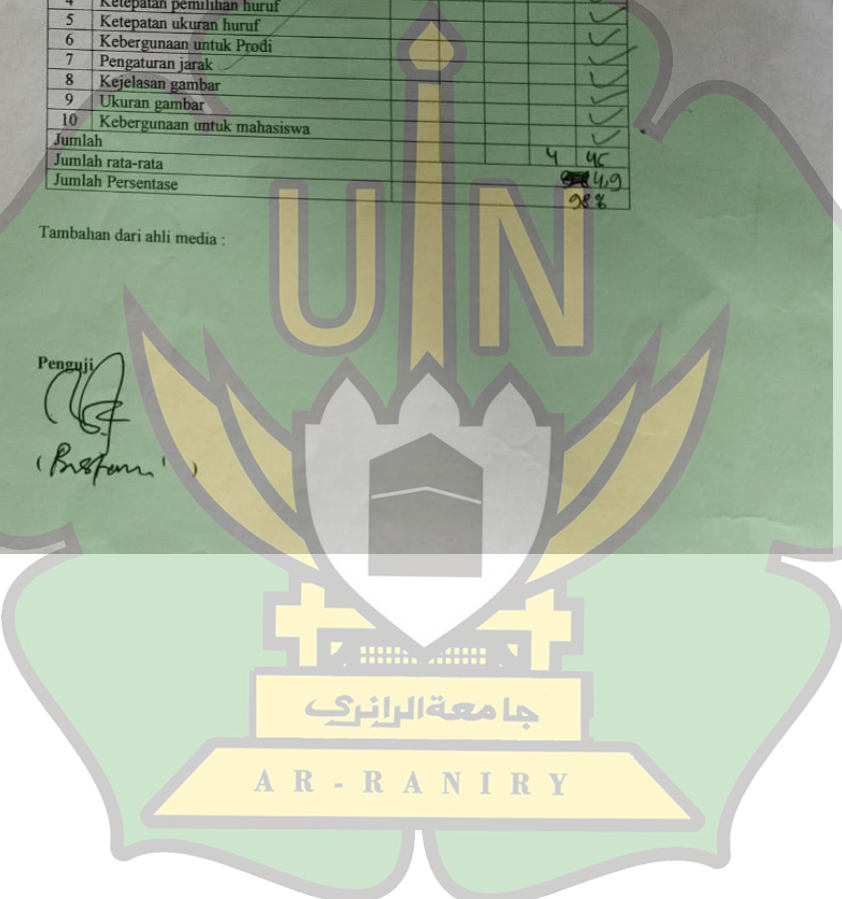
Lampiran 2 Kuisisioner Ahli Media

Nama Ahli media : *Bisfami*
 Tanggal : *9 Juni 2021*

NO	Aspek yang Dinilai	Alternatif Pilihan				
		1	2	3	4	5
1	Pemilihan tampilan	SS	TS	R	S	SS
2	Layout/tata letak tombol					
3	Kemenarikan desain					
4	Ketepatan pemilihan huruf					
5	Ketepatan ukuran huruf					
6	Kebergunaan untuk Prodi					
7	Pengaturan jarak					
8	Kejelasan gambar					
9	Ukuran gambar					
10	Kebergunaan untuk mahasiswa					
Jumlah						4 40
Jumlah rata-rata						4,0
Jumlah Persentase						28%

Tambahan dari ahli media :

Penguji
(Bisfami)



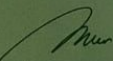
Nama Ahli media : HENDRI AHMADIAN
Tanggal : 24 / 6 / 2021

NO	Aspek yang Dinilai	Alternatif Pilihan				
		1 SS	2 TS	3 R	4 S	5 SS
1	Pemilihan tampilan				✓	
2	Layout/tata letak tombol					✓
3	Kemamarikan desain					✓
4	Ketepatan pemilihan huruf					✓
5	Ketepatan ukuran huruf					✓
6	Kebergunaan untuk Prodi					✓
7	Pengaturan jarak					✓
8	Kejelasan gambar				✓	
9	Ukuran gambar				✓	
10	Kebergunaan untuk mahasiswa					✓
Jumlah					8	40
Jumlah rata-rata					4.8	96%
Jumlah Persentase						96%

Tambahan dari ahli media :

→ Tar Buat menu Daftar laporan pada menu mahasiswa.

Penguji


(HENDRI AHMADIAN)

جامعة الرانيري
AR - RANIRY

Nama Ahli media : Rahmat Muspikar
 Tanggal : 23 Juni 2021

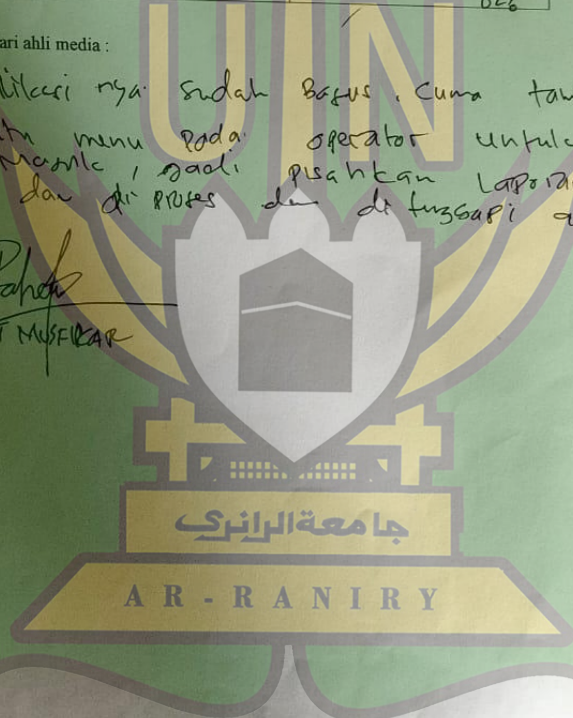
NO	Aspek yang Dinilai	Alternatif Pilihan				
		1 SS	2 TS	3 R	4 S	5 SS
1	Pemilihan tampilan				X	
2	Layout/tata letak tombol					X
3	Kemenarikan desain				X	
4	Ketepatan pemilihan huruf					X
5	Ketepatan ukuran huruf			X		
6	Kebergunaan untuk Prodi				X	
7	Pengaturan jarak				X	
8	Kejelasan gambar			X		
9	Ukuran gambar			X		
10	Kebergunaan untuk mahasiswa					X
Jumlah				0	12	20
Jumlah rata-rata					4.1	82%
Jumlah Persentase						82%

Tambahan dari ahli media :

Applikasi nya sudah bagus. Cuma tambah
 satu menu pada operator untuk laporan
 masalah, jadi pisahkan laporan masalah
 dan di proses dan di fungsioni atau tidak.

Penguji

Rahmat Muspikar
 RAHMAT MUSPIKAR




Lampiran 3 Kuisisioner *Integration Test*

1. Integration Testing (Mahasiswa)

No.	Test Case	Keluaran	Hasil
1.	Masukkan alamat berikut http://localhost/ptireport/ . Pada <i>browser</i> anda.	Tampil halaman <i>Login</i>	✓
2.	Masukkan <i>Username</i> dan <i>Password</i>	<ul style="list-style-type: none"> - Jika login benar maka tampil Beranda mahasiswa - Jika login salah tampil pesan kesalahan 	✓
3.	Tulis pelaporan	<i>Text box</i> tulis laporan akan aktif	✓
4.	Tekan tombol pilih gambar	Tampil <i>File explorer</i> untuk mencari gambar yang diinginkan	✓
5.	Tekan tombol kirim	Tampil pesan mengatakan pengaduan akan segera di tanggapi	✓
6.	Tekan tombol "More"	Tampil detail laporan yang telah di tanggapi	✓
7.	Tekan tombol "Hapus"	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data laporan - Jika Cancel, sistem akan membatalkan penghapusan data 	✓
8.	Pilih menu <i>Logout</i>	Keluar dari sistem	✓

Penguji (Mahasiswa)


(Nushadida)

جامعة الرانيري

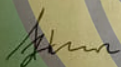
AR - RANIRY

2. Integration Testing (Admin)

No.	Test Case	Keluaran	Hasil
1.	Masukkan alamat berikut http://localhost/ptireport/ . Pada browser anda.	Tampil halaman <i>Login</i>	✓
2.	Masukkan <i>Username</i> dan <i>Password</i>	<ul style="list-style-type: none"> - Jika login benar maka tampil Beranda Admin - Jika login salah tampil pesan kesalahan 	✓
3.	Pilih menu Registrasi	Tampil daftar mahasiswa yang sudah memiliki akun pelaporan	✓
	3.1 Pilih menu edit pada salah satu akun mahasiswa	Tampil form untuk mengedit data akun Mahasiswa	✓
	3.2 Pilih menu Hapus pada salah satu akun mahasiswa	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data mahasiswa - Jika Cancel, sistem akan membatalkan penghapusan data mahasiswa 	✓
	3.3 pilih menu tambah	Tampil form untuk menambahkan akun mahasiswa yang baru	✓
4.	Pilih menu Pengaduan	Tampil daftar aduan yang sudah selesai maupun yang masih tahap proses	✓
	4.1 Pilih menu "More" pada salah satu data pengaduan	Tampil detail Laporan Pengaduan	✓
	4.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data laporan - Jika Cancel, sistem akan membatalkan penghapusan data 	✓
5.	Pilih menu Respon	Tampil data tanggapan operator prodi terhadap pengaduan yang masuk	✓
	5.1 Pilih menu "More" pada salah satu data pengaduan	Tampil detail Respon	✓
	5.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data Respon - Jika Cancel, sistem akan membatalkan penghapusan data 	✓

6.	Pilih Menu User	Tampil seluruh daftar petugas yang ada di sistem termasuk operator prodi dan admin	✓
	6.1 Pilih menu edit pada salah satu akun	Tampil form untuk mengedit data akun	✓
	6.2 Pilih menu Hapus pada salah satu akun	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data - Jika Cancel, sistem akan membatalkan penghapusan data akun 	✓
	6.3 pilih menu tambah	Tampil form untuk menambahkan akun petugas yang baru	✓
7.	Pilih menu Laporan	Tampil seluruh data pengaduan yang telah di tanggapi	✓
	7.1 Pilih menu "More" pada salah satu data pengaduan	Tampil detail Laporan	✓
	7.2 Klik tombol Printer	<ul style="list-style-type: none"> - Tampil <i>preview</i> cetak pada sistem - menyimpan daftar pengaduan 	✓
8.	Pilih menu Logout	Keluar dari sistem	✓

Penguji (Admin)


(CUT AZHAR F.)

جامعة الرانيري
A R - R A N I R Y

3. Integration sistem (Operator Prodi)

No.	Test Case	Keluaran	Hasil
1.	Masukkan alamat berikut http://localhost/ptireport/ . Pada <i>browser</i> anda.	Tampil halaman <i>Login</i>	✓
2.	Masukkan <i>Username</i> dan <i>Password</i>	<ul style="list-style-type: none"> - Jika login benar maka tampil Beranda Operator prodi - Jika login salah tampil pesan kesalahan 	✓
3.	Pilih menu Pengaduan	Tampil daftar aduan yang sudah selesai maupun yang masih tahap proses	✓
	3.1 Pilih menu " <i>More</i> " pada salah satu data pengaduan	Tampil detail Laporan Pengaduan	✓
	3.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data pengaduan - Jika Cancel, sistem akan membatalkan penghapusan data 	✓
	3.3 tambahkan tanggapan	Text Box tanggapan akan aktif	✓
	3.4 kirim tanggapan	Muncul pesan bahwa tanggapan telah di kirim	✓
4.	Pilih menu Respon	Tampil data tanggapan operator prodi terhadap pengaduan yang masuk	✓
	4.1 Pilih menu " <i>More</i> " pada salah satu data pengaduan	Tampil detail Respon	✓
	4.2 Pilih menu Hapus pada salah satu data pengaduan	<ul style="list-style-type: none"> - Jika klik OK maka sistem akan menghapus data Respon - Jika Cancel, sistem akan membatalkan penghapusan data 	✓
5.	Pilih menu Logout	Keluar dari sistem	✓

Penguji (Petugas Prodi)

Nurul Fajri

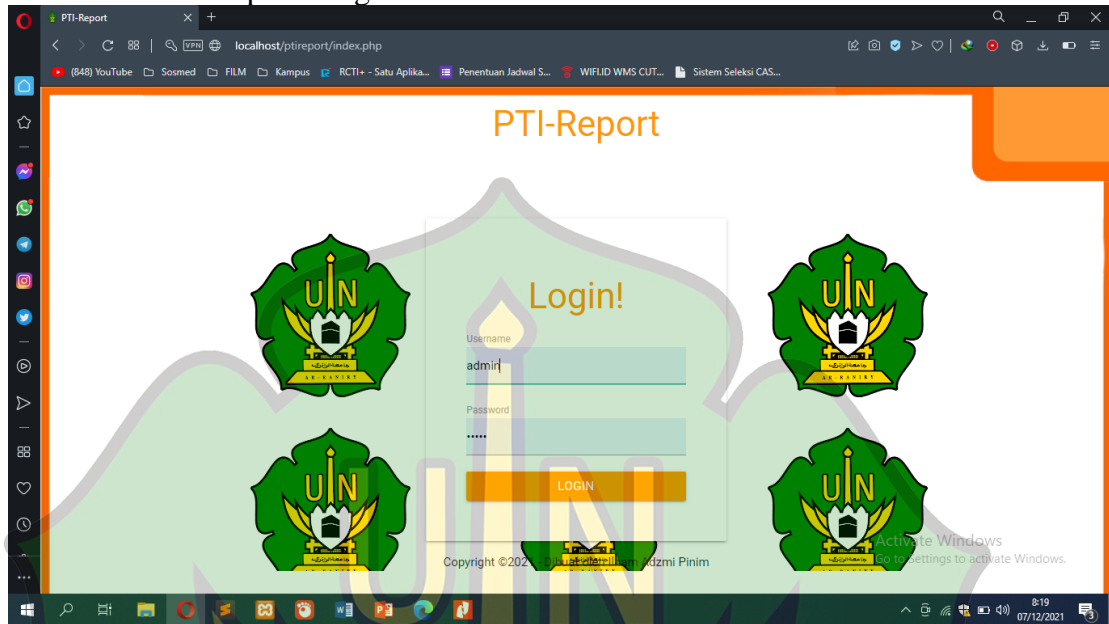
(Nurul Fajri)

جامعة الرانيري

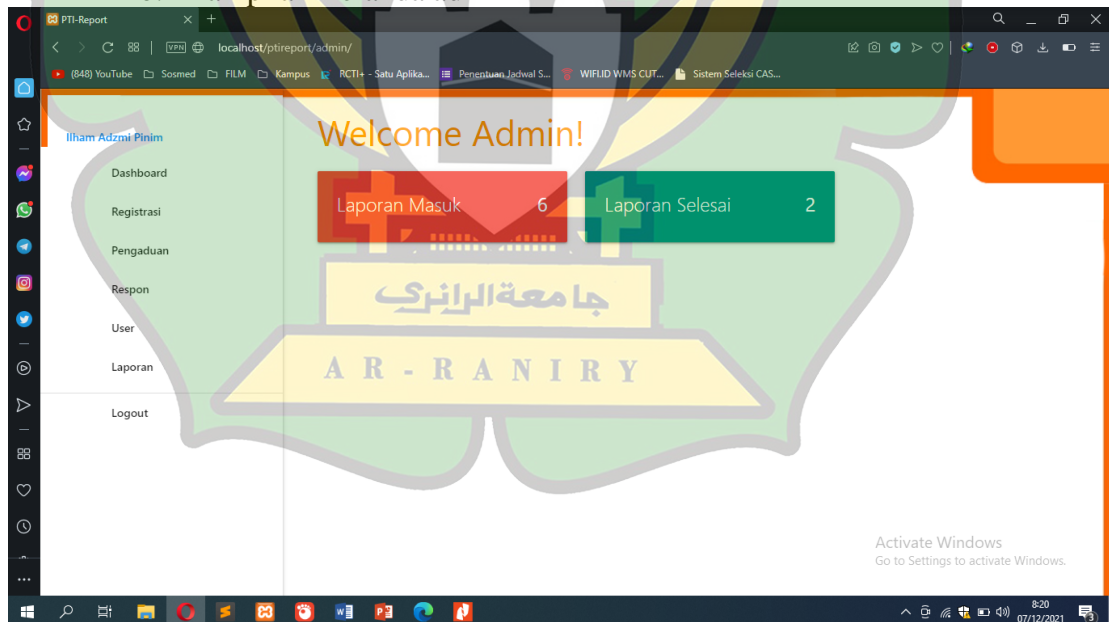
AR - RANIRY

Lampiran 4 Tangkapan Layar tampilan website yang telah di buat.

a. Tampilan Login



b. Tampilan Beranda admin



c. Tampilan tambah mahasiswa

The screenshot shows the 'Mahasiswa' page in the PTI-Report admin interface. The page title is 'Mahasiswa' and it features a search bar and a 'Show' dropdown set to 10. Below this is a table with 3 entries. The table has columns for 'No', 'NIM', 'Nama', 'Username', 'Telp', and 'Opsi'. The 'Opsi' column contains 'EDIT' and 'HAPUS' buttons for each entry. A sidebar on the left lists navigation options: Dashboard, Registrasi, Pengaduan, Respon, User, Laporan, and Logout. The bottom of the page shows 'Showing 1 to 3 of 3 entries' and navigation buttons for 'Previous', '1', and 'Next'. An 'Activate Windows' watermark is visible in the bottom right corner.

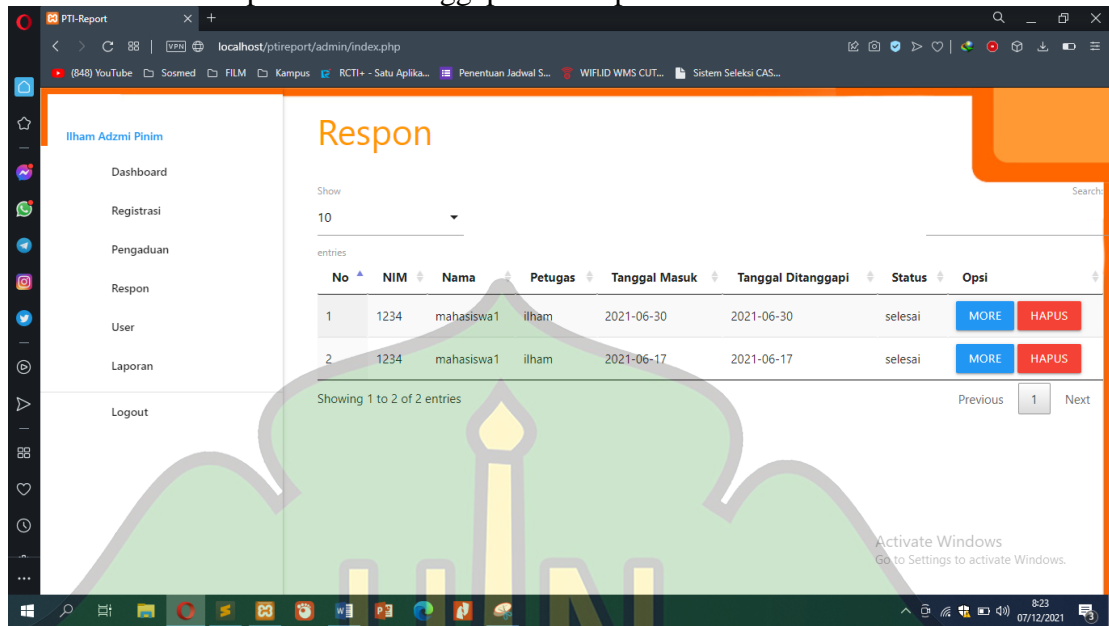
No	NIM	Nama	Username	Telp	Opsi
1	1234	mahasiswa1	m1	081921	EDIT HAPUS
2	12345	amin	amin	0987	EDIT HAPUS
3	1602111111	Adzmi	mahasiswa	085364287180	EDIT HAPUS

d. Tampilan daftar pengaduan yang masuk

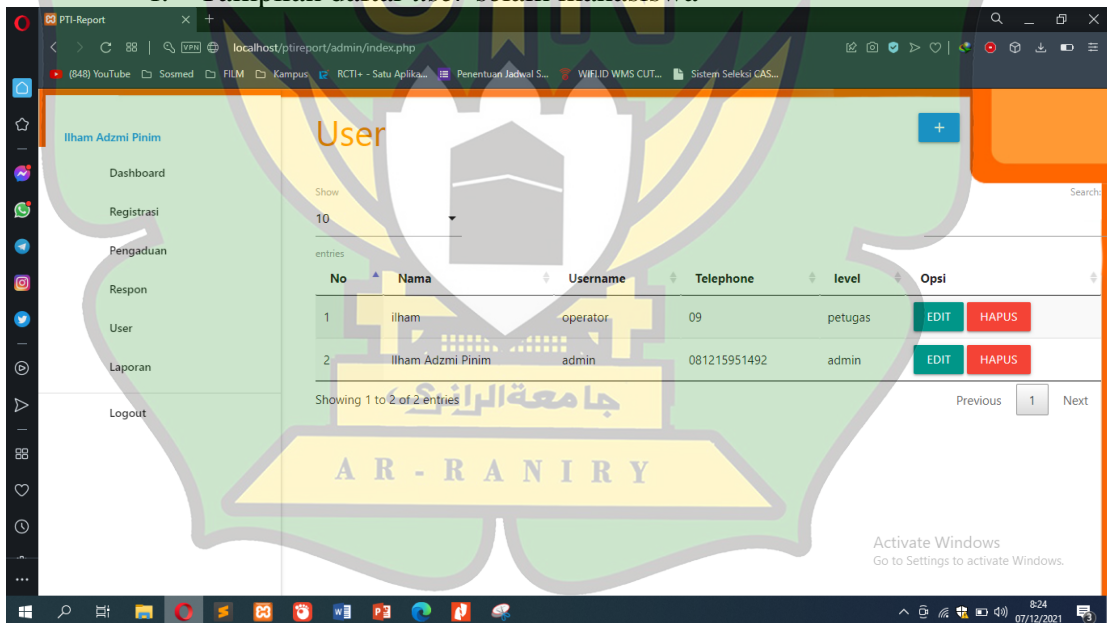
The screenshot shows the 'Pengaduan' page in the PTI-Report admin interface. The page title is 'Pengaduan' and it features a search bar and a 'Show' dropdown set to 10. Below this is a table with 6 entries. The table has columns for 'No', 'NIM', 'Nama', 'Tanggal Masuk', 'Status', and 'Opsi'. The 'Opsi' column contains 'MORE' and 'HAPUS' buttons for each entry. A sidebar on the left lists navigation options: Dashboard, Registrasi, Pengaduan, Respon, User, Laporan, and Logout. The bottom of the page shows 'Showing 1 to 6 of 6 entries' and navigation buttons for 'Previous', '1', and 'Next'. An 'Activate Windows' watermark is visible in the bottom right corner.

No	NIM	Nama	Tanggal Masuk	Status	Opsi
1	12345	amin	2021-06-30	proses	MORE HAPUS
2	1234	mahasiswa1	2021-06-30	selesai	MORE HAPUS
3	1234	mahasiswa1	2021-06-23	proses	MORE HAPUS
4	1234	mahasiswa1	2021-06-17	selesai	MORE HAPUS
5	1234	mahasiswa1	2021-06-17	proses	MORE HAPUS
6	1234	mahasiswa1	2021-06-16	proses	MORE HAPUS

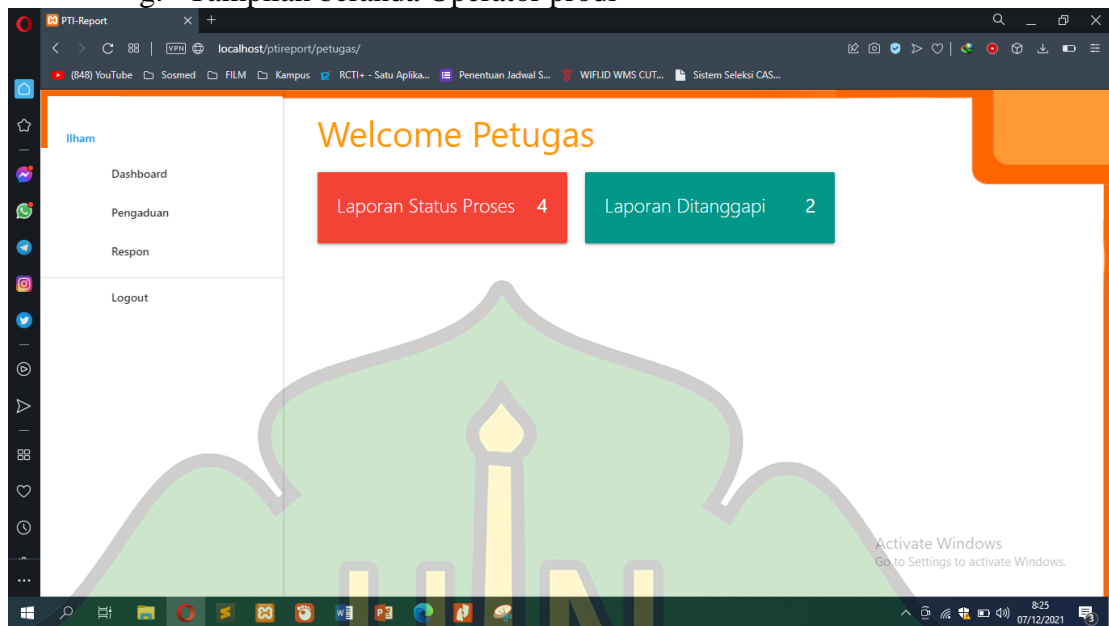
e. Tampilan daftar tanggapan dari operator



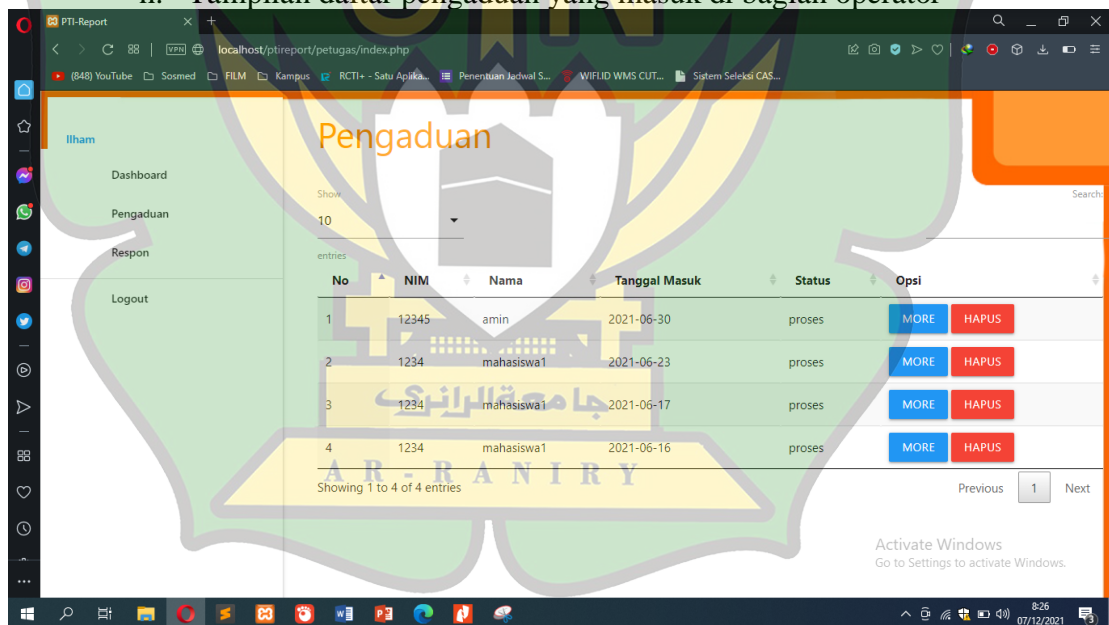
f. Tampilan daftar user selain mahasiswa



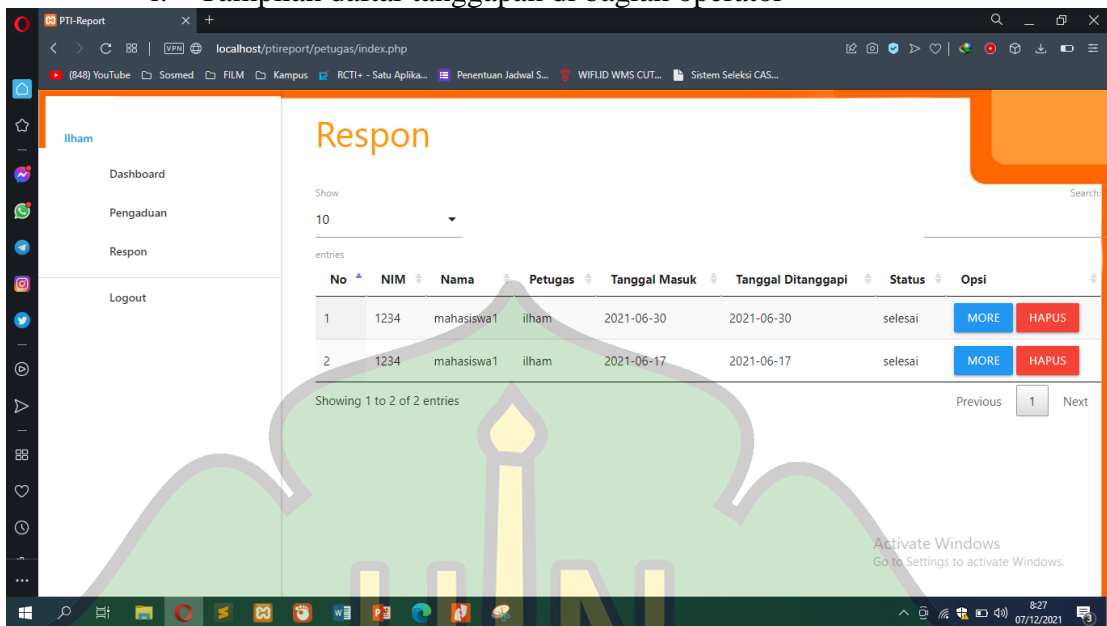
g. Tampilan beranda Operator prodi



h. Tampilan daftar pengaduan yang masuk di bagian operator



i. Tampilan daftar tanggapan di bagian operator



j. Tampilan beranda mahasiswa

