

**PENGEMBANGAN APLIKASI PENGELOLAAN  
LAYANAN *HELPDESK* BERBASIS WEB  
DI UNIVERSITAS WIDYATAMA**

**SKRIPSI**

**Diajukan Sebagai Salah Satu Syarat untuk memperoleh kelulusan  
Jenjang Strata Satu (S1) pada  
Program Studi Teknik Informatika**

**Disusun Oleh:  
GIFARI ABDALLAH  
361942007**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI**

**BANDUNG**

**2023**

## **LEMBAR PENGESAHAN**

### **PENGEMBANGAN APLIKASI PENGELOLAAN LAYANAN *HELPDESK* BERBASIS WEB DI UNIVERSITAS WIDYATAMA**

**Oleh:**

**GIFARI ABDALLAH**

**361942007**

Skripsi ini telah diterima dan disahkan  
untuk memenuhi persyaratan mencapai gelar.

**SARJANA TEKNIK INFORMATIKA**

Pada  
**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

Bandung, 25 Juni 2023

Disahkan Oleh,

Ketua Program Studi

Dosen Pembimbing

Asto Purwanto., S.T., M.Kom.

Chalifa Chazar,S.T.,M.T

NIDN : 0404078903

NIDN : 0421098704

## **LEMBAR PERSETUJUAN REVISI**

### **PENGEMBANGAN APLIKASI PENGELOLAAN LAYANAN *HELPDESK* BERBASIS WEB DI UNIVERSITAS WIDYATAMA**

**Oleh:**

**GIFARI ABDALLAH**

**361942007**

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, 21 Juni 2022

Menyetujui

No	Nama Dosen	Keterangan	Tanda tangan
1.	Chalifa Chazar,S.T.,M.T	Dosen Pembimbing	
2.	Yudhi W. Arthana R., S.T., M.kom.	Penguji 1	
3.	Moch. Ali Ramdhani, S.T., M.Kom.	Penguji 2	

Mengetahui  
Ketua Program Studi Teknik Informatika

Asto Purwanto., S.T., M.Kom.

NIDN : 0404078903

## **SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa:

- 1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- 2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- 3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, Juni 2023

Yang membuat pernyataan

Gifari Abdallah

361942007

## **ABSTRAK**

Di era Teknologi Informasi (TI) hampir semua aktivitas di dunia berhubungan TI termasuk dalam pendidikan, dalam pendidikan khususnya di Universitas sudah menggunakan TI meliputi komputer, printer, layanan internet, absensi, Sistem Informasi (SI), bahkan dalam pembelajaran. Universitas Widyatama (UTama) memiliki unit TI yang disebut Pusat Teknologi Informasi (PTI) yang terdiri dari 8 orang staf, serta beberapa tenaga laboran yang dikepalai oleh kepala biro PTI dan kepala pusat PTI. Pekerjaan dalam unit ini meliputi pemeliharaan perangkat keras dan perangkat lunak untuk perkantoran, laboratorium komputer, pemeliharaan jaringan ruang, kelas, dosen dan jaringan nirkabel mahasiswa. Dalam pelayanan tersebut terkadang bisa saja terjadi gangguan dan kerusakan, dari gangguan tersebut menyebabkan terganggunya proses perkuliahan terutama dalam perkuliahan daring. Dalam penangannya banyak alat yang digunakan untuk menangani hal tersebut tetapi banyak kendala - kendala yang dihadapi seperti tidak terorganisir dengan baik dikarenakan banyaknya jalur pelaporan dan perlunya pengecekan kembali gangguan/komplain, kesulitan pengenalan identitas pelapor, penyampaiannya kepada staf teknisi, serta pencatatan dan penghitungan masih dituliskan dan dihitung secara manual selain itu mahasiswa dan dosen pun mengalami kesulitan dalam mengecek status gangguan/komplain mereka, yang menyebabkan ketidakjelasan apakah gangguan sudah ditangani atau belum. Oleh karena itu diperlukan suatu alat atau aplikasi yang dapat menangani pelayanan yang di dalamnya meliputi pelaporan dan penanganan, serta pengelolaan yang meliputi pencatatan, penghitungan, dan penyampaian gangguan/komplain secara cepat dan akurat. Maka penelitian ini bertujuan untuk membangun Aplikasi Pengelolaan Layanan *Helpdesk* yang akan dikelola oleh *admin*, aplikasi yang dibuat berbasis *website* menggunakan metode *waterfall*, Aplikasi Pengelolaan Layanan *Helpdesk* ini dibuat dengan menggunakan bahasa pemrograman PHP sebagai *interface* sistem. Untuk pengelolaan *database* dalam hal ini menggunakan MYSQL. Aplikasi dibuat berbasis *website* agar memudahkan pengguna baik admin, mahasiswa dan dosen dalam mengaksesnya dimanapun dan kapanpun dengan hanya menggunakan perangkat apapun yang memiliki *web browser* dan jaringan internet. Dengan adanya aplikasi *web* ini diharapkan dapat memudahkan tim *helpdesk* Widyatama dalam melakukan pelayanan dan pengelolaan data komplain baik pencatatan atau penghitungan, serta penyampaiannya.

Kata Kunci : Komplain, Aplikasi, Pengelolaan, *Helpdesk*, Internet, UML, PHP

## **ABSTRACT**

*In the Information Technology (IT) era, almost all activities in the world related to IT are included in education, in education, especially at universities, already using IT, including computers, printers, internet services, attendance, Information Systems (IS), even in learning. Widyatama University (UTama) has an IT unit called the Information Technology Center (PTI) which consists of 8 staff, as well as several laboratory staff headed by the head of the PTI bureau and the head of the PTI center. Work in this unit includes maintenance of hardware and software for offices, computer laboratories, maintenance of room networks, classrooms, lecturers and student wireless networks. In this service, sometimes interruptions and damage can occur, from these disturbances it causes disruption to the lecture process, especially in online lectures. In handling, many tools are used to handle this, but many obstacles are encountered, such as not being organized properly due to the large number of reporting lines and the need to re-check disturbances/complaints, difficulties in identifying the identity of the complainant, submitting it to technical staff, and recording and calculating are still being written down. and calculated manually. Besides that, students and lecturers also experience difficulties in checking the status of their disturbances/complaints, which causes it to be unclear whether the disturbances have been handled or not. Therefore we need a tool or application that can handle services which include reporting and handling, as well as management which includes recording, counting, and submitting disturbances/complaints quickly and accurately. So this research aims to build a Helpdesk Service Management Application that will be managed by the admin, a website-based application using the waterfall method, this Helpdesk Service Management Application is made using the PHP programming language as a system interface. For database management in this case using MYSQL. The application is made website-based to make it easier for users, both admins, students and lecturers, to access it anywhere and anytime using only any device that has a web browser and internet network. With this web application, it is hoped that it will make it easier for the Widyatama helpdesk team to service and manage complaint data, both recording and calculating, as well as submitting it.*

*Keywords:* Complaints, Applications, Management, Helpdesk, Internet, UML, PHP

## **KATA PENGANTAR**

Puji syukur kehadirat Allah subhanahu wa taala atas limpahan rahmat dan kasihnya sehingga penulis dapat menyelesaikan skripsi ini dengan lancar. Pengembangan Aplikasi Pengelolaan Layanan *Helpdesk* Berbasis Web di Universitas Widyatama ini ditulis untuk memenuhi sebagian syarat memenuhi tugas.

Penyusunan skripsi ini tidak lepas dari hambatan dan rintangan serta kesulitan-kesulitan. Penulis mengakui bahwa dalam mempersiapkan, melaksanakan penelitian, dan menyelesaikan penulisan skripsi ini tidak akan selesai tanpa bantuan dari berbagai pihak, oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada berbagai pihak

Akhirnya penulis berharap semoga skripsi ini bermanfaat. Penulis menyambut baik segala kritik dan saran yang bersifat membangun untuk kesempurnaan skripsi ini dengan tangan terbuka

Bandung, Juni 2023

Penulis

Gifari Abdallah

## DAFTAR ISI

<b>LEMBAR PENGESAHAN.....</b>	<b>i</b>
<b>LEMBAR PERSETUJUAN REVISI .....</b>	<b>ii</b>
<b>SURAT PERNYATAAN .....</b>	<b>iii</b>
<b>ABSTRAK.....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI .....</b>	<b>vii</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Identifikasi Masalah .....	3
1.3    Tujuan Penulisan.....	4
1.4    Batasan Masalah .....	4
1.5    Metode Penelitian.....	5
1.6    Sistematika Penulisan.....	7
<b>BAB II LANDASAN TEORI.....</b>	<b>8</b>
2.1    Definisi <i>Website</i> .....	8
2.2    Definisi Sistem.....	8
2.3 <i>Unified Modeling Language (UML)</i> .....	9
2.4 <i>Entity Relationship Diagram</i> .....	13
2.5 <i>Flowchart</i> .....	14
2.6    Definisi Pengelolaan .....	15
2.7    Definisi Layanan .....	16
2.8    Definisi <i>Helpdesk</i> .....	16
2.9    Apache .....	17
2.10    Bahasa Pemrograman PHP .....	17
2.11 <i>Framework Laravel</i> .....	17
2.12    Basis Data .....	18
2.13    Sistem Manajemen Basis Data.....	19
2.14 <i>Black Box Testing</i> .....	23

<b>BAB III ANALISIS DAN PERANCANGAN SISTEM.....</b>	<b>25</b>
3.1    Pengumpulan Data .....	25
3.1.1    Observasi .....	25
3.1.2    Wawancara .....	27
3.1.3    Studi Pustaka.....	27
3.2    Analisis Sistem.....	29
3.2.1    Analisis Sistem yang Sedang Berjalan.....	29
3.3    Perancangan Sistem yang Diusulkan .....	29
3.3.1    Hasil analisis .....	31
3.4    Perancangan Sistem.....	31
3.4.1 <i>Unified Modelling Language (UML)</i> .....	32
3.4.1.1    Aktor.....	32
3.4.1.2 <i>Use Case Diagram</i> .....	33
3.4.1.3    Skenario <i>Use Case</i> .....	34
3.4.1.3.1    Skenario <i>Use Case Admin</i> .....	34
3.4.1.3.2    Skenario <i>Use Case Teknisi</i> .....	42
3.4.1.3.3    Skenario <i>Use Case Pengguna</i> .....	44
3.4.1.3.4    Skenario <i>Use Case Semua Pengguna</i> .....	46
3.4.1.4 <i>Activity Diagram</i> .....	47
3.4.1.4.1 <i>Activity Diagram Admin</i> .....	47
3.4.1.4.2 <i>Activity Diagram Teknisi</i> .....	55
3.4.1.4.3 <i>Activity Diagram Pengguna</i> .....	57
3.4.1.4.4 <i>Activity Diagram Semua Pengguna</i> .....	59
3.4.1.5 <i>Sequence Diagram</i> .....	60
3.4.1.5.1 <i>Sequence Diagram Admin</i> .....	60
3.4.1.5.2 <i>Sequence Diagram Teknisi</i> .....	68
3.4.1.5.3 <i>Sequence Diagram Pengguna</i> .....	70
3.4.1.5.4 <i>Sequence Diagram Semua Pengguna</i> .....	72
3.4.1.6 <i>Class Diagram</i> .....	73
3.4.1.7 <i>Entity Relationship Diagram</i> .....	74
3.4.2    Perancangan Antarmuka.....	75

<b>BAB IV IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>88</b>
4.1    Implementasi Dan Pengujian .....	88
4.1.1 <i>Server</i> .....	88
4.1.2 <i>Client</i> .....	89
4.2    Struktur Tabel .....	89
4.3    Relasi Tabel .....	91
4.4    Implementasi Antarmuka .....	92
4.5    Pengujian .....	105
<b>BAB V PENUTUP .....</b>	<b>122</b>
5.1    Simpulan.....	122
5.2    Saran.....	123
<b>DAFTAR PUSTAKA .....</b>	<b>124</b>
<b>LAMPIRAN-LAMPIRAN.....</b>	<b>127</b>

## DAFTAR TABEL

Tabel 2.1 Simbol <i>Use Case Diagram</i> (Suryana, 2018) .....	10
Tabel 2.2 Simbol <i>Sequence Diagram</i> (Suryana, 2018).....	11
Tabel 2.3 <i>Class Diagram</i> (Suryana, 2018).....	12
Tabel 2.4 Simbol <i>Activity Diagram</i> (Suryana, 2018) .....	13
Tabel 2.5 Simbol <i>Entity Relationship Diagram</i> (Triaulia, 2021).....	14
Tabel 3.1 Wawancara.....	27
Tabel 3.2 Studi Pustaka.....	28
Tabel 3.3 Aktor.....	32
Tabel 3.4 Menerima Pengajuan Komplain.....	34
Tabel 3.5 Mengubah Komplain .....	34
Tabel 3.6 Menghapus Komplain .....	35
Tabel 3.7 Merespon Pesan Komplain .....	35
Tabel 3.8 Mengubah Status.....	36
Tabel 3.9 Mengirim Komplain Ke Teknisi .....	36
Tabel 3.10 Membuat Kategori.....	37
Tabel 3.11 Mengubah Kategori .....	37
Tabel 3.12 Menghapus Kategori .....	38
Tabel 3.13 Membuat Tipe Kendala .....	38
Tabel 3.14 Mengubah Tipe Kendala.....	39
Tabel 3.15 Menghapus Tipe Kendala .....	39
Tabel 3.16 Membuat Pengguna .....	40
Tabel 3.17 Mengubah Pengguna .....	40
Tabel 3.18 Menghapus Pengguna.....	41
Tabel 3.19 Laporan .....	41
Tabel 3.20 Mencetak Laporan.....	42
Tabel 3.21 Menerima Kiriman Komplain .....	42
Tabel 3.22 Melihat Pesan .....	43
Tabel 3.23 Memproses Komplain.....	43
Tabel 3.24 Menyelesaikan Komplain .....	44
Tabel 3.25 Melakukan Pendaftaran .....	44
Tabel 3.26 Membuat Pengajuan Komplain.....	45
Tabel 3.27 Membalas Pesan Komplain.....	45
Tabel 3.28 <i>Login</i> .....	46
Tabel 3.29 Mengubah <i>Password</i> .....	46
Tabel 4.1 Spesifikasi Perangkat Keras Minimum .....	88
Tabel 4.2 Spesifikasi Perangkat Lunak Minimum .....	89
Tabel 4.3 Spesifikasi Minimum Client .....	89
Tabel 4.4 Pengujian <i>Blackbox</i> Menerima Pengajuan Komplain .....	105

Tabel 4.5 Pengujian <i>Blackbox</i> Mengubah Komplain .....	105
Tabel 4.6 Pengujian <i>Blackbox</i> Menghapus Komplain.....	106
Tabel 4.7 Pengujian <i>Blackbox</i> Merespon Pesan.....	106
Tabel 4.8 Pengujian <i>Blackbox</i> Mengubah Status .....	107
Tabel 4.9 Pengujian <i>Blackbox</i> Mengirim Komplain Ke Teknisi .....	107
Tabel 4.10 Pengujian <i>Blackbox</i> Membuat Kategori .....	107
Tabel 4.11 Pengujian <i>Blackbox</i> Mengubah Kategori .....	108
Tabel 4.12 Pengujian <i>Blackbox</i> Menghapus Kategori.....	109
Tabel 4.13 Pengujian <i>Blackbox</i> Membuat Tipe Kendala.....	109
Tabel 4.14 Pengujian <i>Blackbox</i> Mengubah Tipe Kendala .....	110
Tabel 4.15 Pengujian <i>Blackbox</i> Menghapus Tipe Kendala.....	111
Tabel 4.16 Pengujian <i>Blackbox</i> Membuat Pengguna .....	111
Tabel 4.17 Pengujian <i>Blackbox</i> Mengubah Pengguna.....	112
Tabel 4.18 Pengujian <i>Blackbox</i> Menghapus Pengguna .....	113
Tabel 4.19 Pengujian <i>Blackbox</i> Melihat Laporan .....	113
Tabel 4.20 Pengujian <i>Blackbox</i> Mencetak Laporan .....	114
Tabel 4.21 Pengujian <i>Blackbox</i> Komplain Masuk .....	114
Tabel 4.22 Pengujian <i>Blackbox</i> Melihat Pesan .....	115
Tabel 4.23 Pengujian <i>Blackbox</i> Memproses Komplain.....	115
Tabel 4.24 Pengujian <i>Blackbox</i> Menyelesaikan Komplain.....	115
Tabel 4.25 Pengujian <i>Blackbox</i> Melakukan Pendaftaran.....	116
Tabel 4.27 Pengujian <i>Blackbox</i> Membuat Pengajuan Komplain .....	117
Tabel 4.28 Pengujian <i>Blackbox</i> Membalas Pesan .....	118
Tabel 4.28 Pengujian <i>Blackbox Login</i> .....	119
Tabel 4.29 Pengujian <i>Blackbox</i> Mengubah Password.....	120

## DAFTAR GAMBAR

Gambar 2.1 Simbol <i>Flowchart</i> (Astuti,2016) .....	15
Gambar 3.1 <i>Flowmap</i> Berdasarkan Prosedur.....	26
Gambar 3.2 <i>Flowmap</i> Prosedur Sistem Yang Diusulkan .....	30
Gambar 3.3 <i>Use Case Diagram</i> .....	33
Gambar 3.4 <i>Activity Diagram</i> Menerima Pengajuan Komplain .....	47
Gambar 3.5 <i>Activity Diagram</i> Mengubah Komplain.....	47
Gambar 3.6 <i>Activity Diagram</i> Menghapus Komplain .....	48
Gambar 3.7 <i>Activity Diagram</i> Merespon Pesan Komplain .....	48
Gambar 3.8 <i>Activity Diagram</i> Mengubah Status Komplain .....	49
Gambar 3.9 <i>Activity Diagram</i> Mengirim Komplain Ke Teknisi.....	49
Gambar 3.10 <i>Activity Diagram</i> Membuat Kategori .....	50
Gambar 3.11 <i>Activity Diagram</i> Mengubah Kategori.....	50
Gambar 3.12 <i>Activity Diagram</i> Menghapus Kategori .....	51
Gambar 3.13 <i>Activity Diagram</i> Membuat Tipe Kendala .....	51
Gambar 3.14 <i>Activity Diagram</i> Mengubah Tipe Kendala .....	52
Gambar 3.15 <i>Activity Diagram</i> Menghapus Tipe Kendala .....	52
Gambar 3.16 <i>Activity Diagram</i> Membuat Pengguna.....	53
Gambar 3.17 <i>Activity Diagram</i> Mengubah Pengguna .....	53
Gambar 3.18 <i>Activity Diagram</i> Menghapus Pengguna.....	54
Gambar 3.19 <i>Activity Diagram</i> Melihat Laporan.....	54
Gambar 3.20 <i>Activity Diagram</i> Mencetak Laporan.....	55
Gambar 3.21 <i>Activity Diagram</i> Menerima Kiriman Komplain.....	55
Gambar 3.22 <i>Activity Diagram</i> Melihat Pesan.....	56
Gambar 3.23 <i>Activity Diagram</i> Memproses Komplain .....	56
Gambar 3.24 <i>Activity Diagram</i> Menyelesaikan Komplain .....	57
Gambar 3.25 <i>Activity Diagram</i> Melakukan Pendaftaran .....	57
Gambar 3.26 <i>Activity Diagram</i> Membuat Pengajuan Komplain.....	58
Gambar 3.27 <i>Activity Diagram</i> Membalas Pesan Komplain .....	58
Gambar 3.28 <i>Activity Diagram Login</i> .....	59
Gambar 3.29 <i>Activity Diagram</i> Mengubah Password .....	59
Gambar 3.30 <i>Sequence Diagram</i> Menerima Pengajuan Komplain .....	60
Gambar 3.31 <i>Sequence Diagram</i> Mengubah Komplain .....	60
Gambar 3.32 <i>Sequence Diagram</i> Menghapus Komplain.....	61
Gambar 3.33 <i>Sequence Diagram</i> Merespon Pesan Komplain .....	61
Gambar 3.34 <i>Sequence Diagram</i> Mengubah Status Komplain .....	62
Gambar 3.35 <i>Sequence Diagram</i> Mengirim Komplain Ke Teknisi .....	62
Gambar 3.36 <i>Sequence Diagram</i> Membuat Kategori.....	63
Gambar 3.37 <i>Sequence Diagram</i> Mengubah Kategori .....	63

Gambar 3.38 <i>Sequence Diagram</i> Menghapus Kategori .....	64
Gambar 3.39 <i>Sequence Diagram</i> Tipe Kendala .....	64
Gambar 3.40 <i>Sequence Diagram</i> Mengubah Tipe Kendala.....	65
Gambar 3.41 <i>Sequence Diagram</i> Menghapus Tipe Kendala .....	65
Gambar 3.42 <i>Sequence Diagram</i> Kelola Pengguna .....	66
Gambar 3.43 <i>Sequence Diagram</i> Mengubah Pengguna .....	66
Gambar 3.44 <i>Sequence Diagram</i> Menghapus Pengguna.....	67
Gambar 3.45 <i>Sequence Diagram</i> Melihat Laporan .....	67
Gambar 3.46 <i>Sequence Diagram</i> Mencetak Laporan.....	68
Gambar 3.47 <i>Sequence Diagram</i> Menerima Kiriman Komplain .....	68
Gambar 3.48 <i>Sequence Diagram</i> Melihat Pesan .....	69
Gambar 3.49 <i>Sequence Diagram</i> Memproses Komplain.....	69
Gambar 3.50 <i>Sequence Diagram</i> Menyelesaikan Komplain .....	70
Gambar 3.51 <i>Sequence Diagram</i> Melakukan Pendaftaran .....	70
Gambar 3.52 <i>Sequence Diagram</i> Membuat Pengajuan Komplain .....	71
Gambar 3.53 <i>Sequence Diagram</i> Membalas Pesan Komplain.....	71
Gambar 3.54 <i>Sequence Diagram</i> Login .....	72
Gambar 3.55 <i>Sequence Diagram</i> Mengubah Password .....	72
Gambar 3.56 <i>Class Diagram</i> .....	73
Gambar 3.57 <i>Entity Relationship Diagram</i> .....	74
Gambar 3.58 Rancangan Menerima Pengajuan Komplain .....	75
Gambar 3.58 Rancangan Mengubah Komplain .....	75
Gambar 3.60 Rancangan Menghapus Komplain .....	76
Gambar 3.61 Rancangan Merespon Pesan .....	76
Gambar 3.62 Rancangan Mengubah Status .....	77
Gambar 3.63 Rancangan Mengirim Komplain Ke Teknisi .....	77
Gambar 3.64 Rancangan Membuat Kategori .....	78
Gambar 3.65 Rancangan Mengubah Kategori .....	78
Gambar 3.66 Rancangan Menghapus Kategori.....	79
Gambar 3.67 Rancangan Membuat Tipe Kendala.....	79
Gambar 3.68 Rancangan Mengubah Tipe Kendala .....	80
Gambar 3.69 Rancangan Menghapus Tipe Kendala .....	80
Gambar 3.70 Rancangan Membuat Pengguna .....	81
Gambar 3.71 Rancangan Mengubah Pengguna.....	81
Gambar 3.72 Rancangan Menghapus Pengguna .....	82
Gambar 3.73 Rancangan Melihat Laporan .....	82
Gambar 3.74 Rancangan Mencetak Laporan .....	83
Gambar 3.75 Rancangan Menerima Kiriman Komplain .....	83
Gambar 3.76 Rancangan Melihat Pesan .....	84
Gambar 3.77 Rancangan Memproses Komplain .....	84

Gambar 3.78 Rancangan Menyelesaikan Komplain.....	85
Gambar 3.79 Melakukan Pendaftaran.....	85
Gambar 3.80 Rancangan Membuat Pengajuan Komplain .....	86
Gambar 3.81 Rancangan Membalas Pesan .....	86
Gambar 3.82 Rancangan <i>Login</i> .....	87
Gambar 3.83 Rancangan Mengubah <i>Password</i> .....	87
Gambar 4.1 Tabel <i>Category</i> .....	89
Gambar 4.2 Tabel <i>Complain</i> .....	90
Gambar 4.3 Tabel <i>Message</i> .....	90
Gambar 4.4 Tabel <i>Problem</i> .....	90
Gambar 4.5 Tabel <i>Users</i> .....	91
Gambar 4.6 Relasi Tabel.....	91
Gambar 4.7 Tampilan Menerima Pengajuan Komplain .....	92
Gambar 4.8 Tampilan Mengubah Komplain.....	92
Gambar 4.9 Tampilan Menghapus Komplain .....	93
Gambar 4.10 Tampilan Merespon Pesan .....	93
Gambar 4.11 Tampilan Mengubah Status.....	94
Gambar 4.12 Tampilan Mengirim Komplain Ke Teknisi .....	94
Gambar 4.13 Tampilan Membuat Kategori .....	95
Gambar 4.14 Tampilan Mengubah Kategori.....	95
Gambar 4.15 Tampilan Menghapus Kategori .....	96
Gambar 4.16 Tampilan Membuat Tipe Kendala .....	96
Gambar 4.17 Tampilan Mengubah Tipe Kendala .....	97
Gambar 4.18 Tampilan Menghapus Tipe Kendala.....	97
Gambar 4.19 Tampilan Membuat Pengguna.....	98
Gambar 4.20 Tampilan Mengubah Pengguna .....	98
Gambar 4.21 Tampilan Menghapus Pengguna .....	99
Gambar 4.22 Tampilan Melihat Laporan.....	99
Gambar 4.23 Tampilan Mencetak Laporan.....	100
Gambar 4.24 Tampilan Komplain Masuk .....	100
Gambar 4.25 Tampilan Melihat Pesan.....	101
Gambar 4.26 Tampilan Memproses Komplain .....	101
Gambar 4.27 Tampilan Menyelesaikan Komplain .....	102
Gambar 4.28 Tampilan Melakukan Pendaftaran .....	102
Gambar 4.29 Tampilan Membuat Pengajuan Komplain .....	103
Gambar 4.30 Tampilan Membalas Pesan.....	103
Gambar 4.31 Tampilan <i>Login</i> .....	104
Gambar 4.32 Tampilan Mengubah <i>Password</i> .....	104

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Di era Teknologi Informasi (TI) hampir semua aktivitas di dunia berhubungan TI termasuk dalam pendidikan, dalam pendidikan khususnya di Universitas sudah menggunakan TI meliputi komputer, printer, layanan internet, absensi, Sistem Informasi (SI), bahkan dalam pembelajaran. Universitas Widyatama (UTama) memiliki unit TI yang disebut Pusat Teknologi Informasi (PTI) yang terdiri dari 8 orang staf, serta beberapa tenaga laboran, yang dikepalai oleh kepala biro PTI dan kepala pusat PTI. Unit PTI mempunyai banyak bagian, dan dalam bagian – bagian tersebut yang bekerja memelihara perangkat keras dan perangkat lunak laboratorium komputer, pemeliharaan jaringan ruang, kelas, dosen dan jaringan nirkabel mahasiswa, merupakan bagian teknisi, dan bagian dalam menerima gangguan/masalah yaitu bagian *helpdesk*. Dalam pelayanan PTI terkadang bisa saja terjadi gangguan dan kerusakan, dari gangguan tersebut menyebabkan terganggunya proses perkuliahan, mahasiswa dan dosen terkadang memerlukan bantuan jika hal tersebut terjadi. Dalam hal ini, yang menerima gangguan/komplain adalah admin *helpdesk* yang merupakan dari bagian *helpdesk*, admin *helpdesk* menerima gangguan/komplain yang kemudian informasi tersebut diberikan kepada staf teknisi untuk ditindaklanjuti.

Dalam masalah yang muncul selama ini dimana mahasiswa dan dosen yang melaporkan gangguan/komplain dapat secara langsung datang ke ruang PTI. Akan tetapi jika perkuliahan dilakukan secara daring di Universitas Widyatama masih menggunakan telepon, *email*, dan aplikasi *chatting* seperti *Whatsapp*, yang mana gangguan tersebut dicatat dan diberikan kepada staf teknisi yang bersangkutan. Dari proses tersebut banyak kendala - kendala yang dihadapi seperti pelayanan lambat dan tidak terorganisir dengan baik dikarenakan banyaknya jalur pelaporan dan perlunya pengecekan kembali gangguan/komplain agar tidak terjadi penggandaan pelaporan gangguan/komplain, ditambah menghubungi melalui *chat* dan telepon terkadang kurang akurat dalam mengenal identitas pelapor dan gangguan yang dihadapi, dimana hal tersebut dapat menyebabkan kesalahan dalam penanganan gangguan/ komplain, serta dalam penyampaiannya kepada staf teknisi yang dikirim melalui *email* terkadang terlewati sehingga dalam penanganannya tidak terlaksanakan. Selain itu dalam pencatatan gangguan/komplain masih dituliskan secara manual di menggunakan excel oleh *admin*, dari laporan gangguan/ komplain tersebut akan dihitung dan dievaluasi, yang selanjutnya akan dilakukan perbaikan pada sistem yang paling banyak dikomplain, akan tetapi pencatatan tersebut terkadang tidak akurat atau keliru yang menyebabkan perbaikan pada sistem yang bermasalah tidak optimal. Lalu selain kendala yang dialami oleh *admin helpdesk*, mahasiswa dan dosen pun mengalami kesulitan dalam mengecek status gangguan/komplain mereka, yang menyebabkan ketidakjelasan apakah gangguan sudah ditangani atau belum.

Berdasarkan permasalahan tersebut diperlukan suatu alat atau aplikasi yang dapat menangani pelayanan yang di dalamnya meliputi pelaporan dan penanganan, serta pengelolaan yang meliputi pencatatan, penghitungan, dan penyampaian gangguan/komplain secara cepat dan akurat. Maka penelitian ini bertujuan untuk membangun Aplikasi Pengelolaan Layanan *Helpdesk* yang akan dikelola oleh *admin*, aplikasi yang dibuat berbasis *website* menggunakan metode *waterfall*, aplikasi berbasis *website* agar memudahkan pengguna baik *admin*, mahasiswa dan dosen dalam mengaksesnya dimanapun dan kapanpun dengan hanya menggunakan perangkat apapun yang memiliki *web browser* dan jaringan *internet*. Hasil dari penelitian ini diharapkan dapat membantu menangani dan mempermudah pelayanan dan pengelolaan. Maka akan dilakukan pengembangan aplikasi yang akan dituangkan dalam penelitian yang berjudul **“PENGEMBANGAN APLIKASI PENGELOLAAN LAYANAN HELPDESK BERBASIS WEB DI UNIVERSITAS WIDYATAMA”**.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang penelitian ini, penulis mengidentifikasi masalah dalam penelitian ini adalah :

1. Bagaimana agar mahasiswa dan dosen dapat melakukan pelaporan komplain secara daring dengan mudah serta membuat mahasiswa dan dosen dapat mengecek status komplain sudah teratas atau belum?
2. Bagaimana mempermudah tim *helpdesk* Universitas Widyatama dalam melakukan pelayanan dan pengelolaan komplain serta penyampaian sistem yang bermasalah kepada staf teknisi?

### **1.3 Tujuan Penulisan**

Berdasarkan uraian latar belakang dan rumusan masalah di atas, maka maksud dari penelitian ini adalah sebagai berikut:

1. Membangun aplikasi pengelolaan layanan *helpdesk* yang memudahkan mahasiswa dan dosen dalam melakukan pelaporan komplain secara daring, serta membuat mahasiswa dan dosen dapat mengecek status sudah ditangani atau belum.
2. Membangun aplikasi pengelolaan layanan *helpdesk* yang memudahkan tim *helpdesk* Widyatama dalam melakukan pelayanan dan pengelolaan komplain baik pencatatan atau penghitungan serta penyampaian sistem yang bermasalah kepada staf teknisi.

### **1.4 Batasan Masalah**

Ruang lingkup aplikasi Pengelolaan Layanan ini memiliki batasan masalah sebagai berikut.

1. Aplikasi yang dibangun menggunakan framework Laravel, Bahasa pemrograman yang digunakan adalah php, dan MySQL sebagai *database*.
2. Aplikasi ini hanya membantu memberikan alternatif terbaik untuk melakukan pengelolaan dan pelayanan komplain bagi mahasiswa dan dosen.
3. Penghitungan komplain dilakukan berdasarkan parameter yang ada.
4. Penelitian ini hanya dilakukan pada lingkungan Universitas Widyatama dan disesuaikan dengan kondisi Universitas.

5. Dalam penelitian ini pengembangan aplikasi hanya sampai pada tahap pengujian dan dalam pengembangannya, aplikasi tersebut dikembangkan oleh pihak Universitas sesuai kebutuhan.

## **1.5 Metode Penelitian**

Adapun metode penelitian ini adalah sebagai berikut :

### **1.5.1 Teknik Pengumpulan Data**

Adapun teknik pengumpulan data yang penulis gunakan, yaitu:

- a) Studi observasi, yaitu peninjauan secara langsung ke Universitas Widyatama.
- b) Wawancara, yaitu dengan melakukan wawancara langsung dengan para staf pada unit TI Universitas Widyatama.
- c) Studi pustaka, yaitu dengan mempelajari buku-buku, referensi-referensi yang berkaitan dengan pembuatan laporan ini.

### **1.5.2 Metode Pengembangan**

Metode pengembangan aplikasi pengelolaan layanan *helpdesk* ini menggunakan metode *waterfall*. Metode air terjun atau yang sering disebut metode *waterfall* seing dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*” dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dari analisis, desain, pengembangan, pengujian (Pressman, 2012). Adapun pengembangan ini memiliki tahapan sebagai berikut:

## 1. Analisis Sistem

Pada tahapan ini bertujuan untuk memahami sistem yang sedang berjalan agar dapat mendefinikan permasalahan dari sistem, yang selanjutnya dapat menentukan kebutuhan sistem secara garis besar dan diterapkan dalam tahap perancangan.

## 2. Perancangan Desain.

Pada tahapan ini bertujuan untuk memberikan gambaran sistem rancang bangun yang lengkap tentang sistem yang akan dibangun, yang selanjutnya perancangan ini digunakan untuk persiapan implementasi kebutuhan perangkat lunak dari tahap analisis ke representasi desain agar dapat diimplementasikan menjadi suatu perangkat lunak pada tahap selanjutnya.

## 3. Pengembangan

Setelah tahap perancangan sistem selanjutnya adalah tahap pengembangan yaitu dengan dilakukannya perubahan rancangan sistem kedalam bahasa pemrograman yang diinginkan, pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi modul program, antarmuka, dan basis data.

## 4. Pengujian

Pada tahap ini dilakukan uji coba untuk mendapatkan serta memastikan bahwa perangkat lunak yang dihasilkan *valid*, bekerja dengan baik dan sesuai dengan kebutuhan yang telah direncanakan.

Berdasarkan poin-poin tersebut dalam tahapan perancangan ini penilitian mengusulkan pemodelan sistem yang baru menggunakan pemodelan fisik (*physical model*) dengan membuat *usecase diagram*, *activity diagram*, *class diagram*, dan *Sequence Diagram*. Pengkodean dilakukan di lingkungan sistem

operasi Linux Mint dengan menggunakan bahasa pemrograman *web* PHP, dan untuk pembuatan basis data menggunakan aplikasi Mysql.

### **1.6 Sistematika Penulisan**

Adapun sistematika penulisan laporan penelitian ini terdiri dari lima bab, dengan rincian sebagai berikut :

**Bab I Pendahuluan**, membahas tentang latar belakang masalah, identifikasi masalah, tujuan penulisan, batasan masalah, metode penelitian dan sistematika penulisan.

**Bab II Landasan teori**, Pada bagian ini akan menguraikan teori yang mendukung laporan penelitian ini.

**Bab III Analisis Sistem dan Perancangan Sistem**, merupakan bagian ketiga dari laporan penelitian ini yang membahas tentang mengidentifikasi masalah dan menganalisa kebutuhan sistem dan membahas tentang perancangan sistem meliputi perancangan sistem dan perancangan antar muka.

**Bab IV Implementasi dan Pengujian**, merupakan bagian empat dari laporan penelitian ini yang membahas tentang implementasi object-object sistem, bentuk tampilan struktur program, langkah-langkah menjalankan program dan pengujian sistem.

**Bab V Penutup**, merupakan bagian lima dari laporan penelitian ini yang berisi kesimpulan dan saran dari hasil penelitian ini.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Definisi *Website***

*Website* merupakan sebuah media yang memiliki banyak halaman yang saling terhubung (*hyperlink*), dimana website memiliki fungsi dalam memberikan informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya. *Website* pada saat sekarang ini umumnya telah bersifat dinamis, meskipun dahulu juga ada *website* yang bersifat statis, namun *website* statis telah jarang dan bahkan hampir tidak ada lagi ditemukan (Elgamar, 2020). Selain itu *website* merupakan salah satu fasilitas di internet yang luas, dan merupakan salah satu media informasi dan sekaligus sebagai sarana promosi. (Susilawati *et al.*, 2020)

#### **2.2 Definisi Sistem**

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan. Sistem merupakan sebuah objek yang dikaji atau dipelajari, dimana memiliki karakteristik tertentu atau spesifikasi tersendiri. (Situmorang, 2019).

### 2.3 ***Unified Modeling Language (UML)***

*Unified modeling language* (UML) merupakan alat perancangan sistem yang berorientasi pada ob UML didasari oleh konsep yaitu konsep permodelan *Object Oriented* (OO), karena konsep ini merupakan sistem seperti kehidupan nyata yang didominasi oleh obyek dan dinotasikan dalam simbol yang cukup spesifik maka *Object Oriented* (OO) memiliki proses standard dan bersifat *independent*.

UML diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

UML mempunyai tiga kategori utama yaitu struktur diagram, *behaviour diagram* dan *interaction diagram*. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi (Fatmasari and Sauda, 2020). Adapun jenis diagram yang paling umum digunakan antara lain :

1. *Use case Diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. Use case bekerja dengan mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai.

Tabel 2.1 Simbol *Use Case Diagram* (Suryana, 2018)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasiikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2	----->	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3	<-----	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4	----->	<i>Include</i>	Menspesifikasiikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5	<-----	<i>Extend</i>	Menspesifikasiikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6	-----	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasiikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2. *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Gambaran *sequence diagram* dibuat minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan, maka *sequence diagram* yang harus dibuat juga semakin banyak.

Tabel 2.2 Simbol *Sequence Diagram* (Suryana, 2018)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
2		<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
3		<i>Object Message</i>	Menggambarkan pasan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
4		<i>Message to Self</i>	Menggambarkan pasan/hubungan objek itu sendiri yang menunjukkan urutan kejadian yang terjadi.

3. *Class diagram* merupakan gambaran struktur sistem dari segi pendefinisian kelas- kelas yang akan dibuat untuk membangun sistem. Class diagram terdiri dari atribut dan operasi dengan tujuan membuat pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai.

*Class* memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda.

Tabel 2.3 *Class Diagram* (Suryana, 2018)

NO	GAMBAR	NAMA	KETERANGAN
1	—	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2	◇	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3	□	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4	○	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5	◀-----	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6	----->	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7	—	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

4. *Activity diagrams* merupakan diagram yang menggambarkan *workflow* atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. (Putra et al., 2019).

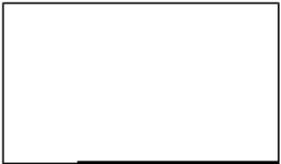
Tabel 2.4 Simbol *Activity Diagram* (Suryana, 2018)

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

## 2.4 Entity Relationship Diagram

Menurut Silberschatz dalam jurnal Benny Bunardi, *Entity Relationship Diagram* atau ERD adalah sebuah diagram yang digunakan untuk menggambarkan sebuah pemodelan penyimpanan data yang terdiri dari kumpulan objek-objek yang disebut dengan *entities* dan *relationship* di antara objek-objek tersebut. Sebuah *entity* adalah "benda" atau "objek" yang ada pada dunia nyata dan dapat dibedakan dengan objek lainnya. Sebagai contoh, tiap-tiap manusia adalah sebuah *entity* dan akun bank dapat dianggap sebagai *entity* (Bunardi *et al.*, 2019).

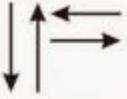
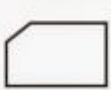
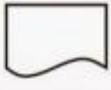
Tabel 2.5 Simbol *Entity Relationship Diagram* (Triaulia, 2021)

No.	Simbol	Keterangan
1.		Entitas, adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
2.		Relasi, menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
3.		Atribut, berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah)
4.		Garis, sebagai penghubung antara relasi dengan dengan entitas, relasi dan entitas dengan atribut.

## 2.5 Flowchart

*Flowchart* mendeskripsikan detail sebuah proses, tahapan dan urutannya secara grafis. *Flowchart* berisi bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. (Astuti, 2016)

Berikut simbol – simbol pada *flowchart*:

	<b>Flow Direction symbol</b> Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		<b>Simbol Manual Input</b> Simbol untuk pemasukan data secara manual on-line keyboard
	<b>Terminator Symbol</b> Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		<b>Simbol Preparation</b> Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	<b>Connector Symbol</b> Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		<b>Simbol Predefine Proses</b> Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	<b>Connector Symbol</b> Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		<b>Simbol Display</b> Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	<b>Processing Symbol</b> Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		<b>Simbol disk and On-line Storage</b> Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	<b>Simbol Manual Operation</b> Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		<b>Simbol magnetik tape Unit</b> Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	<b>Simbol Decision</b> Simbol pemilihan proses berdasarkan kondisi yang ada.		<b>Simbol Punch Card</b> Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	<b>Simbol Input-Output</b> Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		<b>Simbol Dokumen</b> Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 2.1 Simbol Flowchart (Astuti,2016)

## 2.6 Definisi Pengelolaan

Pengelolaan merupakan terjemahan dari kata “*management*”, istilah Inggris tersebut lalu di Indonesia menjadi manajemen. Manajemen berasal dari kata *to manage* yang artinya mengatur, pengaturan dilakukan melalui proses dan diatur berdasarkan urutan dari fungsi-fungsi manajemen. Jadi manajemen merupakan suatu proses untuk mewujudkan tujuan yang di inginkan melalui aspek-aspeknya antara lain *planning*, *organising*, *actuating*, dan *controlling*.

Secara umum pengelolaan merupakan kegiatan merubah sesuatu hingga menjadi baik berat memiliki nilai- nilai yang tinggi dari semula. Pengelolaan dapat juga diartikan sebagai untuk melakukan sesuatu agar lebih sesuai serta cocok dengan kebutuhan sehingga lebih bermanfaat. (Suawa *et al.*, 2021).

Dari definisi diatas, pengelolaan atau yang sering disebut menejemen pada umumnya sering dikaitkan dengan aktivitas-aktivitas dalam organisasi berupa perencanaan, pengorganisasian, pengendalian, pengarahan, dan pengawasan.

## **2.7 Definisi Layanan**

Secara umum layanan adalah proses memenuhi kebutuhan orang lain secara langsung atau tidak langsung dengan harapan memperoleh sesuatu. Dalam kamus bahasa Indonesia usaha melayani kebutuhan orang lain dengan memperoleh imbalan (uang). Pengertian layanan adalah menyediakan atau memenuhi segala apa yang menjadi kebutuhan bagi orang lain sebagai penerima layanan tersebut. (Kai *et al.*, 2018).

## **2.8 Definisi Helpdesk**

*Helpdesk ticketing system* adalah layanan teknologi informasi berupa *support/maintenance* dan berisikan laporan permintaan bantuan, masalah yang menyangkut maintenance IT yang dihadapi kantor pusat dan kantor cabang (Koneksi, PC rusak dll), yang dibuat untuk memuat keluhan-keluhan seluruh pegawai di kantor pusat dan cabang secara tersusun.(Wardhani *et al.*, 2020) *Helpdesk* pada dasarnya adalah *point* dimana masalah atau *issue* dilaporkan dan diatur secara terurut dan diorganisasikan dari perspektif umum. *Helpdesk* juga

berfungsi sebagai pelengkap dari sebuah fungsi pelayanan dan bertanggung jawab sebagai sumber dari pemecahan masalah atau issue lainnya. Adapun fungsi *helpdesk* ini adalah untuk membantu pekerja perusahaan yang mengajukan permasalahan teknis kepada tim pendukung, agar masalah dapat tersampaikan dan dapat diselesaikan.(Rachmawati, *et al.*, 2019)

## 2.9 Apache

Apache adalah *web server open source* yang digunakan untuk menghubungkan antara *server* dan *user (browser)*. Apache adalah salah satu *software web server* yang populer dan yang paling banyak digunakan didunia. *Software* ini dikelola oleh perusahaan Apache Software Foundation. Apache diluncurkan pada tahun 1995 dan tetap popular hingga saat ini.

## 2.10 Bahasa Pemrograman PHP

PHP : *Hypertext Preprocessor* (PHP) merupakan sebuah bahasa pemrograman digunakan pada *website*. PHP berperan sebagai pemroses data pada sisi *server* sesuai yang diminta oleh *client* menjadi informasi yang siap ditampilkan, juga sebagai penghubung aplikasi *web* dengan *database* (Abdulloh, 2018). Banyak aplikasi dibuat menggunakan bahasa pemrograman, contoh aplikasi terkenal yang dibuat oleh bahasa pemrograman ini yaitu Wordpress, Drupal, Joomla, dan lain-lain.

## 2.11 Framework Laravel

*Framework* adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan atau isu-isu kompleks. Laravel adalah salah

satu dari *framework* yang bersifat *open source* yang dapat digunakan secara gratis. Laravel dikembangkan oleh programmer asal Amerika yaitu Taylor Otwell pada tahun 2011. Pengembangan Laravel memudahkan programmer dalam membuat aplikasi *web* berbasis PHP. Laravel salah satu *framework* populer memiliki banyak macam fitur yang membantu para programmer dalam membuat Aplikasi.

Kelebihan Laravel :

1. Laravel merupakan *framework* PHP yang ekspresif, artinya sintaks pada Laravel menggunakan bahasa yang mudah dimengerti, sehingga memudahkan pemula dalam memahami kegunaan suatu sintaks.
2. Laravel memiliki dokumentasi cukup lengkap, setiap versi memiliki dokumentas tersendiri mulai dari cara instalasi hingga penggunaan fitur-fiturnya.
3. Laravel didukung oleh *Composer*, yaitu sebuah *Dependency Management PHP*, yang membantu pengguna mendapatkan dengan mudah *library-library* Laravel dari internet.
4. Laravel digunakan oleh banyak programmer, sehingga banyak *library* yang mendukung Laravel.
5. Laravel memiliki template *engine* tersendiri yang diberi nama *blade* yang memudahkan pengguna dalam menampilkan data pada *template* HTML.  
(Abdulloh, 2018).

## 2.12 Basis Data

Basis data atau *database* adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu

program komputer untuk memperoleh informasi dari basis data tersebut. *Database* adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

*Database* merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu pada tujuan tertentu pula. *Database* adalah susunan record data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna. (Helmund, 2021)

### **2.13 Sistem Manajemen Basis Data**

Pengelolaan atau penggunaan *database* tidak dilakukan pemakai secara langsung, tetapi ditangani oleh sebuah Perangkat Lunak (Sistem) yang khusus. Perangkat lunak ini disebut dengan DBMS (*Database Management System*) atau Sistem Manajemen Basis Data yang sebagaimana menentukan data diorganisasi, disimpan, diubah dan diambil kembali. Fungsi dari DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, konsistensi data, dan sebagainya. (Setya Budi *et al.*, 2022)

Sebuah DBMS melakukan beberapa fungsi penting yang menjamin integritas dan konsistensi data dalam *database*.

1. *Data Storage Management/Manajemen Penyimpanan Data:* DBMS menciptakan struktur kompleks yang diperlukan untuk penyimpanan data dalam

*database* fisik. Ini menyediakan mekanisme untuk pengelolaan penyimpanan permanen data.

2. Manajemen Transaksi: Transaksi adalah serangkaian operasi *Database*, yang dilakukan oleh program aplikasi, yang mengakses atau mengubah isi *Database*. Oleh karena itu, DBMS harus menyediakan mekanisme untuk memastikan bahwa semua pembaruan yang terkait dengan transaksi tertentu dilakukan atau tidak ada satu pun yang dibuat.
3. Layanan Integritas: Integritas *database* mengacu pada kebenaran dan konsistensi data yang disimpan dan sangat penting dalam sistem *database* berorientasi transaksi. Oleh karena itu, DBMS harus menyediakan untuk memastikan bahwa baik data dalam *database* dan perubahan data mengikuti aturan tertentu. Ini meminimalkan redundansi data dan memaksimalkan konsistensi data. Hubungan data yang disimpan dalam kamus data digunakan untuk menegakkan integritas data. Berbagai jenis mekanisme dan batasan integritas dapat didukung untuk membantu memastikan bahwa nilai data dalam *database* valid, bahwa operasi yang dilakukan pada nilai tersebut valid dan *database* tetap dalam keadaan konsisten.
4. Manajemen Pencadangan dan Pemulihan: DBMS menyediakan mekanisme untuk berbagai jenis kegagalan. Ini mencegah hilangnya data. Mekanisme pemulihan DBMS, memastikan bahwa *database* dikembalikan ke keadaan yang konsisten setelah transaksi gagal atau dibatalkan karena sistem crash, kegagalan media, kesalahan perangkat keras atau perangkat lunak, kegagalan daya, dan sebagainya.

5. *Concurrency Control Services*: Karena DBMS mendukung berbagi data di antara banyak pengguna, mereka harus menyediakan mekanisme untuk mengelola akses bersamaan ke *database*. DBMS memastikan bahwa *database* disimpan dalam keadaan yang konsisten dan integritas data dipertahankan. Ini memastikan bahwa *database* diperbarui dengan benar ketika banyak pengguna memperbarui *database* secara bersamaan.
6. *Data Manipulation Management/Manajemen Manipulasi Data*: DBMS melengkapi pengguna dengan kemampuan untuk mengambil, memperbarui dan menghapus data yang ada dalam *database* atau untuk menambahkan data baru ke *database*. Ini termasuk komponen prosesor DML untuk menangani bahasa manipulasi data (DML).
7. *Data Dictionary/Manajemen Katalog Sistem*: DBMS menyediakan kamus data atau fungsi katalog sistem di mana deskripsi item data disimpan dan dapat diakses oleh pengguna. Katalog sistem atau kamus data adalah *Database* sistem, yang merupakan tempat penyimpanan informasi yang menjelaskan data dalam *Database*. Ini adalah data tentang data atau metadata. Misalnya, DBMS akan berkonsultasi dengan katalog sistem untuk memverifikasi bahwa tabel yang diminta ada dan pengguna yang mengeluarkan permintaan memiliki hak akses yang diperlukan.
8. Otorisasi/Manajemen Keamanan : DBMS melindungi *database* terhadap akses yang tidak sah, baik disengaja atau tidak disengaja. Ini melengkapi mekanisme untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses *database*. Ini menciptakan sistem keamanan yang memberlakukan keamanan

pengguna dan privasi data dalam *database*. Aturan keamanan menentukan pengguna mana yang dapat mengakses *database*, item data mana yang dapat diakses setiap pengguna, dan operasi data mana (menambah, menghapus, dan memodifikasi) yang dapat dilakukan pengguna.

9. *Utility Services/Layanan Utilitas*: DBMS menyediakan satu set layanan utilitas yang digunakan oleh DBA dan perancang *Database* untuk membuat, mengimplementasikan, memantau, dan memelihara *Database*. Layanan utilitas ini membantu DBA untuk mengelola *database* secara efektif.

10. Akses *Database* dan Antarmuka Pemrograman Aplikasi: Semua DBMS menyediakan antarmuka untuk memungkinkan aplikasi menggunakan layanan DBMS. Mereka menyediakan akses data melalui bahasa kueri terstruktur (SQL). Bahasa query DBMS berisi dua komponen: a. Bahasa definisi data/*data definition language* (DDL) b. Bahasa manipulasi data/*data manipulation language* (DML) mendefinisikan struktur di mana data disimpan dan DML memungkinkan pengguna akhir untuk mengekstrak data dari *database*. DBMS juga menyediakan akses data ke pemrogram aplikasi melalui bahasa prosedural seperti C, C++, Java dan lain-lain.

11. *Data Independence Services/Layanan Independensi Data*: DBMS harus mendukung independensi program dari struktur *database* yang sebenarnya.

12. *Data Definition Services/Layanan Definisi Data*: DBMS menerima definisi data seperti skema eksternal, skema konseptual, skema internal, dan semua pemetaan terkait dalam bentuk sumber. Ini mengubahnya ke bentuk objek yang

sesuai menggunakan komponen prosesor DDL untuk masing-masing dari berbagai bahasa definisi data (DDL). (Raharjo, 2021)

#### **2.14 Black Box Testing**

*Black box testing* merupakan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan pengguna. (Moch Zawaruddin Abdullah *et al.*, 2021). Proses pengujian *black box* adalah pengujian yang dilakukan dengan cara mencoba program aplikasi dengan memasukkan data ke dalam form - form yang telah disediakan. Pengujian ini memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya semua persyaratan fungsional untuk suatu program. (Wahyudi *et al.*, 2016). Teknik pengujian pada *Black Box* testing ada banyak macamnya yaitu:

- a). Teknik *Equivalence Partitioning* yaitu dengan cara melakukan partition atau pembagian menjadi beberapa partisi dari input data.
- b). Teknik *Boundary Value Analysis* yaitu dengan cara mencari adakah error dari luar atau sisi dalam software, minimum maupun maximum nilai dari error yang di temukan.
- c). Teknik *Fuzzing* yaitu merupakan teknik untuk mencari Bug/gangguan dari *software* dengan menggunakan injeksi data yang terbilang cacat .
- d). Teknik *Cause-Effect Graph*ialah suatu Teknik *testing* dimana menggunakan *graphic* sebagai acuannya. Dimana dalam grafik ini menggambarkan relasi diantara efek dan penyebabnya.

- e). Teknik *Orthogonal Array Testing* adalah jenis Teknik yang digunakan jika input domain yang *relative* terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar.
- f). Teknik *All Pair Testing* yaitu semua pasangan dari test case di desain sedemikian rupa agar dapat di eksekusi semua kemungkinan kombinasi diskrit dari seluruh pasangan berdasar input parameternya, Tujuan *testing* ini adalah memiliki pasangan *test case* yang mencakup semua pasangan tersebut.
- g). Teknik state *Transition*. Teknik ini berguna untuk melakukan pengetesan terhadap kondisi dari mesin dan navigasi dalam bentuk grafik. (Uminingsih *et al.*, 2022). Pengujian yang dilakukan pada sistem informasi penilaian kinerja ini menggunakan masukan data random yang bertujuan untuk memastikan sistem menolak untuk menyimpan data masukan pada database, sehingga sistem dikatakan layak untuk digunakan. Pengujian *Black Box* bertumpu pada memastikan tiap proses sudah berfungsi sesuai dengan kebutuhan yang diharapkan. Penguji dapat menartikan himpunan kondisi masukan dan menjalankan pengujian pada pengkhususan fungsi dari sistem. Sehingga pengujian merupakan suatu cara pelaksanaan program yang bertujuan menemukan kesalahan atau error kemudian memperbaikinya sehingga sistem dapat dikatakan layak untuk digunakan.(Wijaya and Astuti, 2021).

Dari definisi diatas, *Black box testing* merupakan salah satu dari pengujian perangkat lunak dengan mencoba serangkaian percobaan pada input dari program sehingga mendapatkan hasil apakah progam berfungsi sesuai dengan kebutuhan yang diharapkan.

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Pengumpulan Data**

Pengumpulan data merupakan sebuah teknik atau cara yang dilakukan oleh peneliti untuk bisa mengumpulkan data yang terkait dengan permasalahan dari penelitian yang diambil. Teknik pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut:

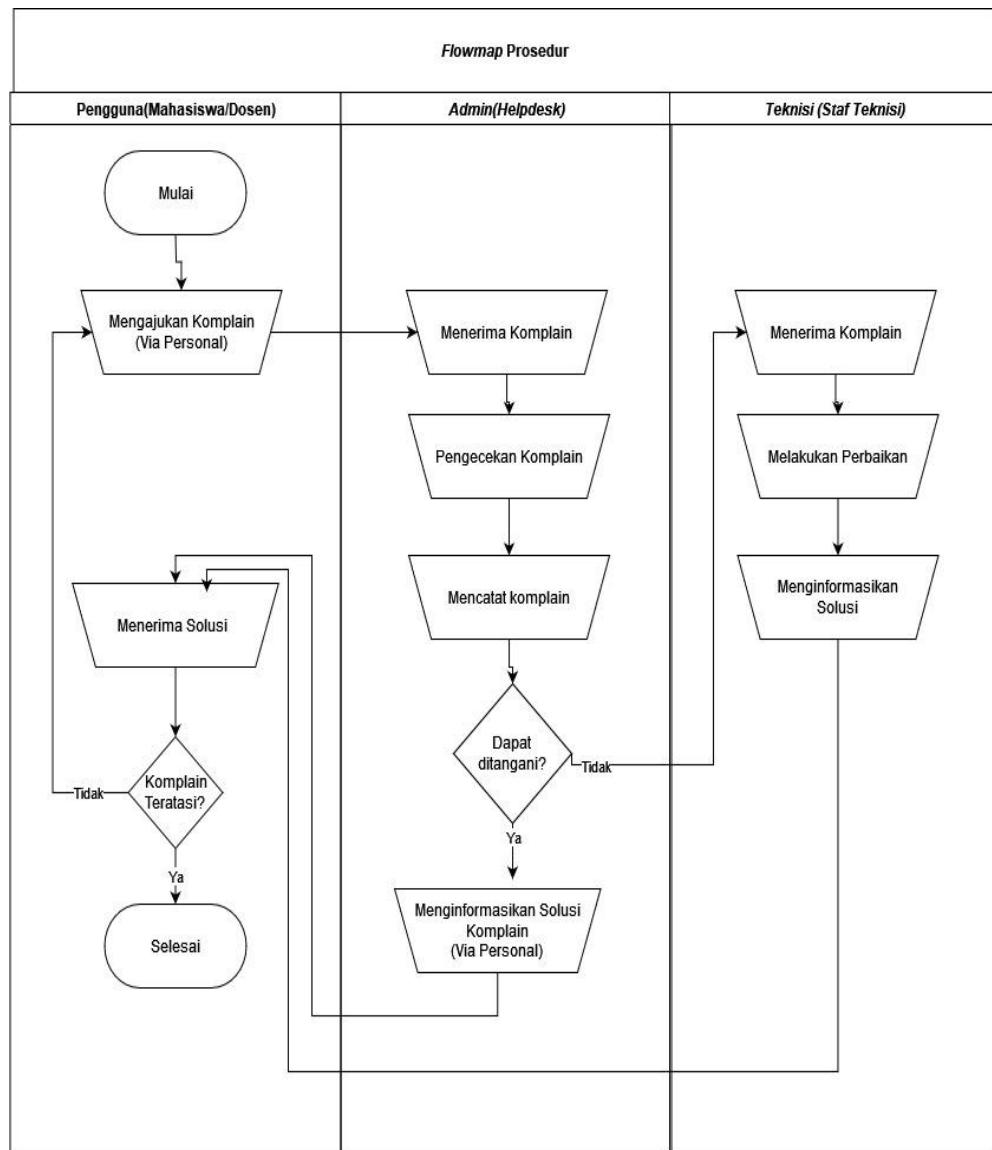
##### **3.1.1 Observasi**

Dalam hal ini penulis peninjauan secara langsung ke Universitas Widyatama untuk melihat proses yang dilakukan dalam penanganan komplain di Universitas Widyatama. Berdasarkan hal tersebut penulis mendapatkan informasi berupa prosedur, perantara, dan cara penanganan yang dilakukan dalam menangani komplain di Universitas Widyatama. Berikut prosedur yang sedang berjalan di Universitas Widyatama :

1. Pengguna mengajukan komplain secara personal ke *admin Helpdesk (admin)* dengan datang secara langsung ke kampus atau secara daring melalui Whatsapp, Gmail.
2. *admin* menerima gangguan yang diajukan Mahasiswa/Dosen.
3. *admin* melakukan pengecekan terhadap komplain yang diajukan.
4. Mencatat data komplain ke Excel

5. Jika *admin* dapat menangani masalah, *admin* dapat langsung menanganinya, apabila tidak, maka *admin* akan mengirimkannya ke bagian teknisi (staf teknisi) melalui *email*.
6. Pengguna menerima solusi baik dari *admin* atau dari teknisi.

Dari prosedur tersebut berikut *flowmap* yang dibuat:



Gambar 3.1 *Flowmap* berdasarkan Prosedur

### 3.1.2 Wawancara

Dalam hal ini penulis melakukan wawancara langsung dengan staf pada *Helpdesk* Universitas Widyatama yaitu Bapak Fandry Nastaliyantono. Berikut ini adalah tabel wawancara yang dilakukan :

Tabel 3.1 Wawancara

No	Pertanyaan	Jawaban
1	Apa itu <i>Helpdesk</i> Universitas Widyatama?	<i>Helpdesk</i> Universitas Widyatama yaitu bagian di Universitas Widyatama yang menerima permasalahan seputar sistem informasi yang ada di Universitas Widyatama yang terjadi pada mahasiswa dan dosen, lalu mengecek dan memberi solusi.
2	Bagaimana penangan yang dilakukan dalam menangani komplain yang masuk?	Penangan yang dilakukan sama seperti pada umumnya yaitu menerima komplain, dan melakukan pengecekan dan pengujian pada sistem terhadap sistem yang terkendala. Yang selanjutnya akan dikirim ke bagian teknisi untuk dilakukan perbaikan.
3	Sistem apa saja yang biasanya terkendala?	Sistem yang ditangani <i>Helpdesk</i> Universitas Widyatama meliputi, Sistem Portal Mahasiswa, Dosen, Absensi, <i>Email</i> , dan Pembelajaran Online.
4	Perantara apa saja yang digunakan dalam menangani komplain?	Dalam menangani komplain perantara yang digunakan yaitu berupa <i>email</i> , aplikasi <i>chat</i> seperti <i>whatsapp</i> , telegram.

### 3.1.3 Studi Pustaka

Pada tahap ini dilakukan studi pustaka terhadap referensi yang berkaitan dengan penelitian. Berikut ini adalah tabel literatur yang menjadi referensi penelitian:

Tabel 3.2 Studi Pustaka

No	Literatur	Pembahasan
1	Mustopa, Ali. 2017. "SISTEM INFORMASI IT-HELPDESK PADA UNIVERSITAS AMIKOM YOGYAKARTA BERBASIS WEB." JIKO (Jurnal Informatika Dan Komputer) 2(2):93. doi: 10.26798/jiko.2017.v2i2.71.	Penelitian ini bertujuan untuk membuat aplikasi untuk membantu mengatasi permasalahan terkait penggunaan fasilitas TIK di Universitas AMIKOM Yogyakarta. IT Helpdesk menyediakan beberapa bentuk layanan yang dapat dimanfaatkan oleh civitas akademik di Universitas AMIKOM Yogyakarta dalam menggunakan layanan TIK.
2	Azmi Farhan. 2022. "Rancang Bangun Aplikasi Pengelolaan Layanan Ticketing Helpdesk Berbasis Web (Studi Kasus: Pt. Ussi Bandung)." 1–49.	Penelitian ini bertujuan untuk membuat aplikasi membantu Untuk membantu intensitas pekerjaan IT Support dalam melayani keluhan dan juga dapat membantu untuk memonitoring kinerja dari staf IT Support untuk manajemen pada PT. USSI.
3	Rizal, Nanda Muhamad. 2020. "Penerapan Customer Relationship Management (Crm ) Berbasis Website ( Studi Kasus Di Toko Online Shop Fauxhawk Store Bandung )." 1–49.	Penelitian ini bertujuan untuk mengimplementasikan suatu produk teknologi komunikasi informasi yang dapat membantu usaha ini menjadi lebih efektif dalam meningkatkan layanan marketing dan customer support, serta dapat menunjang pengambilan keputusan yang lebih baik bagi pemilik usaha.
4	Kurniawan, Nurul Apri. 2019. "PEMBUATAN APLIKASI HELPDESK BERBASIS WEB PADA BAGIAN IT SUPPORT TERHADAP PERMASALAHAN PEGAWAI TENTANG TEKNOLOGI INFORMASI DI PT. PENERBIT ERLANGGA."	Penelitian ini bertujuan untuk membuat aplikasi yang mempermudah dalam melakukan pelaporan keluhan, memonitoring teknisi dan data keluhan serta melakukan penjadwalan yang tersusun untuk teknisi dalam melakukan perbaikan tentang teknologi informasi.pada PT. Penerbit Erlangga Bandung.

### **3.2 Analisis Sistem**

Dalam pembuatan suatu program aplikasi, terlebih dahulu dilakukan perancangan pengembangan perangkat lunak berdasarkan pengumpulan data dan kebutuhan dari pengguna yang menggunakan sistem ini. Adapun langkah-langkah atau tahapan pembangunan sistem ini adalah sebagai berikut :

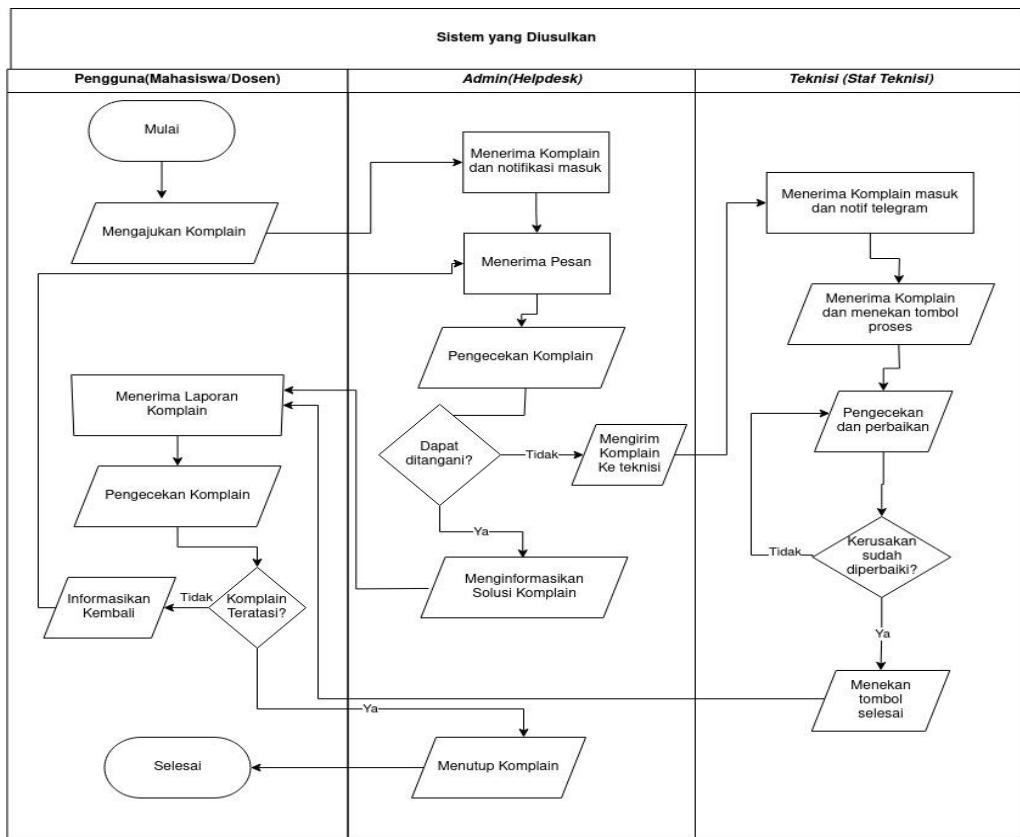
#### **3.2.1 Analisis Sistem yang Sedang Berjalan**

Berdasarkan hasil observasi, didapatkan beberapa masalah dari prosedur yang diobservasi yaitu:

1. Dalam hal pelaporan gangguan/ komplain pengguna mengajukan masih harus melaporkan secara personal kepada *admin Helpdesk (admin)* dengan datang secara langsung ke kampus.
2. Semua prosedur masih secara manual seperti pencatatan komplain masih diketik secara manual di Excel yang memakan waktu lama dalam melayani komplain.
3. Jika *admin* tidak bisa menangani masalah, *admin* akan memberikannya ke teknisi melalui *email* yang memungkinkan teknisi tidak membuka *email* atau terlewati jika banyak *email* yang masuk.

### **3.3 Perancangan Sistem yang Diusulkan**

Sistem yang dikembangkan merupakan PENGEMBANGAN APLIKASI PENGELOLAAN LAYANAN *HELPDESK* BERBASIS WEB DI UNIVERSITAS WIDYATAMA, berikut adalah prosedur umum sistem tersebut.



Gambar 3.2 Flowmap Prosedur Sistem yang diusulkan

#### Prosedur Pengajuan Komplain/Gangguan:

1. Pengguna mengajukan komplain melalui *website*.
2. Komplain dan pesan yang masuk diterima oleh *admin helpdesk (admin)* dan notifikasi masuk.
3. *Admin* melakukan pengecekan komplain yang masuk.
4. Jika *admin* dapat menangani masalah, *admin* dapat langsung menanganinya dan memberi solusi, apabila tidak, maka *admin* akan mengirimkannya ke bagian teknisi (staf teknisi).

5. Teknisi menerima komplain masuk dan notifikasi telegram. Lalu menekan tombol proses untuk menerima komplain dan dilanjutkan dengan perbaikan perangkat/sistem.
6. Jika sudah teknisi akan menekan tombol selesai.
7. Pengguna menerima laporan komplain dan akan memastikannya,
8. Jika komplain sudah teratasi *admin* akan menutup komplain yang diajukan, namun jika gangguan yang diajukan belum teratasi maka *admin* akan melakukan pengecekan kembali.

### **3.3.1 Hasil analisis**

Berdasarkan analisis masalah pada sistem berjalan dan gambaran sistem yang akan dibangun, maka dapat disimpulkan bahwa sistem yang akan dibangun harus memiliki fungsi sebagai berikut :

1. Sistem dapat menerima komplain atau gangguan secara daring.
2. Sistem dapat menampung dan mengolah data komplain atau gangguan mahasiswa dan dosen yang dikirim.
3. Sistem dapat mengidentifikasi mahasiswa/dosen yang mengajukan komplain.
4. Sistem menampilkan penghitungan secara akurat, sehingga admin *helpdesk* dapat dengan mudah memantau sistem yang bermasalah.

### **3.4 Perancangan Sistem**

Perancangan sistem yang akan dibangun, pada aplikasi pengelolaan layanan *helpdesk* ini menangani pengolahan data-data mahasiswa/dosen di Universitas Widyatama, meliputi data-data pribadi berupa nama, nomor telepon

dan komplain yang diajukan. Sistem ini digunakan mahasiswa/dosen Universitas Widyatama dalam menyampaikan komplain/gangguannya. Perancangan sistem dibuat menggunakan UML. Berdasarkan hasil analisis, sistem yang akan dibuat memiliki beberapa kemampuan yang dijabarkan sebagai berikut:

### **3.4.1 *Unified Modelling Language* (UML)**

UML adalah notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem ini berisi informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer. Berdasarkan hasil analisis, sistem yang akan dikembangkan memiliki beberapa kemampuan yang dijabarkan sebagai berikut :

#### **3.4.1.1 Aktor**

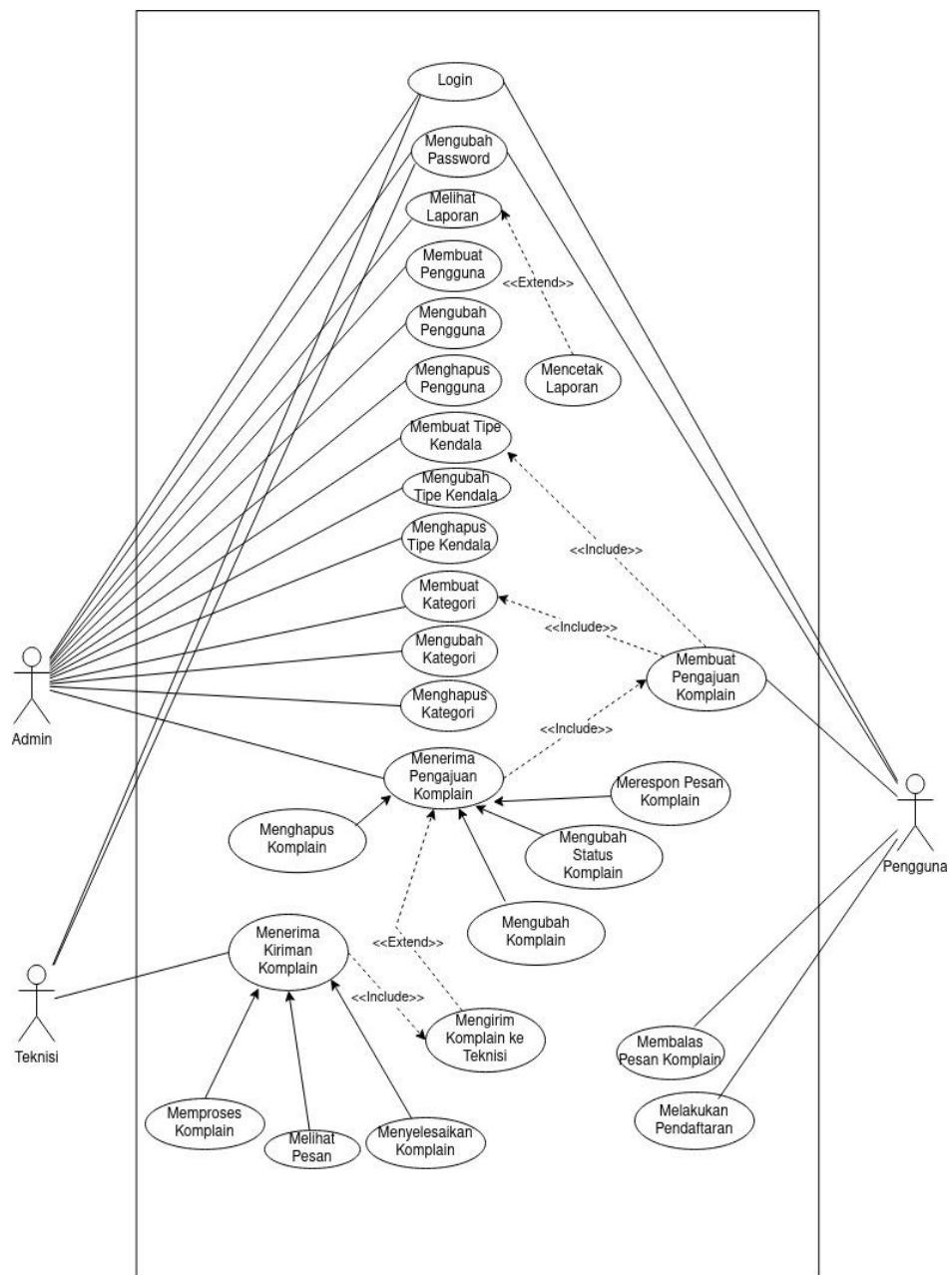
Sebuah use case tidak dapat bertindakan sendiri, oleh sebab itu use case memerlukan Actor / Aktor untuk memulai sebuah aksi. Aktor menggambarkan pihak-pihak yang berperan dalam sistem. Pada sistem ini ada beberapa aktor yang terdapat pada sistem ini :

Tabel 3.3 Aktor

No	Aktor	Deskripsi
1	Admin	Fungsi Admin di sini adalah admin <i>helpdesk</i> yang mengelola seluruh data mulai dari melihat informasi di menu utama/ <i>dashboard</i> , mengelola data master, mengelola komplain, dan mengelola pengguna-pengguna.
2	Teknisi	Teknisi di sini adalah seorang memproses komplain yang dikirimkan oleh <i>admin</i> . Teknisi bertugas memperbaiki perangkat/sistem.
3	Pengguna	Fungsi pengguna di sini adalah mahasiswa/dosen yang melakukan pelaporan.

### 3.4.1.2 Use Case Diagram

Pada *Use Case* diagram ini, digambarkan fitur dan aktor yang terdapat pada sistem yang akan dibuat, fitur tersebut dibagi ke dalam beberapa aktor sebagai berikut.



Gambar 3.3 Use Case Diagram

### 3.4.1.3 Skenario Use Case

#### 3.4.1.3.1 Skenario Use Case Admin

Nama *Use case* : Menerima Pengajuan Komplain

Aktor : Teknisi

Tujuan : Menerima pengajuan Komplain yang dikirimkan aktor pengguna.

Tabel 3.4 Menerima Pengajuan Komplain

AKTOR	SISTEM
1. Aktor mengakses menu komplain	2. Sistem menampilkan halaman menu komplain berisi daftar pengajuan komplain

Nama *Use case* : Mengubah Komplain

Aktor : Admin

Tujuan : Mengubah Komplain

Tabel 3.5 Mengubah Komplain

AKTOR	SISTEM
1 Aktor menekan menu complain	
	2. Sistem menampilkan halaman menu komplain
3. Aktor menekan tombol kunci (tombol ubah)	
	4. Sistem menampilkan form ubah complain
5. Aktor mengubah data pada form	
6. Aktor menekan tombol simpan	
	7. Sistem menyimpan data
	8. Sistem menampilkan data komplain

Nama *Use case* : Menghapus Komplain

Aktor : Admin

Tujuan : Menghapus Komplain

Tabel 3.6 Menghapus Komplain

AKTOR	SISTEM
1. Aktor menekan menu complain	
	2. Sistem menampilkan halaman menu komplain
3. Aktor menekan tombol sampah (tombol hapus)	
	4. Sistem menghapus data yang dipilih
	5. Sistem menampilkan halaman data komplain

Nama *Use case* : Merespon Pesan

Aktor : Admin

Tujuan : Merespon Pesan Komplain yang dikirimkan pengguna

Tabel 3.7 Merespon Pesan Komplain

AKTOR	SISTEM
1. Aktor mengakses menu komplain/detail komplain	
	2. Sistem menampilkan halaman complain
3. Aktor menekan tombol pesan(tombol surat) pada komplain	
	4. Sistem menampilkan halaman pesan complain
5. Aktor mengisi kolom pesan	
6. Aktor menekan tombol simpan	
	7. Sistem menampilkan pesan yang dibuat aktor di halaman pesan

Nama *Use case* : Mengubah Status

Aktor : Admin

Tujuan : Mengubah status Komplain

Tabel 3.8 Mengubah Status

AKTOR	SISTEM
1. Aktor mengakses menu komplain	
	2. Sistem menampilkan halaman komplain
3. Aktor menekan tombol surat(tombol pesan) pada komplain	
	4. Sistem menampilkan halaman pesan
5. Aktor menekan tombol <i>close/open</i>	
	6. Sistem mengubah status komplain.

Nama *Use case* : Mengirim Komplain Ke Teknisi

Aktor : Admin

Tujuan : Mengirim komplain ke komplain masuk teknisi dan notifikasi ke telegram teknisi.

Tabel 3.9 Mengirim Komplain Ke Teknisi

AKTOR	SISTEM
1. Aktor mengakses menu komplain	
	2. Sistem menampilkan halaman complain
3. Aktor menekan tombol kapal kertas (tombol kirim ke teknisi) pada halaman komplain	
	4. Sistem mengirimkan komplain ke komplain masuk teknisi dan notifikasi ke telegram teknisi.

Nama *Use case* : Membuat Kategori

Aktor : Admin

Tujuan : Membuat Kategori Komplain

Tabel 3.10 Membuat Kategori

AKTOR	SISTEM
1. Aktor menekan menu master data, lalu kategori	
	2. Sistem menampilkan halaman kategori
3. Aktor menekan tombol tambah data	
	4. Sistem menampilkan form tambah
5. Aktor mengisi form	
6. Aktor menekan tombol simpan	
	7. Sistem menyimpan data
	8. Sistem menampilkan data kategori

Nama *Use case* : Mengubah Kategori

Aktor : Admin

Tujuan : Mengubah Kategori Komplain

Tabel 3.11 Mengubah Kategori

AKTOR	SISTEM
1. Aktor menekan menu master data, lalu kategori	
	2. Sistem menampilkan halaman kategori
3. Aktor menekan tombol ubah pada salah satu kategori	
	4. Sistem menampilkan form ubah
5. Aktor mengubah isi form	
6. Aktor menekan tombol simpan	
	7. Sistem menyimpan data
	8. Sistem menampilkan data kategori

Nama *Use case* : Menghapus Kategori

Aktor : Admin

Tujuan : Menghapus Kategori Komplain

Tabel 3.12 Menghapus Kategori

AKTOR	SISTEM
1. Aktor menekan menu master data, lalu kategori	
	2. Sistem menampilkan halaman kategori
3. Aktor menekan tombol hapus pada salah satu kategori	
	4. Sistem menghapus data yang dipilih
	5. Sistem menampilkan halaman data kategori

Nama *Use case* : Membuat Tipe Kendala

Aktor : Admin

Tujuan : Membuat Tipe Kendala

Tabel 3.13 Membuat Tipe Kendala

AKTOR	SISTEM
1. Aktor menekan menu master data, lalu tipe kendala	
	2. Sistem menampilkan halaman tipe kendala
3. Aktor menekan tombol tambah	
	4. Sistem menampilkan form tambah
5. Aktor mengisi form	
6. Aktor menekan tombol simpan	
	7. Sistem menyimpan data
	8. Sistem menampilkan data tipe kendala

Nama *Use case* : Mengubah Tipe Kendala

Aktor : Admin

Tujuan : Mengubah Tipe Kendala

Tabel 3.14 Mengubah Tipe Kendala

AKTOR	SISTEM
1. Aktor menekan menu master data, lalu tipe kendala	
	2. Sistem menampilkan halaman tipe kendala
3. Aktor menekan tombol ubah pada salah satu tipe kendala	
	4. Sistem menampilkan form
5. Aktor mengubah isi form	
6. Aktor menekan tombol simpan	
	7. Sistem menyimpan data
	8. Sistem menampilkan tipe kendala

Nama *Use case* : Menghapus Tipe Kendala

Aktor : Admin

Tujuan : Menghapus Tipe Kendala

Tabel 3.15 Menghapus Tipe Kendala

AKTOR	SISTEM
<b>Hapus Tipe Kendala</b>	
1. Aktor menekan menu master data, lalu tipe kendala	
	2. Sistem menampilkan halaman tipe kendala
3. Aktor menekan tombol hapus pada salah satu tipe kendala	
	4. Sistem menghapus data yang dipilih
	5. Sistem menampilkan halaman data tipe kendala

Nama *Use case* : Membuat Pengguna

Aktor : Admin

Tujuan : Membuat Pengguna Baru.

Tabel 3.16 Membuat Pengguna

AKTOR	SISTEM
1. Aktor menekan menu pengguna	
	2. Sistem menampilkan halaman pengguna
3. Aktor menekan tombol tambah data	
	4. Sistem menampilkan form tambah pengguna
5. Aktor mengisi form	
6. Aktor menekan tombol simpan	
	7. Sistem menyimpan data
	8. Sistem menampilkan data pengguna

Nama *Use case* : Mengubah Pengguna

Aktor : Admin

Tujuan : Mengubah data pengguna.

Tabel 3.17 Mengubah Pengguna

AKTOR	SISTEM
1. Aktor menekan menu pengguna	
	2. Sistem menampilkan halaman pengguna
3. Aktor menekan tombol ubah pada salah satu pengguna	
	4. Sistem menampilkan form ubah pengguna
5. Aktor mengubah isi form	
6. Aktor menekan tombol simpan	
	7. Sistem menyimpan data
	8. Sistem menampilkan data pengguna

Nama *Use case* : Menghapus Pengguna

Aktor : Admin

Tujuan : Menghapus Pengguna.

Tabel 3.18 Menghapus Pengguna

AKTOR	SISTEM
1. Aktor menekan menu pengguna	
	2. Sistem menampilkan halaman pengguna
3. Aktor menekan tombol hapus pada salah satu pengguna	
	4. Sistem menghapus data yang dipilih
	5. Sistem menampilkan halaman data pengguna

Nama *Use case* : Melihat Laporan

Aktor : Admin

Tujuan : Melihat Laporan

Tabel 3.19 Laporan

AKTOR	SISTEM
1. Aktor menekan menu <i>dashboard</i>	
	2. Sistem menampilkan halaman <i>dashboard</i>
3. Aktor memilih tahun, lalu menekan tombol <i>submit</i>	
	4. Sistem menampilkan laporan berdasarkan tahun yang dipilih.

Nama *Use case* : Mencetak Laporan

Aktor : Admin

Tujuan : Mencetak Laporan

Tabel 3.20 Mencetak Laporan

AKTOR	SISTEM
1. Aktor menekan menu <i>dashboard</i>	
	2. Sistem menampilkan halaman <i>dashboard</i>
3. Aktor memilih tahun, lalu menekan tombol <i>print</i>	
	4. Sistem mencetak laporan berdasarkan tahun

### 3.4.1.3.2 Skenario *Use Case* Teknisi

Nama *Use case* : Menerima Kiriman Komplain

Aktor : Teknisi

Tujuan : Menerima Kiriman Komplain yang dikirimkan oleh *admin*.

Tabel 3.21 Menerima Kiriman Komplain

AKTOR	SISTEM
1. Aktor mengakses menu komplain masuk	
	2. Sistem menampilkan halaman menu komplain masuk berisikan kiriman komplain yang dikirimkan.

Nama *Use case* : Melihat Pesan

Aktor : Teknisi

Tujuan : Melihat Pesan Komplain

Tabel 3.22 Melihat Pesan

AKTOR	SISTEM
1. Aktor mengakses menu Komplain Masuk	
	2. Sistem menampilkan halaman Komplain Masuk
3. Aktor menekan tombol pesan pada salah satu komplain.	
	4. Sistem menampilkan halaman pesan komplain

Nama *Use case* : Memproses Komplain

Aktor : Teknisi

Tujuan : Memproses Komplain untuk ditangani oleh teknisi.

Tabel 3.23 Memproses Komplain

AKTOR	SISTEM
1. Aktor mengakses menu Komplain Masuk	
	2. Sistem menampilkan halaman Komplain Masuk
3. Aktor menekan tombol pesan pada komplain	
	4. Sistem menampilkan halaman pesan komplain
5. Aktor menekan tombol proses	
	6. Sistem mengubah status komplain menjadi “Sedang diproses”

Nama *Use case* : Menyelesaikan Komplain

Aktor : Teknisi

Tujuan : Menyelesaikan komplain yang sudah ditangani oleh teknisi.

Tabel 3.24 Menyelesaikan Komplain

AKTOR	SISTEM
1. Aktor mengakses menu Komplain Masuk	
	2. Sistem menampilkan halaman Komplain Masuk
3. Aktor menekan tombol pesan	
	4. Sistem menampilkan halaman pesan komplain
5. Aktor menekan tombol selesai	
	6. Sistem mengubah status komplain menjadi "Selesai"

### 3.4.1.3.3 Skenario *Use Case* Pengguna

Nama *Use case* : Melakukan Pendaftaran

Aktor : Pengguna

Tujuan : Untuk mendaftarkan diri untuk *login* ke website *helpdesk*.

Tabel 3.25 Melakukan Pendaftaran

AKTOR	SISTEM
	1. Sistem menampilkan halaman <i>login</i>
2. Aktor menekan tulisan Cek NPM/NRK	
	3. Sistem menampilkan halaman cek NPM/NRK
4. Aktor memasukan NPM/NRK	
5. Aktor menekan tombol cek	
	6. Sistem menampilkan halaman registrasi
7. Aktor mengisi data biodata dan	

AKTOR	SISTEM
password	
8. Aktor menekan tombol daftar	
	9. Sistem menampilkan pesan berhasil mendaftar

Nama *Use case* : Membuat Pengajuan Komplain

Aktor : Pengguna

Tujuan : Membuat Pengajuan Komplain

Tabel 3.26 Membuat Pengajuan Komplain

AKTOR	SISTEM
	1. Sistem menampilkan halaman utama
2. Aktor menekan tombol tambah	
3. Aktor mengisi kategori, tipe kendala, dan gangguan	
4. Aktor menekan simpan	
	5. Sistem menampilkan halaman utama dengan

Nama *Use case* : Membalas Pesan Komplain

Aktor : Pengguna

Tujuan : Membalas Pesan Komplain

Tabel 3.27 Membalas Pesan Komplain

AKTOR	SISTEM
	1. Sistem menampilkan halaman utama
2. Aktor menekan tombol pesan pada komplain	
	3. Sistem menampilkan halaman pesan komplain
4. Aktor mengisi kolom pesan	
5. Aktor menekan tombol simpan	
	6. Sistem menampilkan pesan

### 3.4.1.3.4 Skenario Use Case Semua Pengguna

Nama *Use case* : *Login*

Aktor : Pengguna, Admin, Teknisi

Tujuan : Masuk ke halaman dasbor *web helpdesk Universitas Widyatama*

Tabel 3.28 *Login*

AKTOR	SISTEM
	1. Sistem menampilkan halaman <i>Login</i>
2. Aktor memasukan NPM/NRK/ <i>Email</i> dan <i>password</i>	
3. Aktor klik <i>Login</i>	
4. Aktor mengisi kolom pesan	
	5. Sistem menampilkan halaman utama aplikasi sesuai masing-masing peran

Nama *Use case* : Mengubah *Password*

Aktor : Pengguna, Admin, Teknisi

Tujuan : Mengubah *Password*

Tabel 3.29 Mengubah *Password*

AKTOR	SISTEM
1. Aktor menekan menu <i>Profile</i>	
	2. Sistem menampilkan halaman <i>profile</i>
3. Aktor mengisi <i>password</i> dan <i>confirm password</i>	
4. Aktor menekan tombol simpan	
	5. Sistem memeriksa data
	6. Sistem menyimpan data

### 3.4.1.4 Activity Diagram

*Activity diagram* adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Berikut adalah gambar dan penjelasan tentang *activity diagram* dari sistem ini.

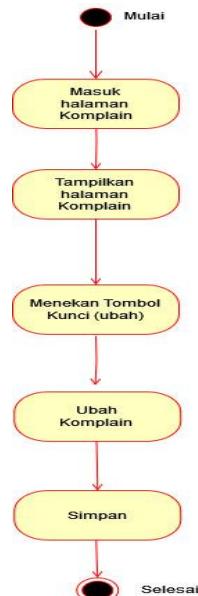
#### 3.4.1.4.1 Activity Diagram Admin

##### 1. Activity Diagram Menerima Pengajuan Komplain



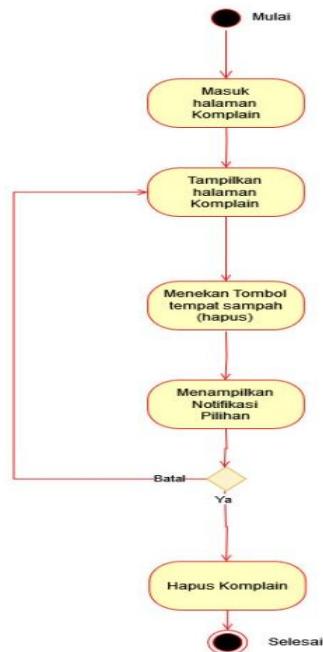
Gambar 3.4 *Activity Diagram* Menerima Pengajuan Komplain

##### 2. Activity Diagram Mengubah Komplain



Gambar 3.5 *Activity Diagram* Mengubah Komplain

### 3. *Activity Diagram* Menghapus Komplain



Gambar 3.6 *Activity Diagram* Menghapus Komplain

### 4. *Activity Diagram* Merespon Pesan Komplain



Gambar 3.7 *Activity Diagram* Merespon Pesan Komplain

### 5. *Activity Diagram* Mengubah Status Komplain



Gambar 3.8 *Activity Diagram* Mengubah Status Komplain

### 6. *Activity Diagram* Mengirim Komplain ke Teknisi



Gambar 3.9 *Activity Diagram* Mengirim Komplain ke Teknisi

### 7. *Activity Diagram* Membuat Kategori



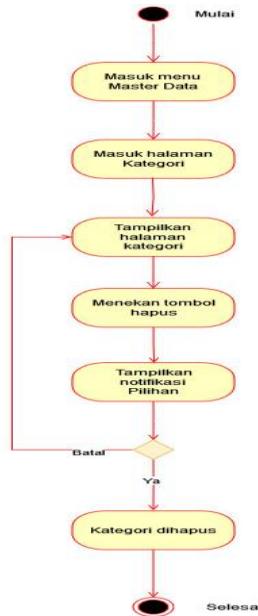
Gambar 3.10 *Activity Diagram* Membuat Kategori

### 8. *Activity Diagram* Mengubah Kategori



Gambar 3.11 *Activity Diagram* Mengubah Kategori

9. *Activity Diagram* Menghapus Kategori



Gambar 3.12 *Activity Diagram* Menghapus Kategori

10. *Activity Diagram* Membuat Tipe Kendala



Gambar 3.13 *Activity Diagram* Membuat Tipe Kendala

### 11. Activity Diagram Mengubah Tipe Kendala



Gambar 3.14 *Activity Diagram* Mengubah Tipe Kendala

### 12. Activity Diagram Menghapus Tipe Kendala



Gambar 3.15 *Activity Diagram* Menghapus Tipe Kendala

13. *Activity Diagram* Kelola Membuat Pengguna



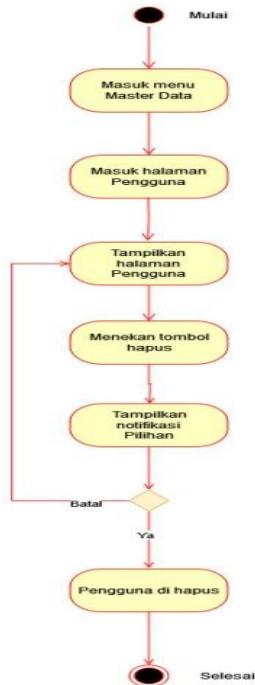
Gambar 3.16 *Activity Diagram* Membuat Pengguna

14. *Activity Diagram* Mengubah Pengguna



Gambar 3.17 *Activity Diagram* Mengubah Pengguna

15. *Activity Diagram* Menghapus Pengguna



Gambar 3.18 *Activity Diagram* Menghapus Pengguna

16. *Activity Diagram* Melihat Laporan



Gambar 3.19 *Activity Diagram* Melihat Laporan

### 17. *Activity Diagram* Mencetak Laporan



Gambar 3.20 *Activity Diagram* Mencetak Laporan

#### 3.4.1.4.2 *Activity Diagram* Teknisi

##### 1. *Activity Diagram* Menerima Kiriman Komplain



Gambar 3.21 *Activity Diagram* Menerima Kiriman Komplain

## 2. Activity Diagram Melihat Pesan



Gambar 3.22 Activity Diagram Melihat Pesan

## 3. Activity Diagram Memproses Komplain



Gambar 3.23 Activity Diagram Memproses Komplain

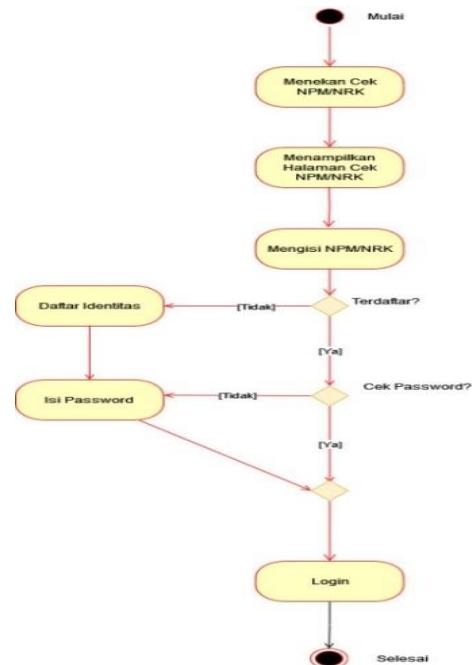
#### 4. Activity Diagram Menyelesaikan Komplain



Gambar 3.24 *Activity Diagram* Menyelesaikan Komplain

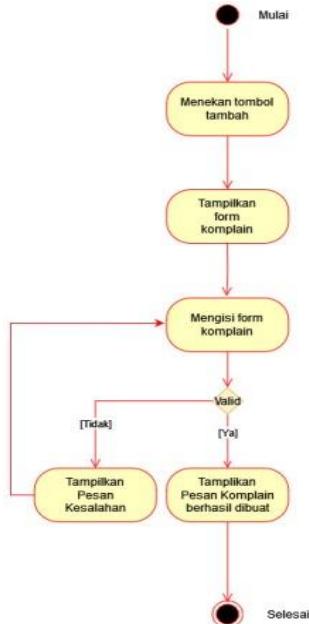
##### 3.4.1.4.3 Activity Diagram Pengguna

###### 1. Activity Diagram Melakukan Pendaftaran



Gambar 3.25 *Activity Diagram* Melakukan Pendaftaran

## 2. Activity Diagram Membuat Pengajuan Komplain



Gambar 3.26 Activity Diagram Membuat Pengajuan Komplain

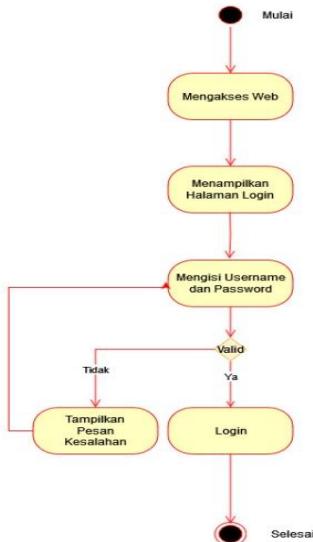
## 3. Activity Diagram Membalas Pesan komplain



Gambar 3.27 Activity Diagram Membalas Pesan Komplain

### 3.4.1.4.4 Activity Diagram Semua Pengguna

#### 1. Activity Diagram login



Gambar 3.28 Activity Diagram login

#### 2. Activity Diagram Mengubah Password



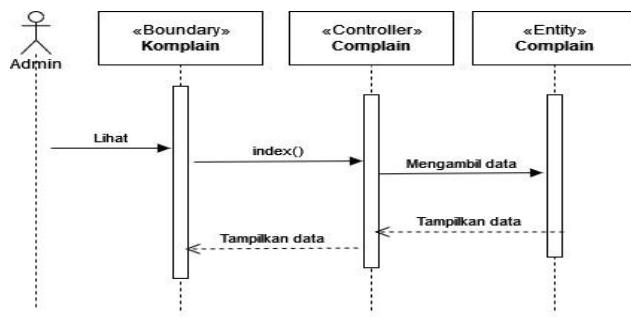
Gambar 3.29 Activity Diagram Mengubah Password

### 3.4.1.5 Sequence Diagram

*Sequence diagram* digunakan untuk memodelkan sistem yang dirancang dan untuk mengetahui interaksi dengan pengguna. Berikut akan dijelaskan gambaran tahap demi tahap aktifitas yang dilakukan sesuai dengan *use case* diagram sebelumnya yang telah dibuat.

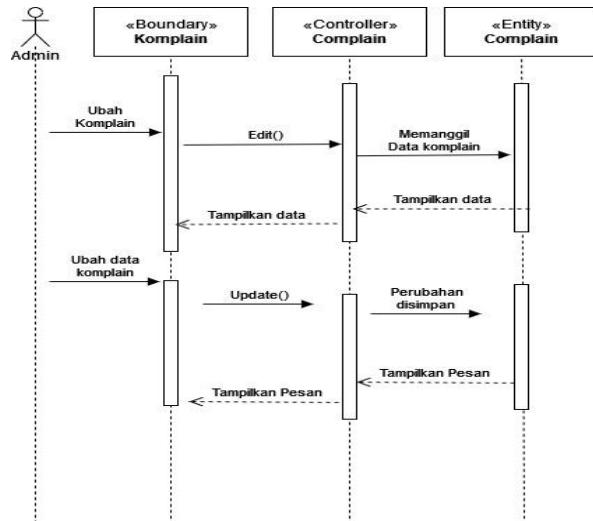
#### 3.4.1.5.1 Sequence Diagram Admin

##### 1. Sequence Diagram Menerima Pengajuan Komplain



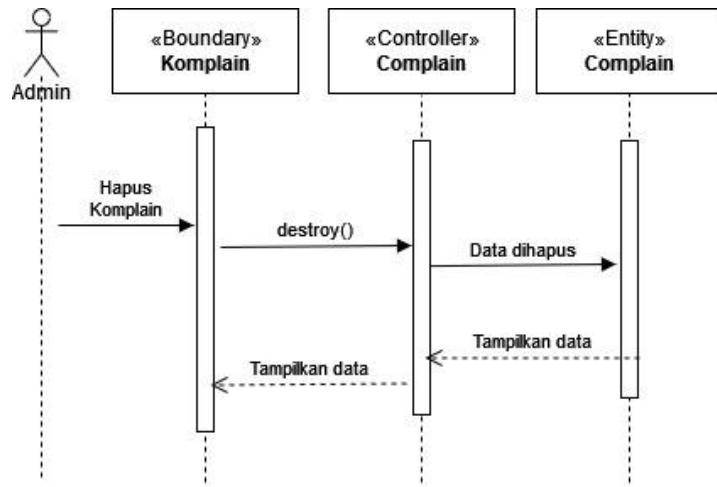
Gambar 3.30 Sequence Diagram Menerima Pengajuan Komplain

##### 2. Sequence Diagram Mengubah Komplain



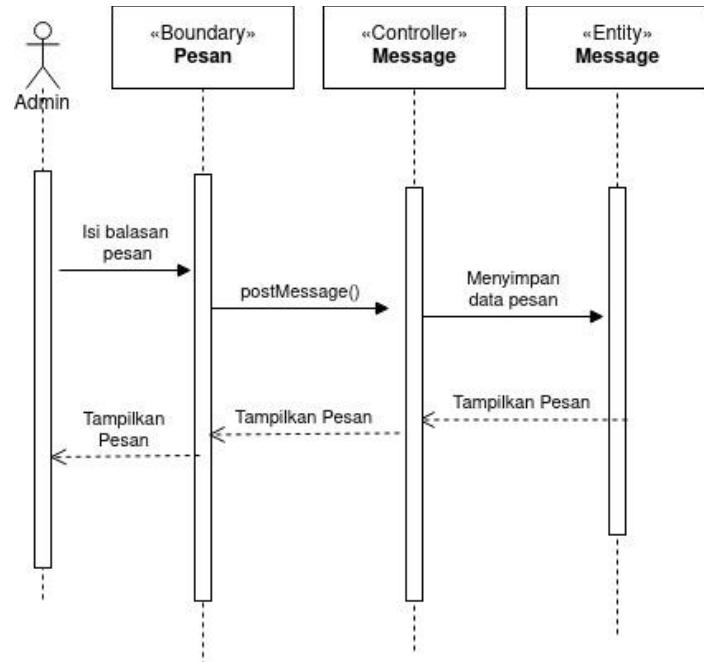
Gambar 3.31 Sequence Diagram Mengubah Komplain

### 3. Sequence Diagram Menghapus Komplain



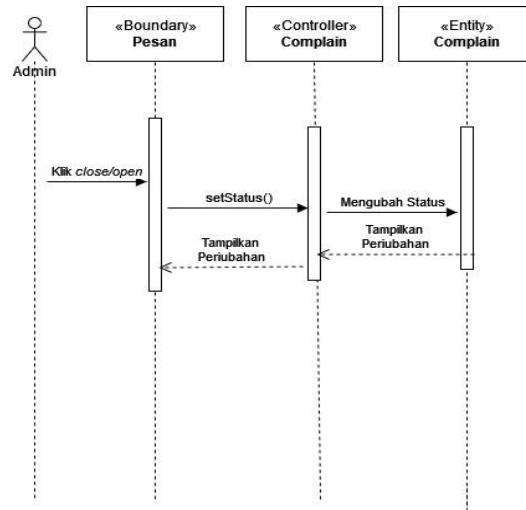
Gambar 3.32 Sequence Diagram Menghapus Komplain

### 4. Sequence Diagram Merespon Pesan Komplain



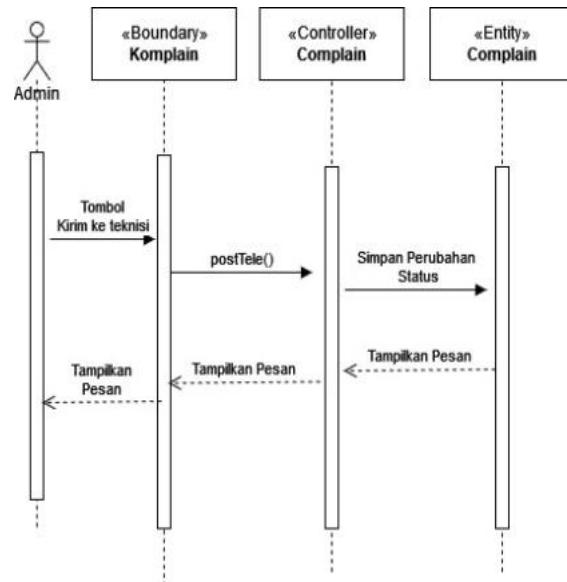
Gambar 3.33 Sequence Diagram Merespon Pesan Komplain

### 5. Sequence Diagram Mengubah Status Komplain



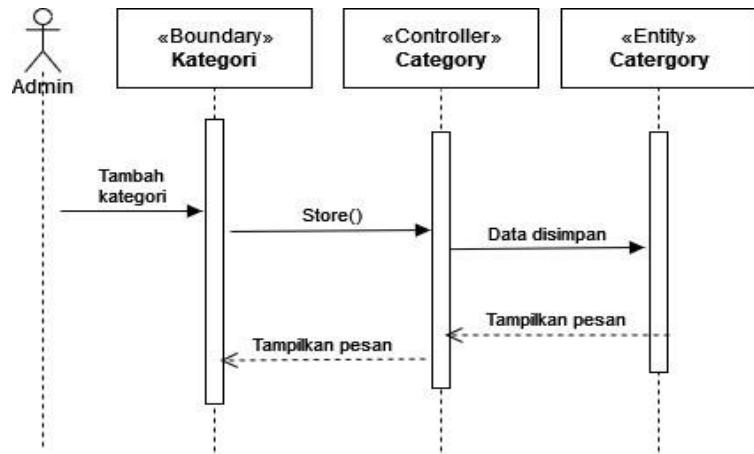
Gambar 3.34 Sequence Diagram Mengubah Status Komplain

### 6. Sequence Diagram Mengirim Komplain ke Teknisi



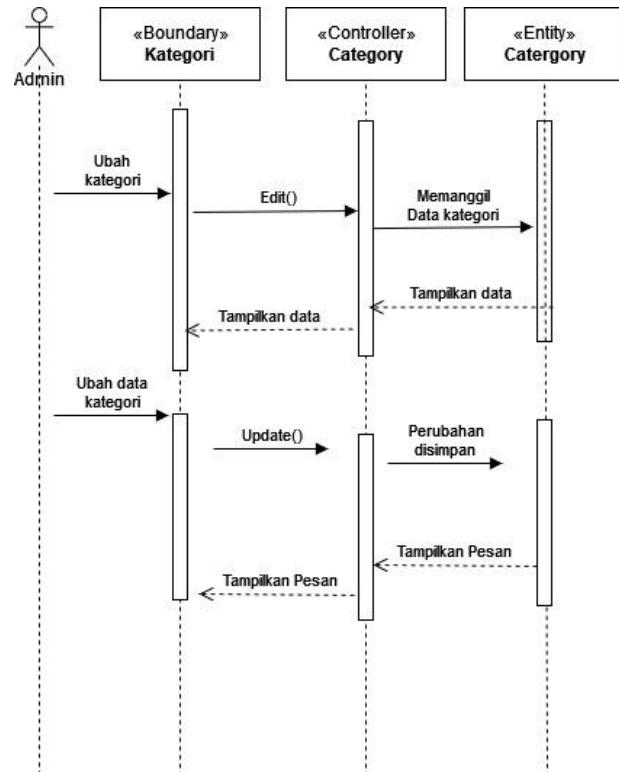
Gambar 3.35 Sequence Diagram Mengirim Komplain ke Teknisi

### 7. Sequence Diagram Membuat Kategori



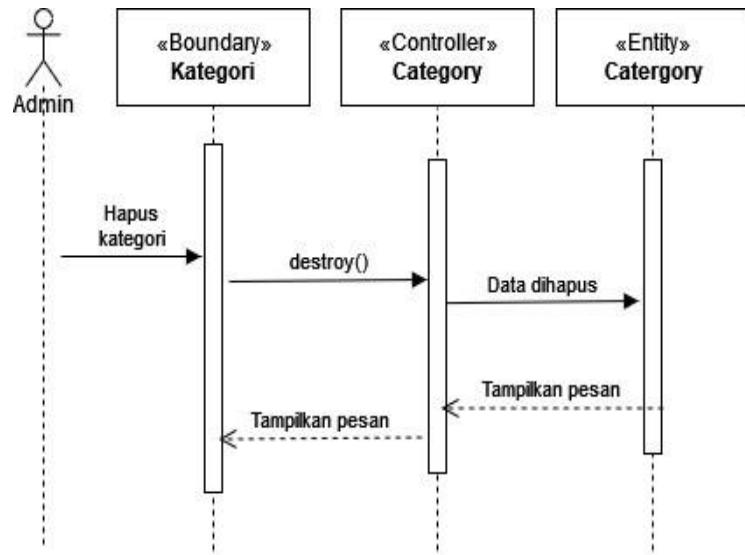
Gambar 3.36 Sequence Diagram Membuat Kategori

### 8. Sequence Diagram Mengubah Kategori



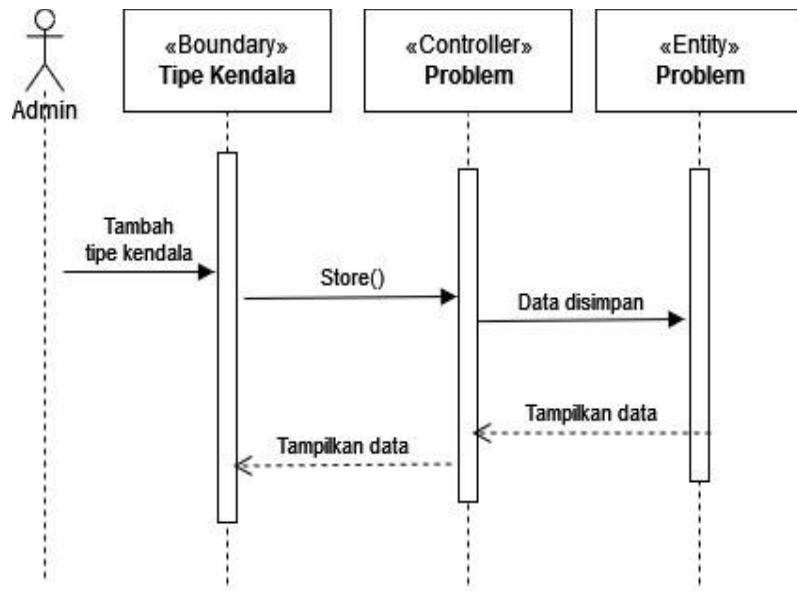
Gambar 3.37 Sequence Diagram Mengubah Kategori

9. *Sequence Diagram* Menghapus Kategori



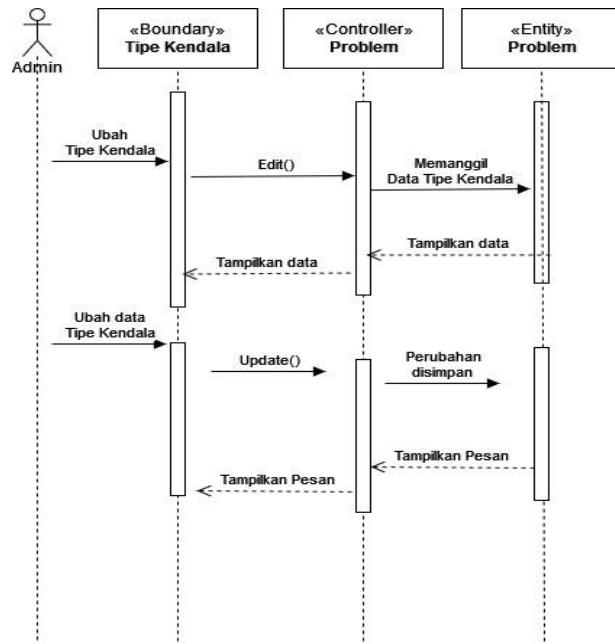
Gambar 3.38 *Sequence Diagram* Menghapus Kategori

10. *Sequence Diagram* Membuat Tipe Kendala



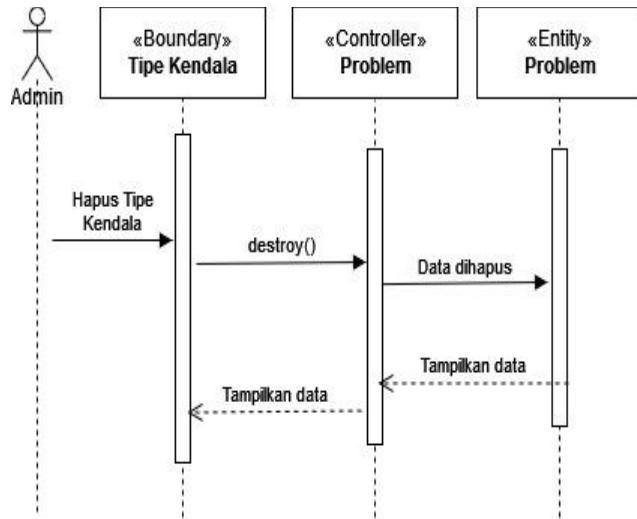
Gambar 3.39 *Sequence Diagram* Tipe Kendala

### 11. Sequence Diagram Mengubah Tipe Kendala



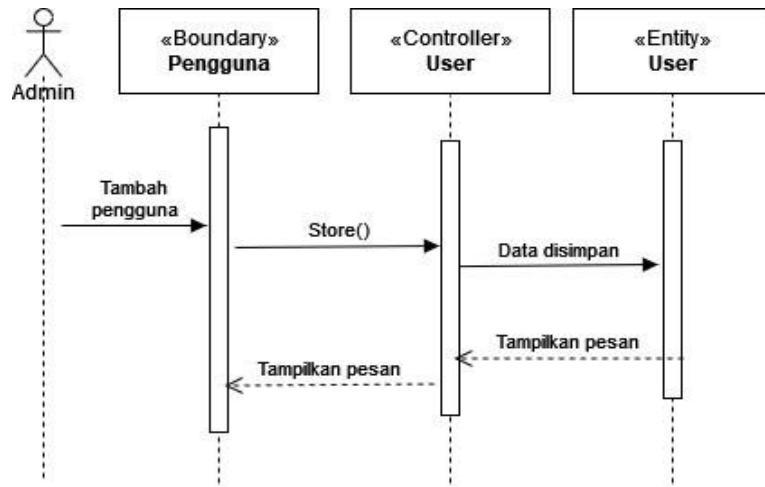
Gambar 3.40 Sequence Diagram Mengubah Tipe Kendala

### 12. Sequence Diagram Menghapus Tipe Kendala



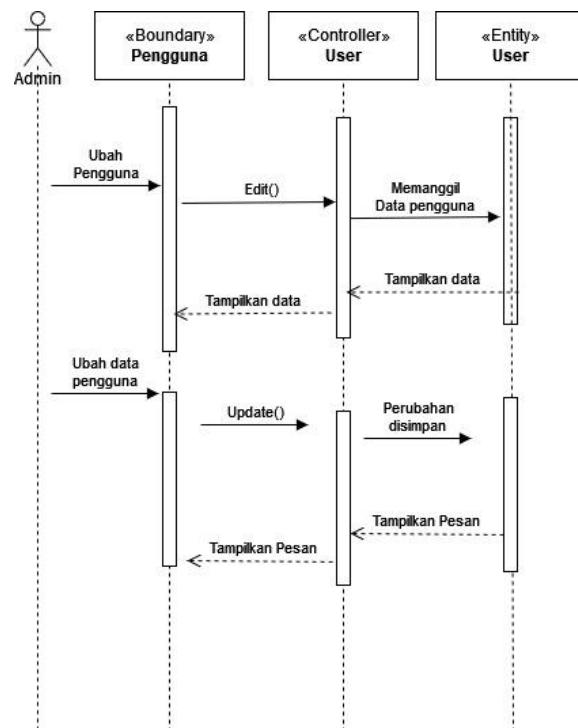
Gambar 3.41 Sequence Diagram Menghapus Tipe Kendala

13. *Sequence Diagram* Membuat Pengguna



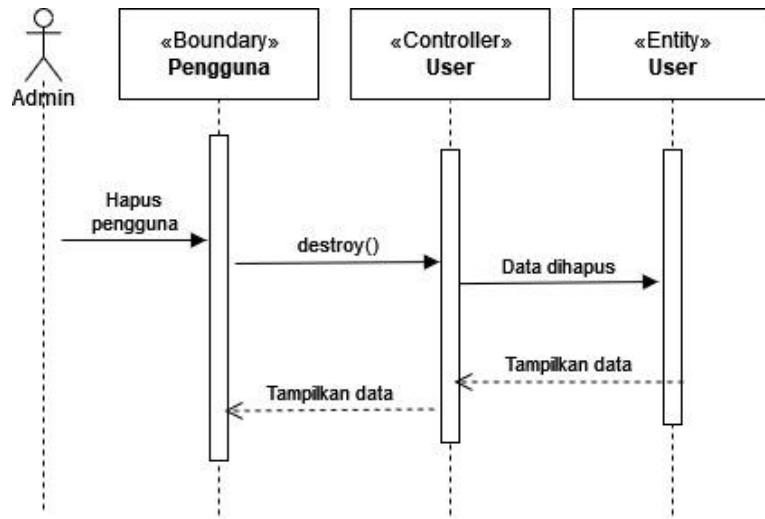
Gambar 3.42 *Sequence Diagram* Kelola Pengguna

14. *Sequence Diagram* Mengubah Pengguna



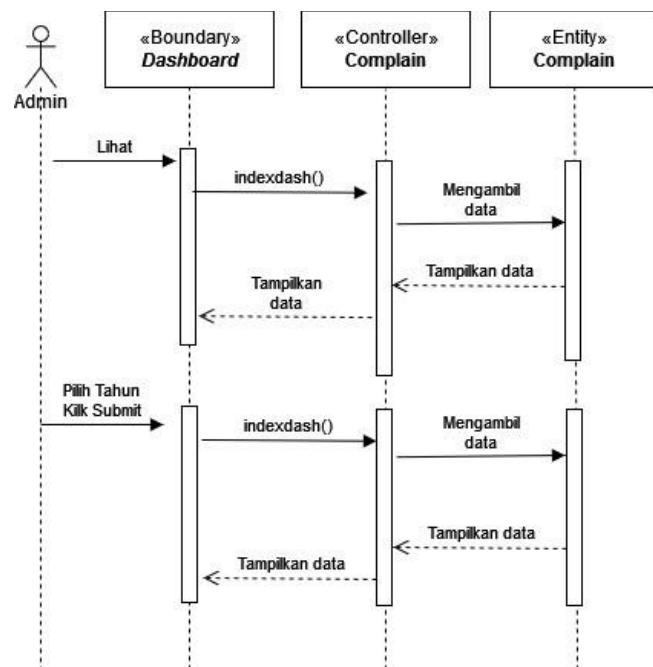
Gambar 3.43 *Sequence Diagram* Mengubah Pengguna

15. *Sequence Diagram Menghapus Pengguna*



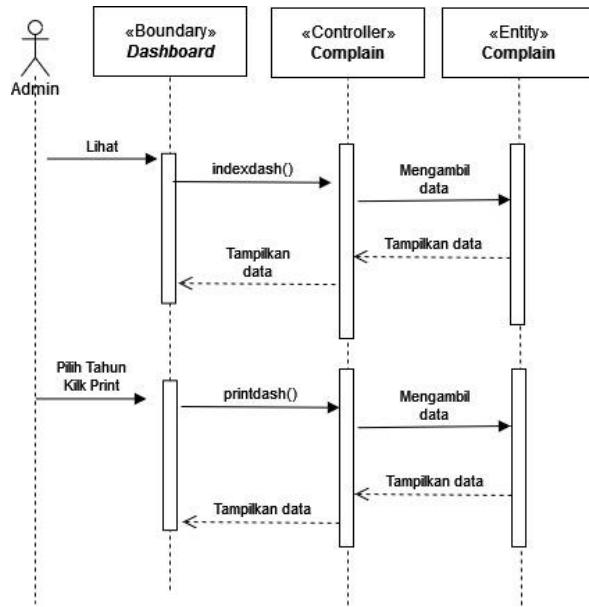
Gambar 3.44 *Sequence Diagram Menghapus Pengguna*

16. *Sequence Diagram Melihat Laporan*



Gambar 3.45 *Sequence Diagram Melihat Laporan*

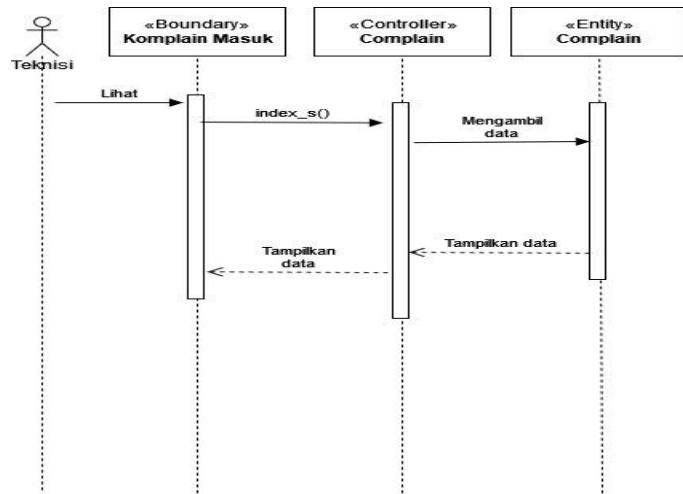
### 16. Sequence Diagram Mencetak Laporan



Gambar 3.46 Sequence Diagram Mencetak Laporan

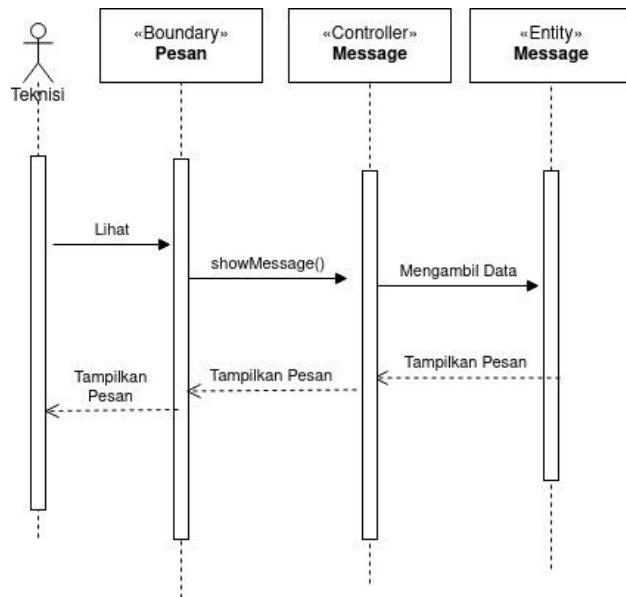
#### 3.4.1.5.2 Sequence Diagram Teknisi

##### 1. Sequence Diagram Menerima Kiriman Komplain



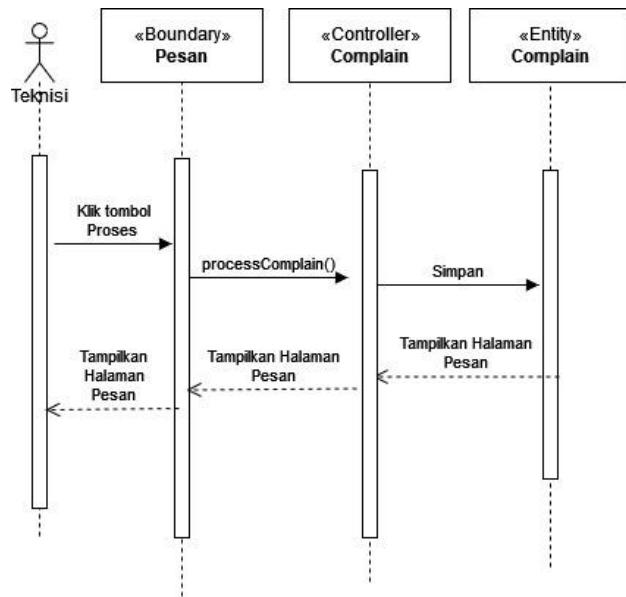
Gambar 3.47 Sequence Diagram Menerima Kiriman Komplain

## 2. Sequence Diagram Melihat Pesan



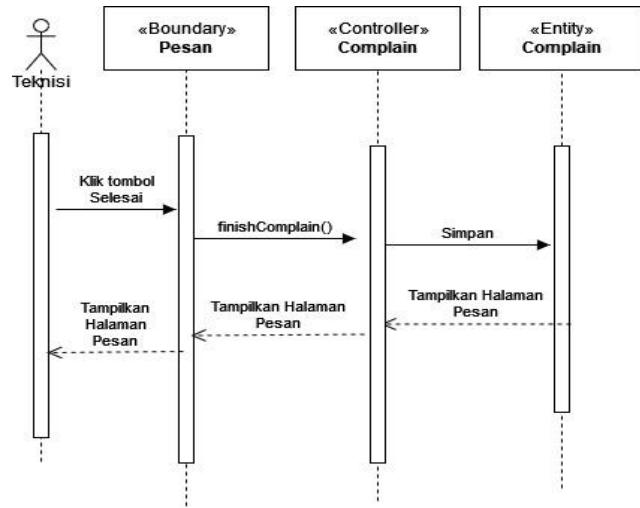
Gambar 3.48 Sequence Diagram Melihat Pesan

## 3. Sequence Diagram Memproses Komplain



Gambar 3.49 Sequence Diagram Memproses Komplain

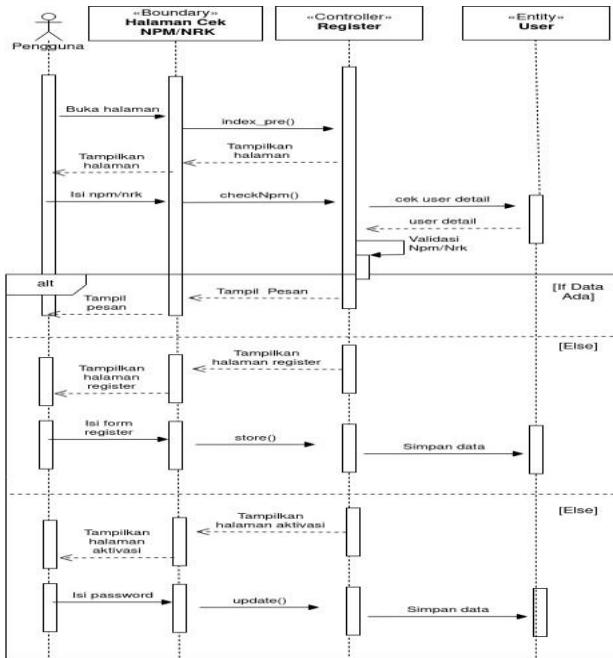
#### 4. Sequence Diagram Menyelesaikan Komplain



Gambar 3.50 *Sequence Diagram* Menyelesaikan Komplain

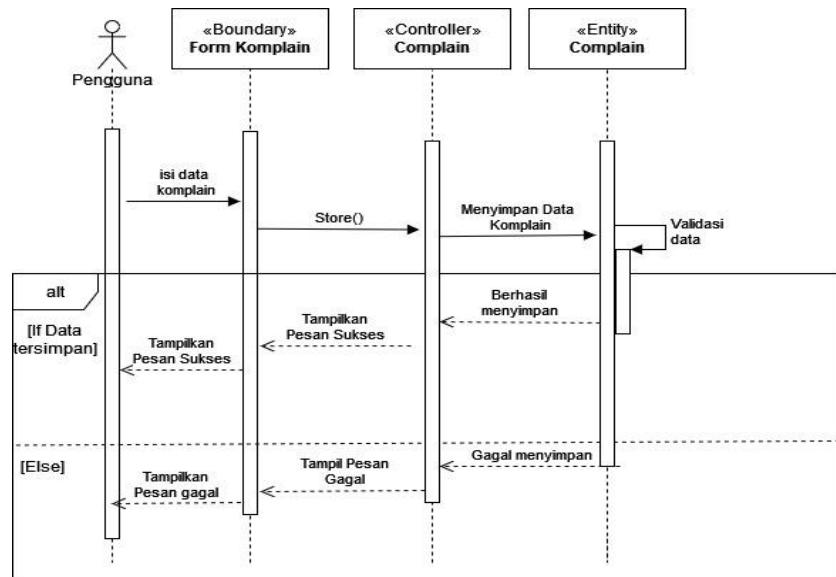
### **3.4.1.5.3 Sequence Diagram Pengguna**

### 1. Sequence Diagram Melakukan Pendaftaran



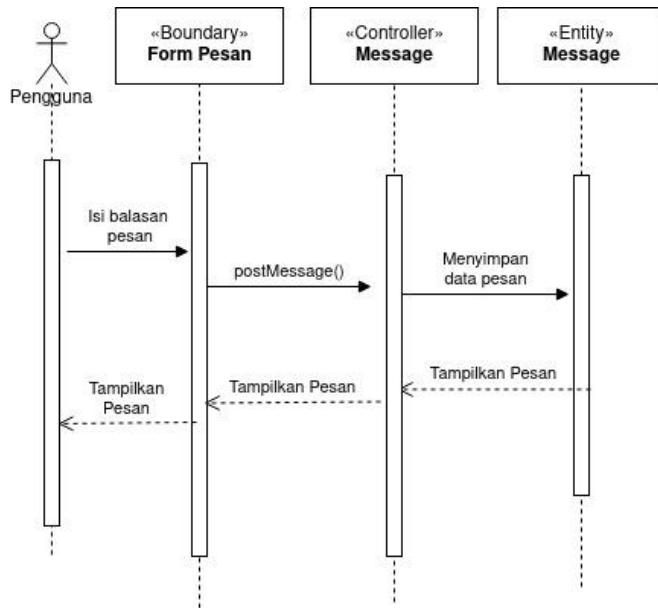
Gambar 3.51 *Sequence Diagram* Melakukan Pendaftaran

## 2. Sequence Diagram Membuat Pengajuan Komplain



Gambar 3.52 Sequence Diagram Membuat Pengajuan Komplain

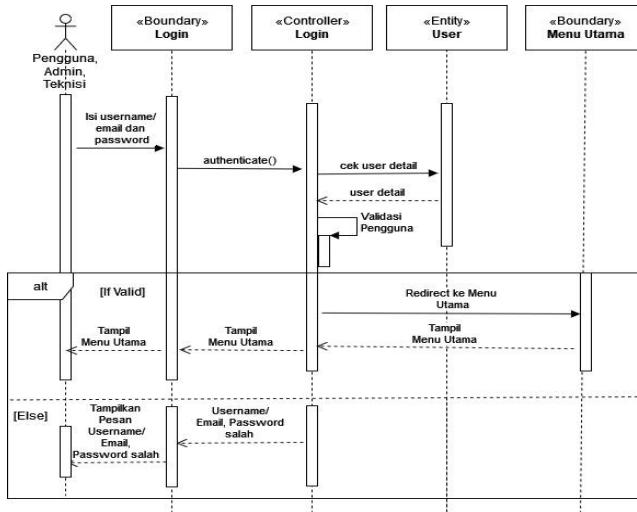
## 3. Sequence Diagram Membalas Pesan Komplain



Gambar 3.53 Sequence Diagram Membalas Pesan Komplain

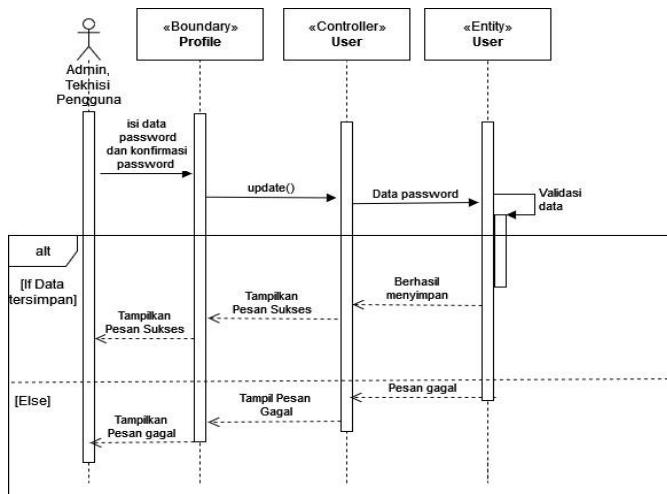
### 3.4.1.5.4 Sequence Diagram Semua Pengguna

#### 1. Sequence Diagram login



Gambar 3.54 Sequence Diagram login

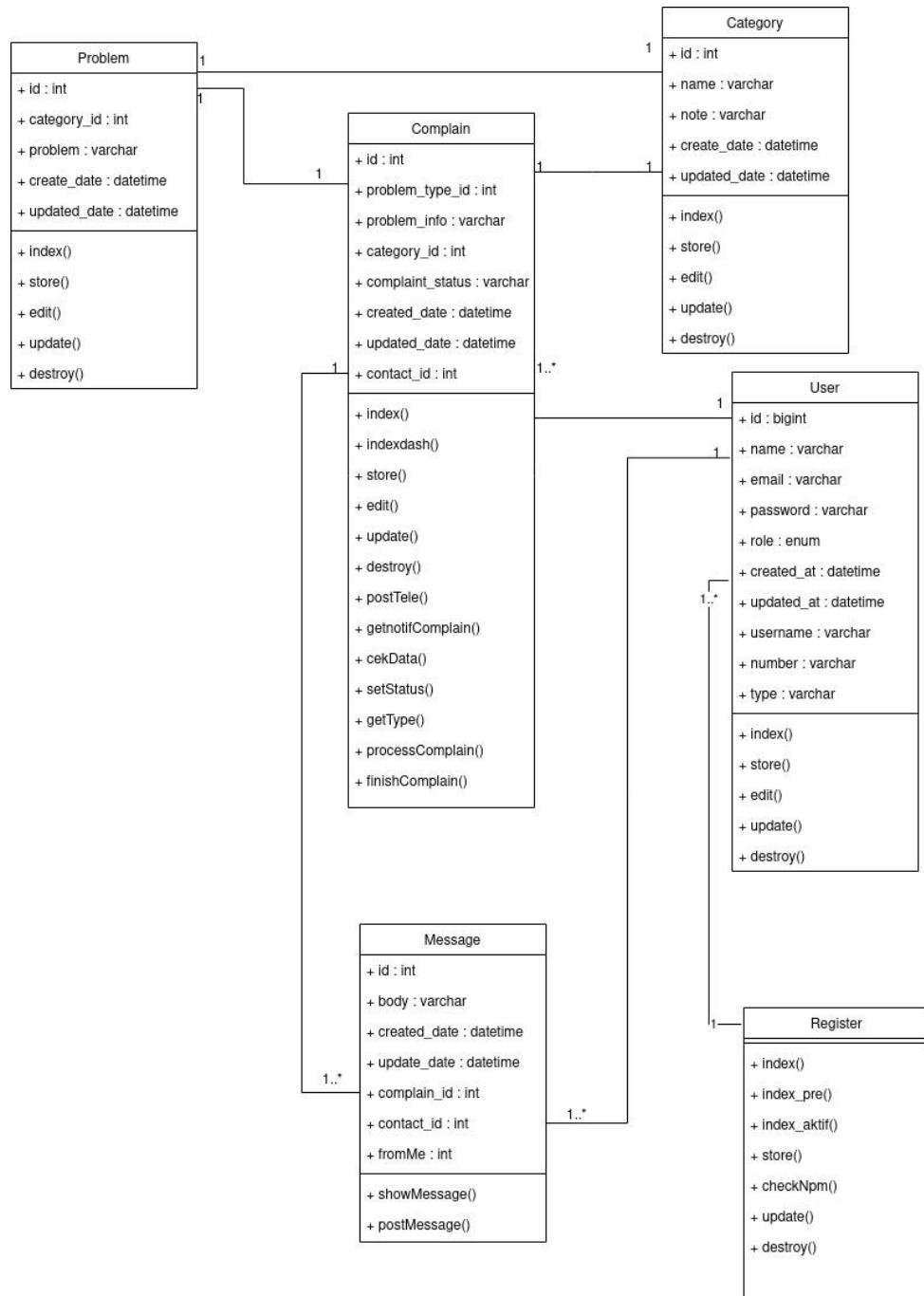
#### 2. Sequence Diagram Mengubah Password



Gambar 3.55 Sequence Diagram Mengubah Password

### 3.4.1.6 Class Diagram

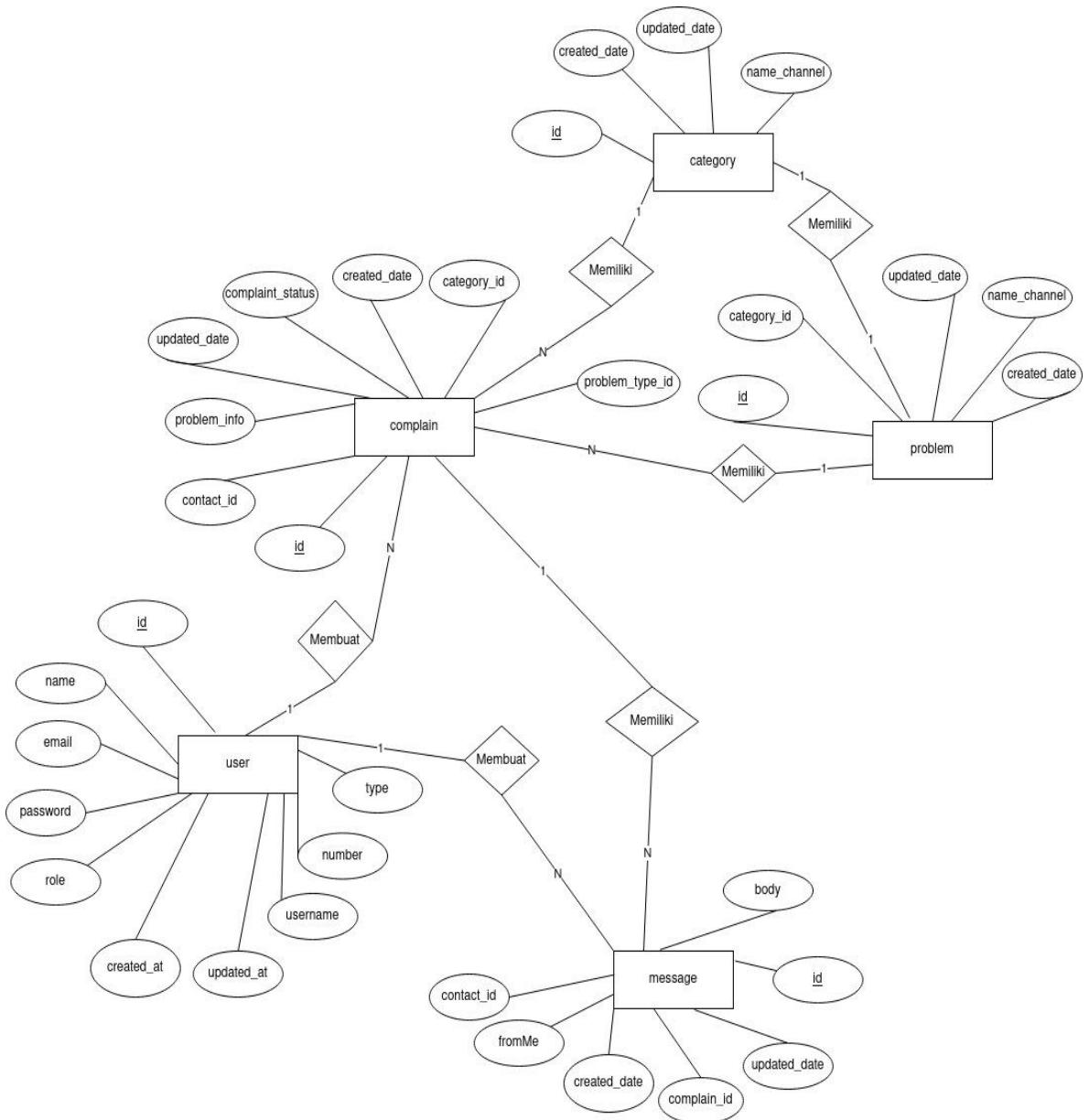
Berikut adalah *class diagram* untuk *helpdesk* yang akan digambarkan dibawah ini.



Gambar 3.56 *Class Diagram*

### 3.4.1.7 Entity Relationship Diagram

Berikut adalah *Entity Relationship Diagram* untuk *helpdesk* yang akan digambarkan dibawah ini:



Gambar 3.57 *Entity Relationship Diagram*

### 3.4.2 Perancangan Antarmuka

#### A. Rancangan tampilan Admin

##### 1. Rancangan Menerima Pengajuan Komplain

The screenshot shows a web-based application interface for managing complaints. On the left, there is a vertical sidebar with a logo at the top, followed by navigation links: Dashboard, Komplain (highlighted in red), Master Data, Kategori, Tipe Kendala, Pengguna, and Profile. The main content area is titled 'Komplain' and contains a table of complaints. The table has columns: No, Pelapor, Nama Pelapor, NPMNPK, Melapor Melalui, Tipe Kendala, Kategori, Masalah, Status, Create, and Action. Two rows of data are shown:

No	Pelapor	Nama Pelapor	NPMNPK	Melapor Melalui	Tipe Kendala	Kategori	Masalah	Status	Create	Action
1	Mahasiswa	Adam Uni	0219103118	1	2022-12-01 15:39:32					<button>Pesan</button> <button>Sampah</button> <button>Kund</button> <button>Kirim Ke Teknis</button>
2	Mahasiswa	Alim	0218201022	11	2023-11-1 14:56:35					<button>Pesan</button> <button>Sampah</button> <button>Kund</button> <button>Kirim Ke Teknis</button>

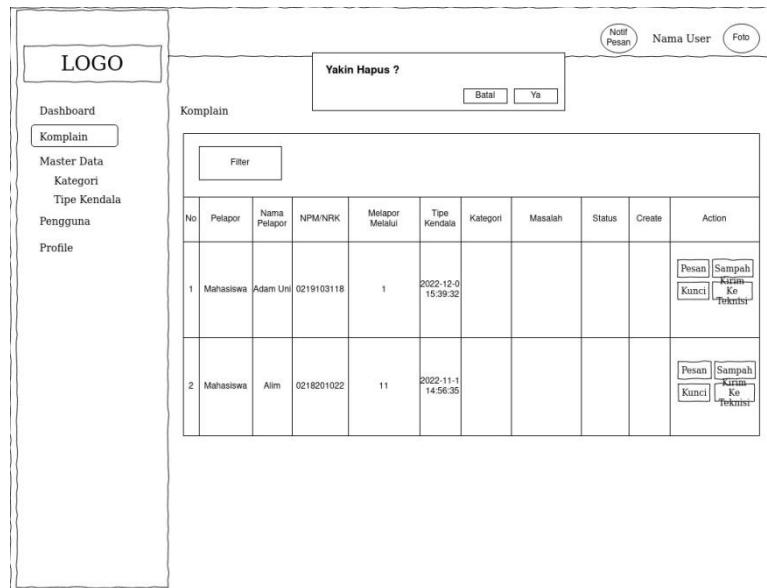
Gambar 3.58 Rancangan Menerima Pengajuan Komplain

##### 2. Rancangan Mengubah Komplain

The screenshot shows a modal dialog box titled 'Edit Data'. It contains three dropdown menus: 'Kategori', 'Tipe Kendala', and 'Kendala'. At the bottom of the dialog are two buttons: 'Close' and 'Simpan'.

Gambar 3.59 Rancangan Mengubah Komplain

### 3. Rancangan Menghapus Komplain



Gambar 3.60 Rancangan Menghapus Komplain

### 2. Rancangan Merespon Pesan



Gambar 3.61 Rancangan Merespon Pesan

### 3. Rancangan Mengubah Status



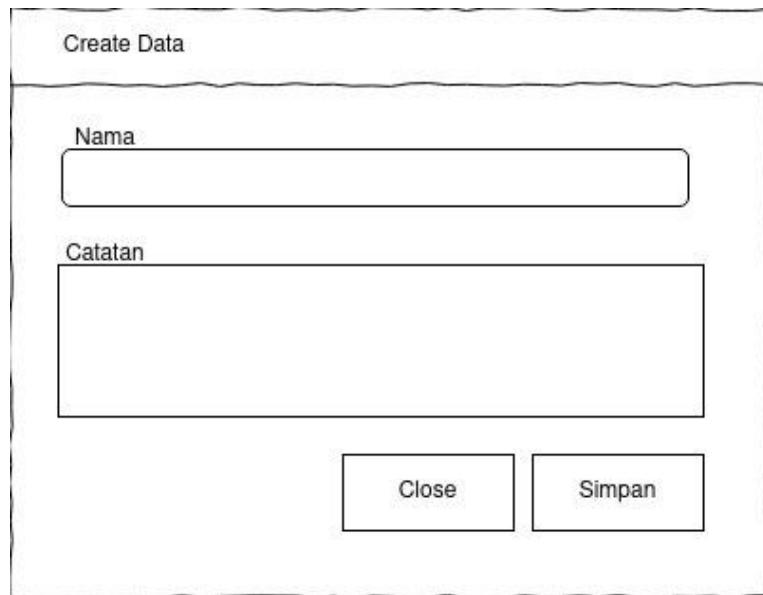
Gambar 3.62 Rancangan Mengubah Status

### 4. Rancangan Mengirim Komplain Ke Teknisi

No	Pelapor	Nama Pelapor	NPM/NRK	Melapor Melalui	Tipe Kendala	Kategori	Masalah	Status	Create	Action
1	Mahasiswa	Adam Uni	0219103118	1	2022-12-0 15:39:32					<input type="button" value="Pesan Sampah"/> <input type="button" value="Kirim Ke Teknis"/>
2	Mahasiswa	Alim	0218201022	11	2022-11-1 14:56:35					<input type="button" value="Pesan Sampah"/> <input type="button" value="Kirim Ke Teknis"/>

Gambar 3.63 Rancangan Mengirim Komplain Ke Teknisi

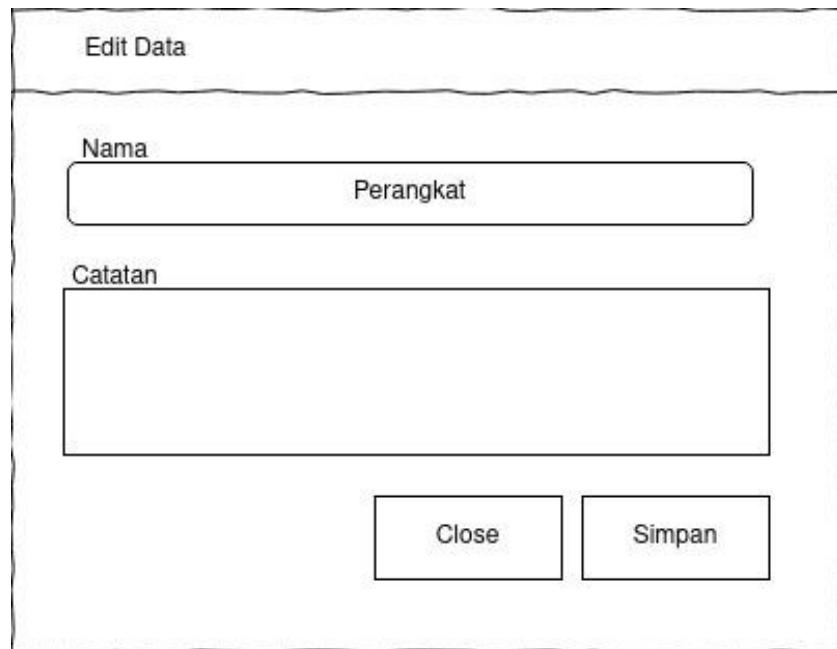
### 5. Rancangan Membuat Kategori



A wireframe diagram of a 'Create Data' form. At the top, it says 'Create Data'. Below that is a 'Nama' field with an input box. Underneath is a 'Catatan' field with a larger input box. At the bottom right are two buttons: 'Close' and 'Simpan'.

Gambar 3.64 Rancangan Membuat Kategori

### 6. Rancangan Mengubah Kategori



A wireframe diagram of an 'Edit Data' form. At the top, it says 'Edit Data'. Below that is a 'Nama' field containing the value 'Perangkat' with an input box. Underneath is a 'Catatan' field with a large input box. At the bottom right are two buttons: 'Close' and 'Simpan'.

Gambar 3.65 Rancangan Mengubah Kategori

## 7. Rancangan Menghapus Kategori

The diagram illustrates a user interface for deleting a category. It features a sidebar on the left with a logo and navigation links: Dashboard, Komplain, Master Data, Kategori (highlighted), Tipe Kendala, Pengguna, and Profile. The main content area has a header with 'Yakin Hapus?' (Are you sure?) and 'Batal' (Cancel) / 'Ya' (Yes) buttons. Below this is a 'Kategori' section with a 'Tambah Data' button and a table listing two categories:

No	Nama	Catatan	Create	Action
1	PERANGKAT	Untuk Kendala Perangkat Komputer/Komputer, Infokus, dll	2022-12-06 15:39:32	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>
2	INTERNET	Untuk kendala internet & wifi	2022-11-14 14:56:35	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>

At the top right of the main area are three circular icons: Notif Pesan, Nama User, and Foto.

Gambar 3.66 Rancangan Menghapus Kategori

## 8. Rancangan Membuat Tipe Kendala

The diagram shows a form for creating a new category. The title is 'Create Data'. It contains two input fields: 'Kategori' and 'Tipe Kendala', each with a downward arrow icon indicating a dropdown menu. At the bottom are two buttons: 'Close' and 'Simpan' (Save).

Gambar 3.67 Rancangan Membuat Tipe Kendala

## 9. Rancangan Mengubah Tipe Kendala

The diagram shows a 'Create Data' form for changing obstacle types. It includes fields for 'Kategori' (Category) and 'Tipe Kendala' (Type of Obstacle), both with dropdown menus. At the bottom are 'Close' and 'Simpan' (Save) buttons.

Gambar 3.68 Rancangan Mengubah Tipe Kendala

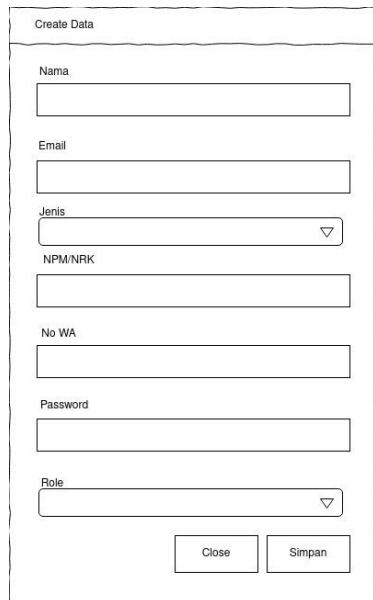
## 10. Rancangan Menghapus Tipe Kendala

The diagram shows a dashboard with a sidebar containing 'Dashboard', 'Komplain', 'Master Data', 'Kategori' (selected), 'Tipe Kendala', 'Pengguna', and 'Profile'. The main area shows a confirmation dialog 'Yakin Hapus?' (Are you sure?) with 'Batal' (Cancel) and 'Ya' (Yes) buttons. Below it is a table titled 'Tambah Data' (Add Data) with columns: No, Kategori, Masalah, Create, and Action. Two rows of data are shown:

No	Kategori	Masalah	Create	Action
1	PERANGKAT	Kerusakan	2022-12-06 15:39:32	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>
2	SIAKAD	Login	2022-11-14 14:56:35	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>

Gambar 3.69 Rancangan Menghapus Tipe Kendala

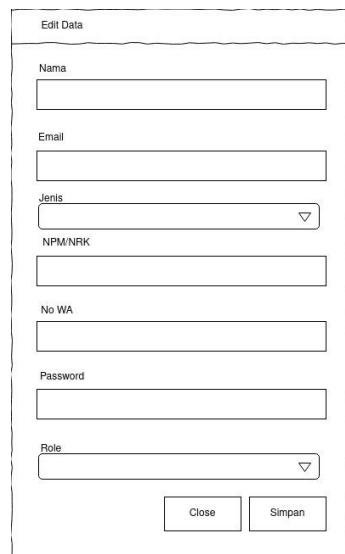
## 11. Rancangan Membuat Pengguna



The diagram shows a user interface for creating a new user. It features a title bar labeled 'Create Data'. Below the title are seven input fields: 'Nama' (Name), 'Email', 'Jenis' (Gender) with a dropdown arrow, 'NPM/NRK', 'No WA' (WhatsApp Number), 'Password', and 'Role' with a dropdown arrow. At the bottom right are two buttons: 'Close' and 'Simpan' (Save).

Gambar 3.70 Rancangan Membuat Pengguna

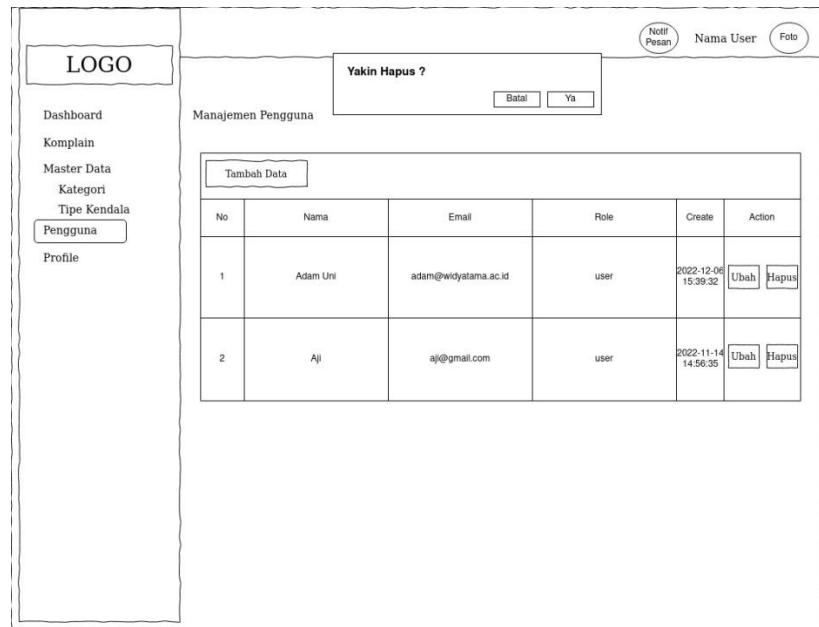
## 12. Rancangan Mengubah Pengguna



The diagram shows a user interface for editing an existing user. It features a title bar labeled 'Edit Data'. Below the title are seven input fields: 'Nama' (Name), 'Email', 'Jenis' (Gender) with a dropdown arrow, 'NPM/NRK', 'No WA' (WhatsApp Number), 'Password', and 'Role' with a dropdown arrow. At the bottom right are two buttons: 'Close' and 'Simpan' (Save).

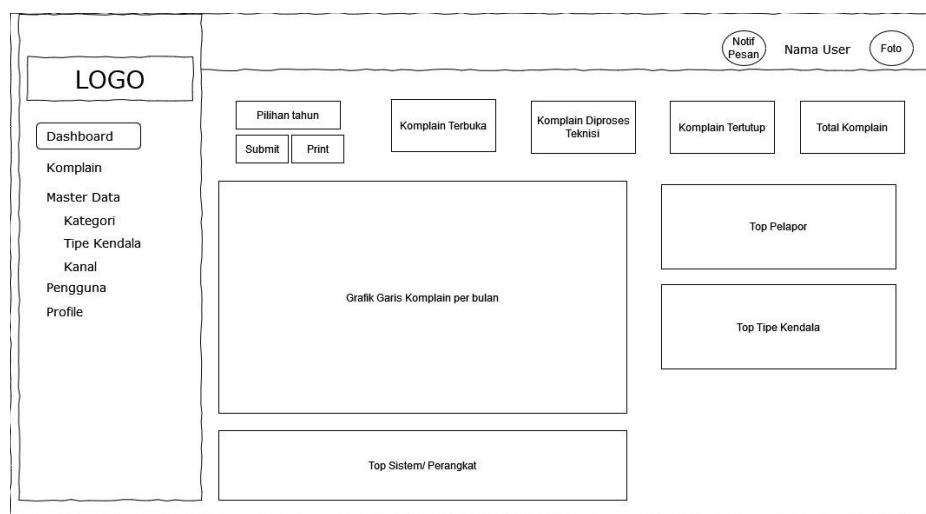
Gambar 3.71 Rancangan Mengubah Pengguna

### 13. Rancangan Menghapus Pengguna



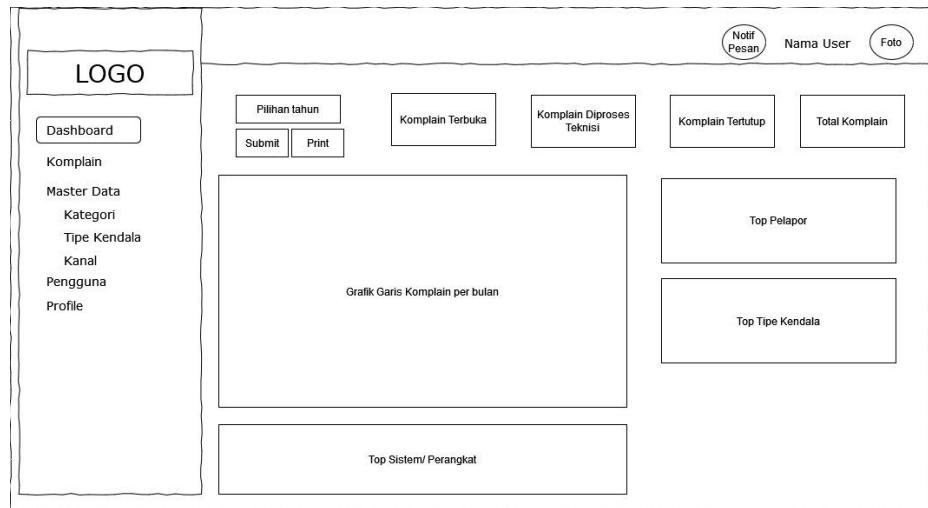
Gambar 3.72 Rancangan Menghapus Pengguna

### 14. Rancangan Melihat Laporan



Gambar 3.73 Rancangan Melihat Laporan

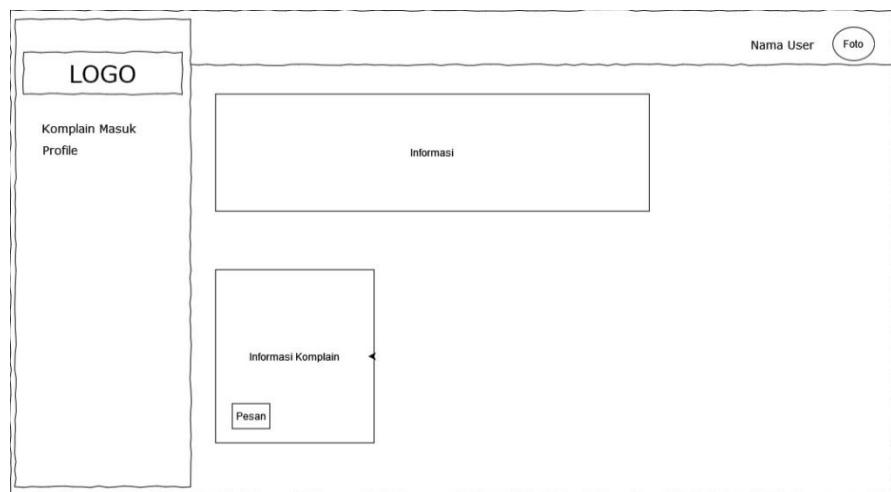
## 15. Rancangan Mencetak Laporan



Gambar 3.74 Rancangan Mencetak Laporan

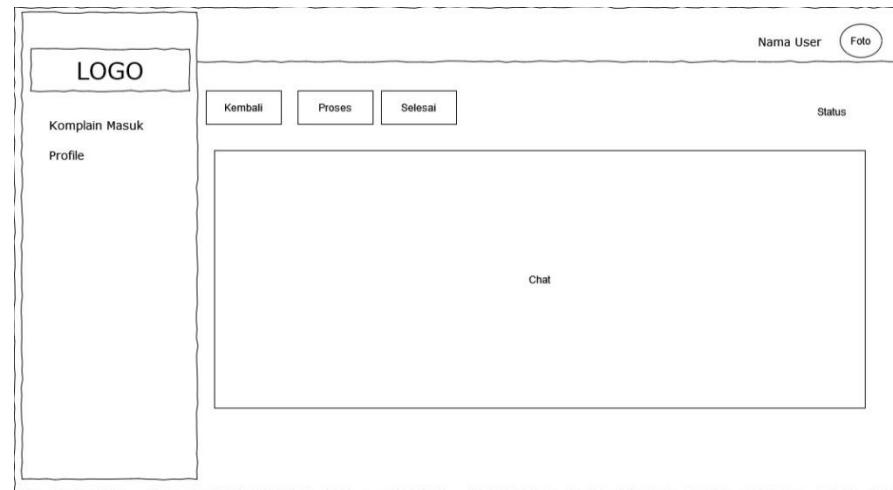
## B. Rancangan Tampilan Teknisi

### 1. Rancangan Menerima Kiriman Komplain



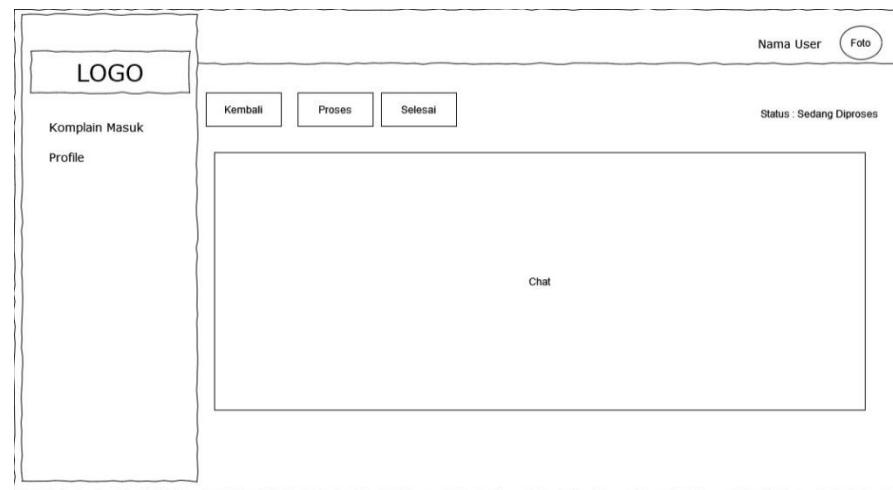
Gambar 3.75 Rancangan Menerima Kiriman Komplain

## 2. Rancangan Melihat Pesan



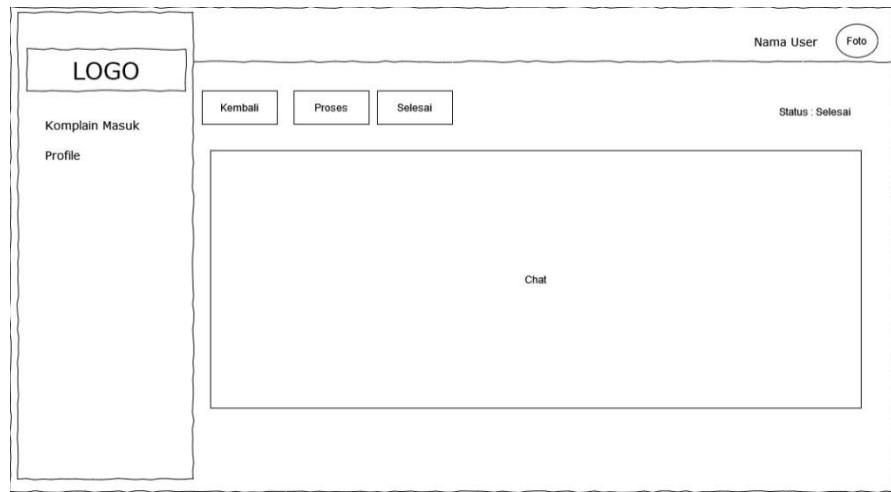
Gambar 3.76 Rancangan Melihat Pesan

## 3. Rancangan Memproses Komplain



Gambar 3.77 Rancangan Memproses Komplain

#### 4. Rancangan Menyelesaikan Komplain



Gambar 3.78 Rancangan Menyelesaikan Komplain

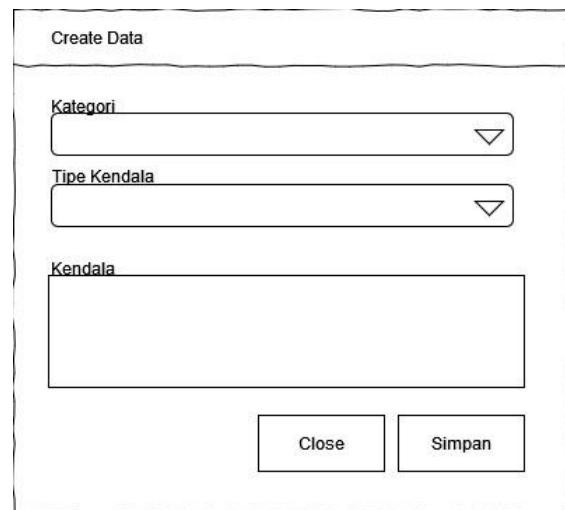
### C. Rancangan Tampilan Pengguna

#### 1. Rancangan Melakukan Pendaftaran

The wireframe shows a registration form titled 'Cek NPM'. It has a 'NPM/NRK' label above a text input field containing the placeholder 'Masukan Npm atau Email Widyatama'. Below the input field is a button labeled 'Cek'. At the bottom of the form is a link 'Sudah punya akun? Login!'

Gambar 3.79 Melakukan Pendaftaran

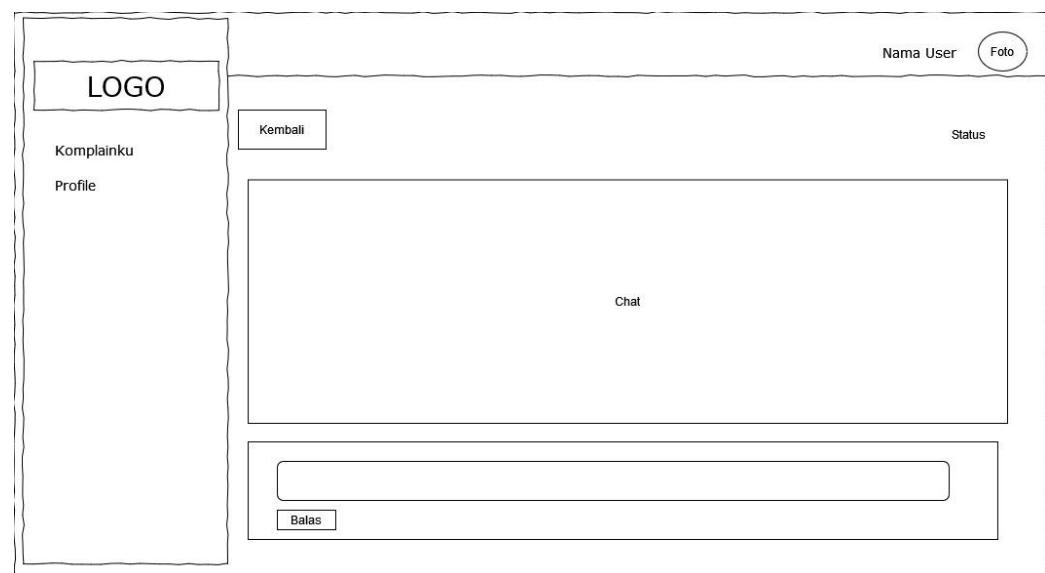
## 2. Rancangan Membuat Pengajuan Komplain



Form titled "Create Data" for creating a complaint application. It contains three dropdown menus labeled "Kategori", "Tipe Kendala", and "Kendala". Below the dropdowns are two buttons: "Close" and "Simpan".

Gambar 3.80 Rancangan Membuat Pengajuan Komplain

## 3. Rancangan Membalas Pesan



Form for replying to messages. The left sidebar includes a logo, "Komplainku", and "Profile" sections. The main area features a "Kembali" button, a large "Chat" window, and a message input field with a "Balas" button.

Gambar 3.81 Rancangan Membalas Pesan

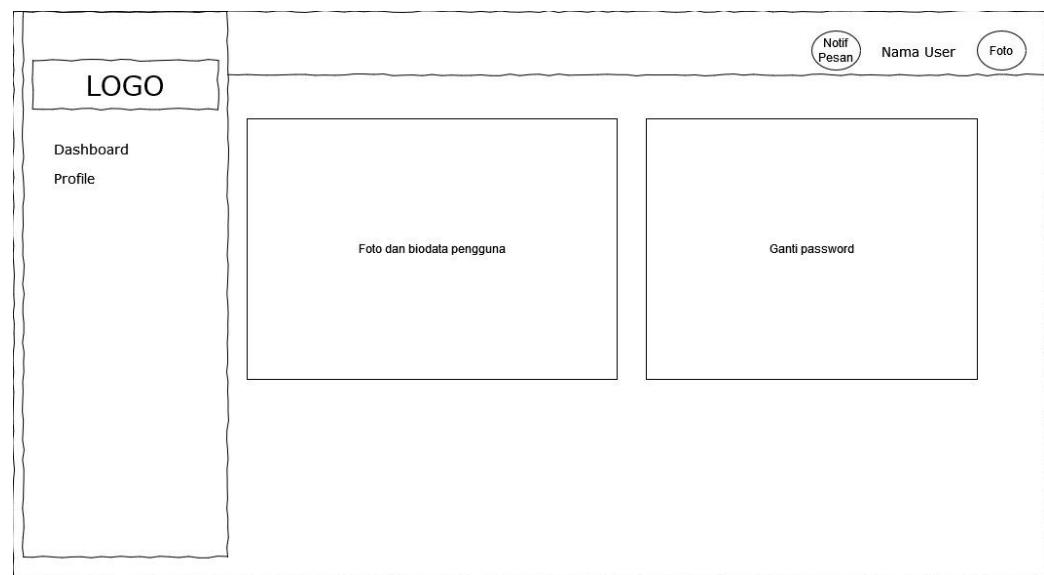
## D. Rancangan Tampilan Semua Pengguna

### 1. Rancangan *Login*



Gambar 3.82 Rancangan *Login*

### 2. Rancangan Mengubah *Password*



Gambar 3.83 Rancangan Mengubah *Password*

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Implementasi Dan Pengujian**

Kebutuhan akan hardware dan software merupakan daftar spesifikasi hardware minimum yang dapat digunakan serta software yang digunakan pada saat pembuatan program dan agar dapat mengimplementasikan perancangan yang telah dibuat sebelumnya, dibutuhkan sebuah hardware dan software yang dapat mendukung untuk setiap kali menjalankan aplikasi tersebut. Berikut ini merupakan analisa dari beberapa hardware dan software yang diperlukan:

##### **4.1.1 Server**

Spesifikasi minimum *server* yang harus dipenuhi agar aplikasi bisa berjalan untuk mengimplementasi aplikasi yang akan dibangun. Berikut adalah penjelasan spesifikasi minimum *hardware* dan *software* yang dibutuhkan oleh *server* agar aplikasi dapat berjalan.

Tabel 4.1 Spesifikasi Perangkat Keras Minimum

<b>Hardware</b>	<b>Spesifikasi</b>
<b>Processor</b>	1,6 GHz CPU
<b>RAM</b>	2 GB
<b>Harddisk</b>	1 GB
<b>Sistem Operasi</b>	Linux

Tabel 4.2 Spesifikasi Perangkat Lunak Minimum

<b>Software</b>	<b>Versi</b>
<b>Apache</b>	2.4. 46.
<b>PHP</b>	7.4
<b>MYSQL</b>	5.7

#### 4.1.2 *Client*

Tabel 4.3 Spesifikasi Minimum Client

<b>Software</b>	<b>Versi</b>
<b>Mozilla Firefox</b>	90. 46.
<b>Chrome</b>	90.40

## 4.2 Struktur Tabel

Berikut struktur tabel aplikasi pengelolaan layanan *helpdesk*:

### 1. Tabel *category*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 	int(11)			No	None		AUTO_INCREMENT
2	<b>name</b>	varchar(255)	latin1_swedish_ci		Yes	NULL		
3	<b>note</b>	varchar(255)	latin1_swedish_ci		Yes	NULL		
4	<b>created_date</b>	datetime			Yes	current_timestamp()		
5	<b>updated_date</b>	datetime			Yes	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.1 Tabel *category*

## 2. Tabel *Complain*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 🔑	int(11)			No	None		AUTO_INCREMENT
2	<b>problem_type_id</b> 💡	int(11)			No	0		
3	<b>problem_info</b>	varchar(255)	latin1_swedish_ci		Yes	NULL		
4	<b>contact_id</b> 💬	int(20)		UNSIGNED	Yes	NULL		
5	<b>category_id</b> 💡	int(11)			No	0		
6	<b>complaint_status</b>	varchar(7)	latin1_swedish_ci		No			
7	<b>created_date</b>	datetime			Yes	current_timestamp()		
8	<b>updated_date</b>	datetime			Yes	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Gambar 4.2 Tabel *Complain*

## 3. Tabel *message*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 🔑	int(11)			No	None		AUTO_INCREMENT
2	<b>body</b>	longtext	latin1_swedish_ci		Yes	NULL		
3	<b>created_date</b>	datetime			Yes	current_timestamp()		
4	<b>update_date</b>	datetime			Yes	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
5	<b>complain_id</b> 💡	int(11)			Yes	NULL		
6	<b>contact_id</b> 💬	int(20)		UNSIGNED	Yes	NULL		
7	<b>fromMe</b>	int(1)			Yes	NULL		

Gambar 4.3 Tabel *message*

## 4. Tabel *problem*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 💡	int(11)			No	None		AUTO_INCREMENT
2	<b>category_id</b> 💡	int(11)			No	None		
3	<b>problem</b>	varchar(255)	latin1_swedish_ci		Yes	NULL		
4	<b>created_date</b>	datetime			Yes	current_timestamp()		
5	<b>updated_date</b>	datetime			Yes	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

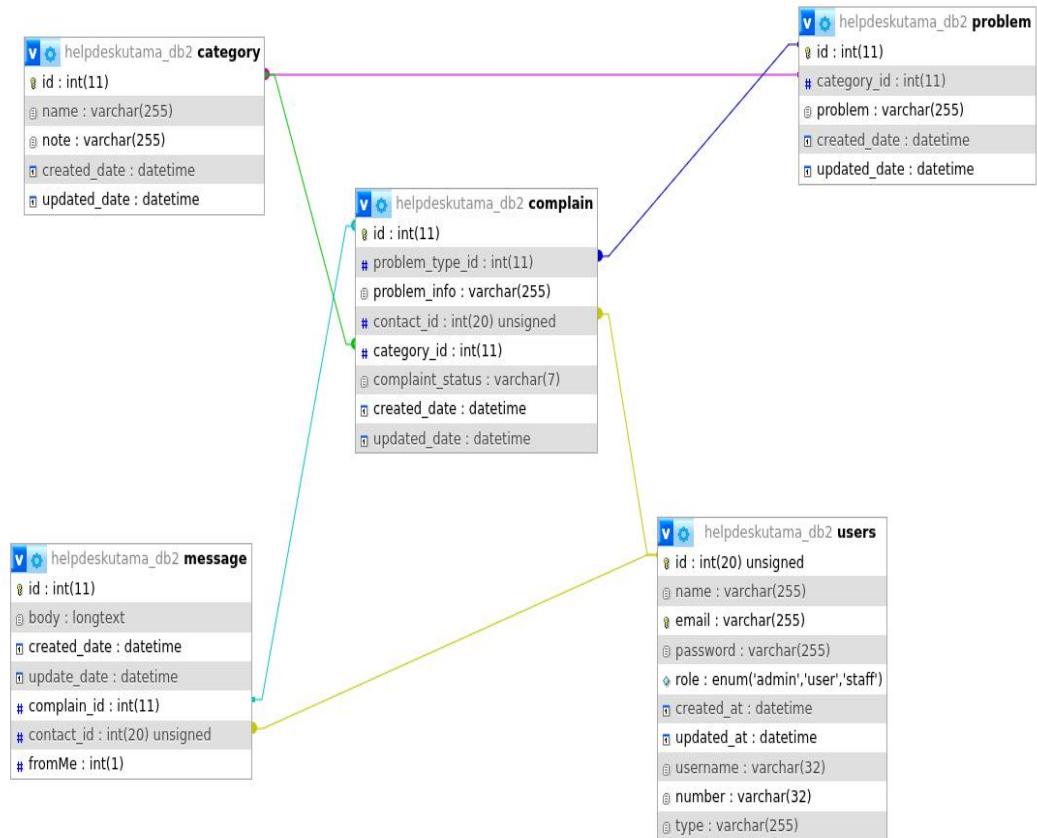
Gambar 4.4 Tabel *problem*

## 5. Tabel *users*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 🤖	int(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>name</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
3	<b>email</b> 📩	varchar(255)	utf8mb4_unicode_ci		No	None		
4	<b>password</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
5	<b>role</b>	enum('admin', 'user', 'teknisi')	utf8mb4_unicode_ci		Yes	NULL		
6	<b>created_at</b>	datetime			Yes	current_timestamp()		
7	<b>updated_at</b>	datetime			Yes	current_timestamp()		
8	<b>username</b>	varchar(32)	utf8mb4_unicode_ci		Yes	NULL		
9	<b>number</b>	varchar(32)	utf8mb4_unicode_ci		Yes	NULL		
10	<b>type</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		

Gambar 4.5 Tabel *users*

## 4.3 Relasi Tabel



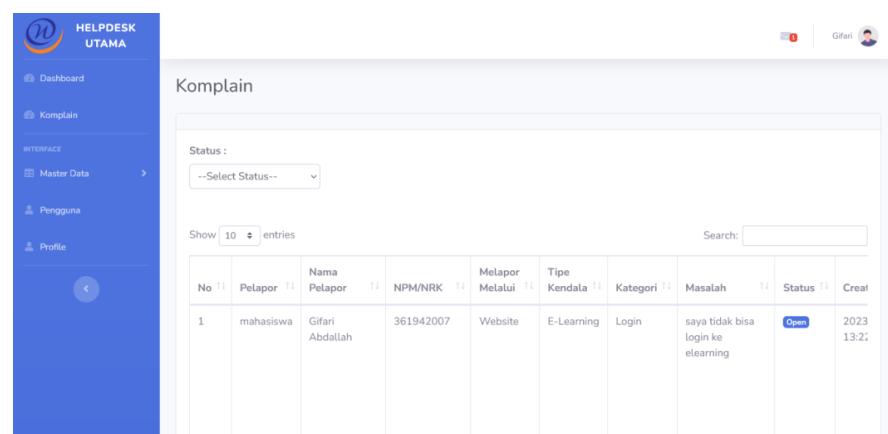
Gambar 4.6 Relasi Tabel

## 4.4 Implementasi Antarmuka

Berikut akan dijelaskan mengenai implementasi antarmuka:

### A. Tampilan Admin

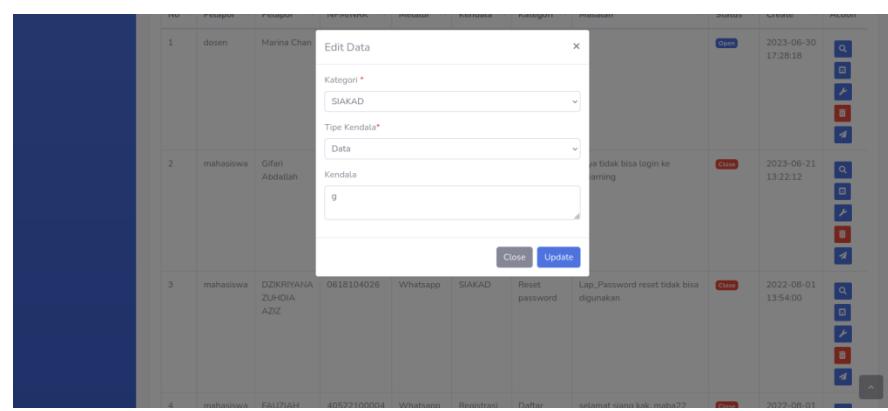
#### 1. Tampilan Menerima Pengajuan Komplain



The screenshot shows a web-based application interface titled "HELPDESK UTAMA". On the left, there is a sidebar with navigation links: Dashboard, Komplain, INTERFACE, Master Data, Pengguna, and Profile. The main content area is titled "Komplain" and displays a table of received complaints. The table has columns for No, Pelapor, Nama Pelapor, NPM/NRK, Melapor Melalui, Tipe Kendala, Kategori, Masalah, Status, and Creat. One row is visible, showing a complaint from a student named Gifari Abdallah with the ID 361942007, reporting a problem with e-learning login. The status is "Open" and it was created on 2023-06-30 at 13:21.

Gambar 4.7 Tampilan Menerima Pengajuan Komplain

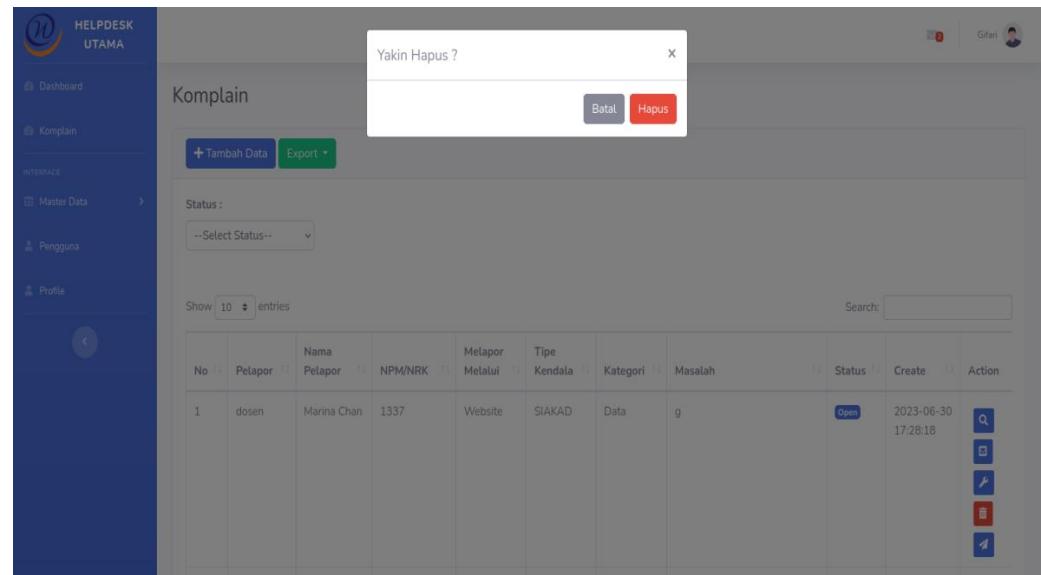
#### 2. Tampilan Mengubah Komplain



The screenshot shows a modal dialog box titled "Edit Data" overlaid on a list of complaints. The list includes three entries, each with a "dosen" or "mahasiswa" status, a name, and a date. The second entry, a complaint from a student named Gifari Abdallah, is selected. The modal dialog contains fields for "Kategori" (set to "SIAKAD") and "Tipe Kendala" (set to "Data"). A text area contains the message: "ya tidak bisa login ke elearning". At the bottom of the modal are "Close" and "Update" buttons.

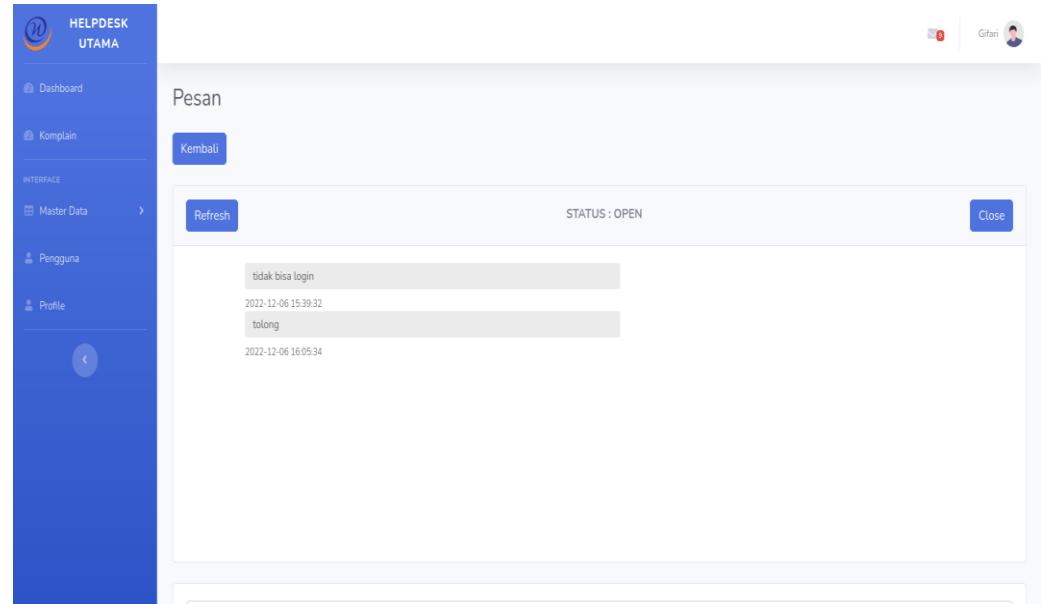
Gambar 4.8 Tampilan Mengubah Komplain

### 3. Tampilan Menghapus Komplain



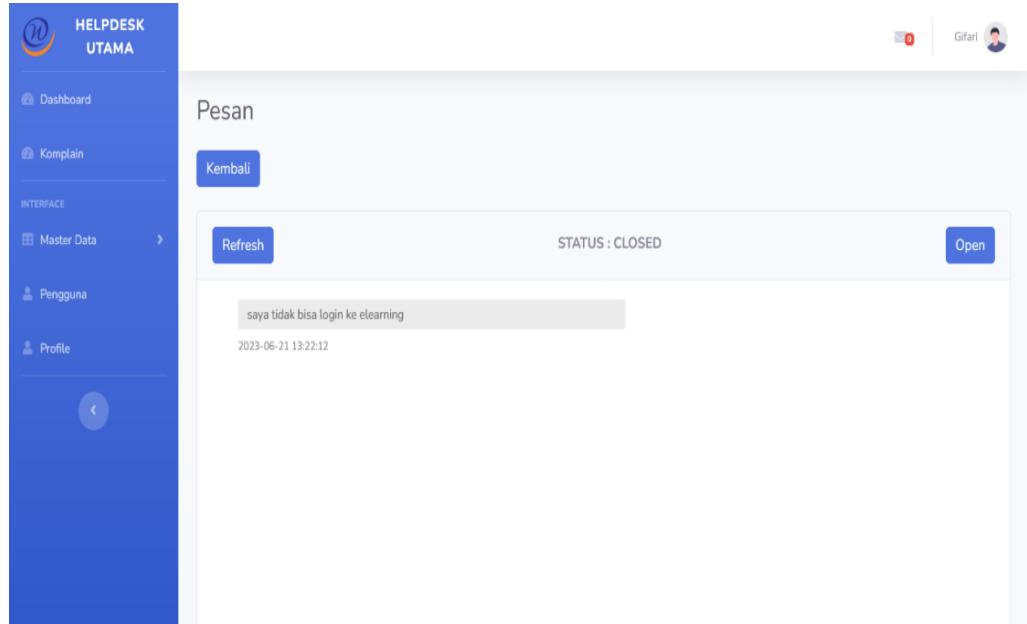
Gambar 4.9 Tampilan Menghapus Komplain

### 4. Tampilan Merespon Pesan



Gambar 4.10 Tampilan Merespon Pesan

## 5. Tampilan Mengubah Status



Gambar 4.11 Tampilan Mengubah Status

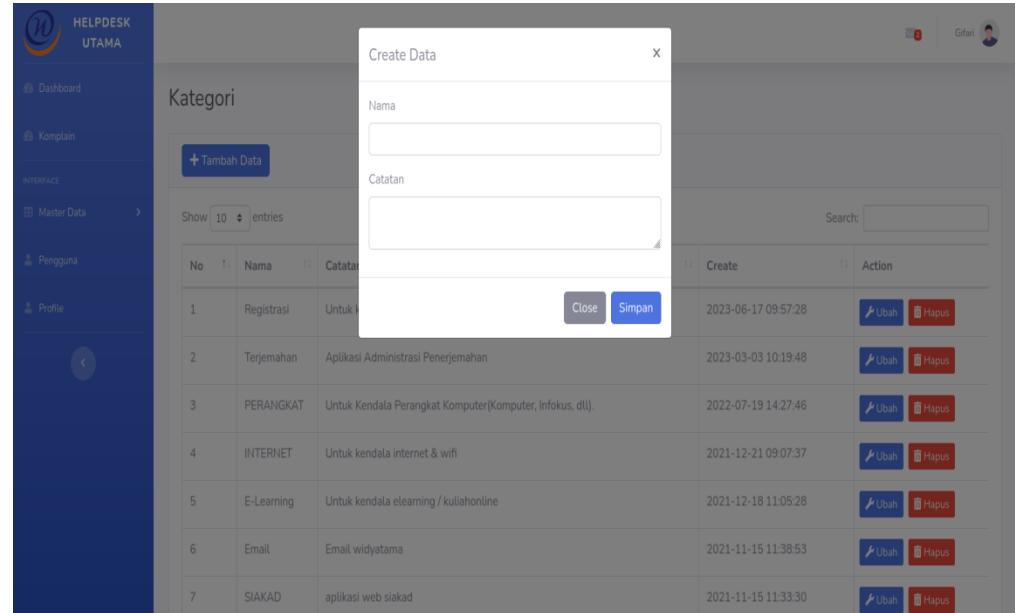
## 6. Tampilan Mengirim Komplain Ke Teknisi

The screenshot shows a list of complaints ('Komplain') in the Helpdesk Utama application. The left sidebar is identical to the previous screenshot. The main area is titled 'Komplain' and contains a table with the following data:

	Nama Pelapor	NPM/NRK	Melapor Melalui	Tipe Kendala	Kategori	Masalah	Status	Create	Action
wa	Gifari Abdallah	361942007	Website	E-Learning	Login	saya tidak bisa login ke elearning	<span style="color: red;">Close</span>	2023-06-21 13:22:12	<span style="color: blue;">Q</span> <span style="color: blue;">S</span> <span style="color: blue;">F</span> <span style="color: red;">D</span> <span style="color: blue;">A</span>

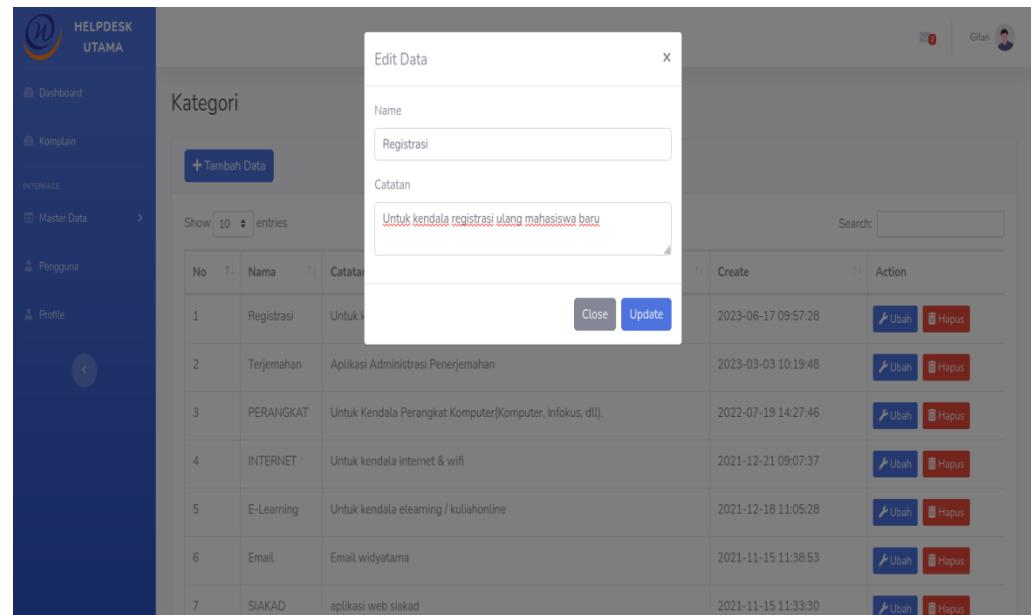
Gambar 4.12 Tampilan Mengirim Komplain Ke Teknisi

## 7. Tampilan Membuat Kategori



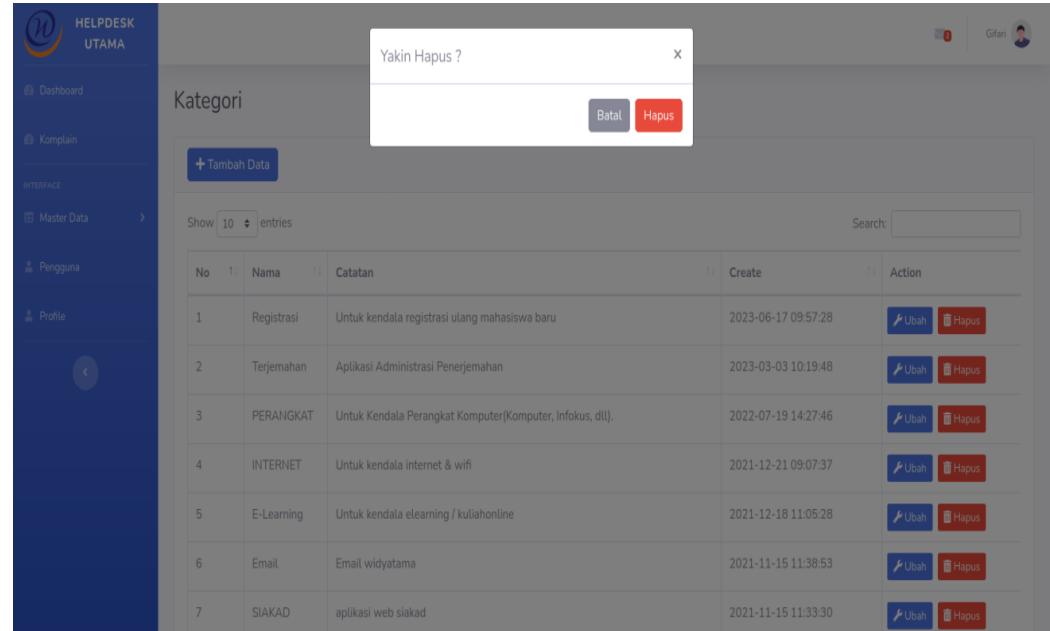
Gambar 4.13 Tampilan Membuat Kategori

## 8. Tampilan Mengubah Kategori



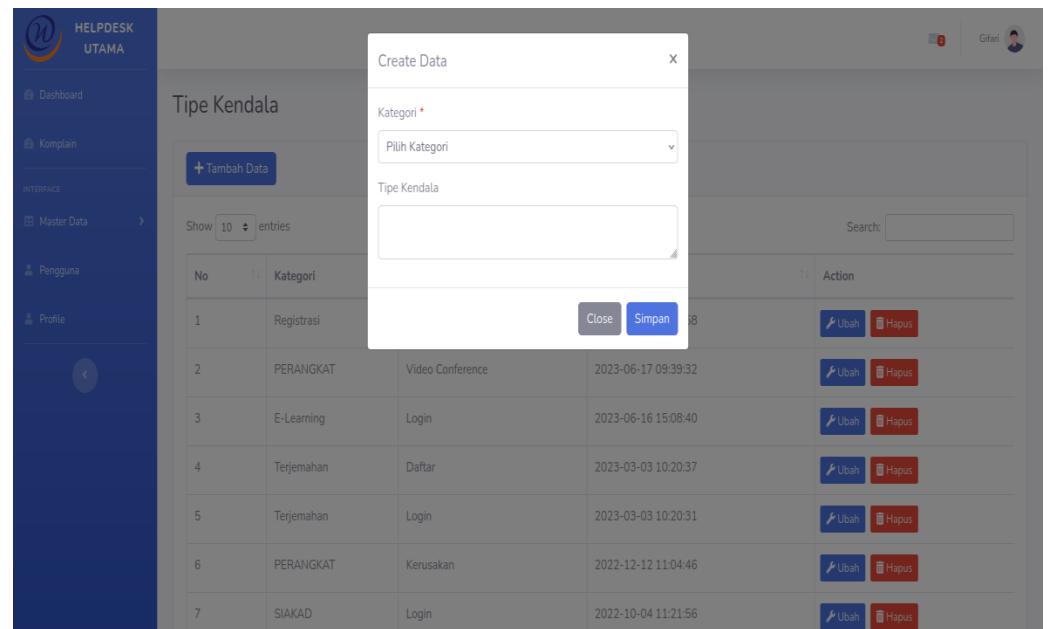
Gambar 4.14 Tampilan Mengubah Kategori

## 9. Tampilan Menghapus Kategori



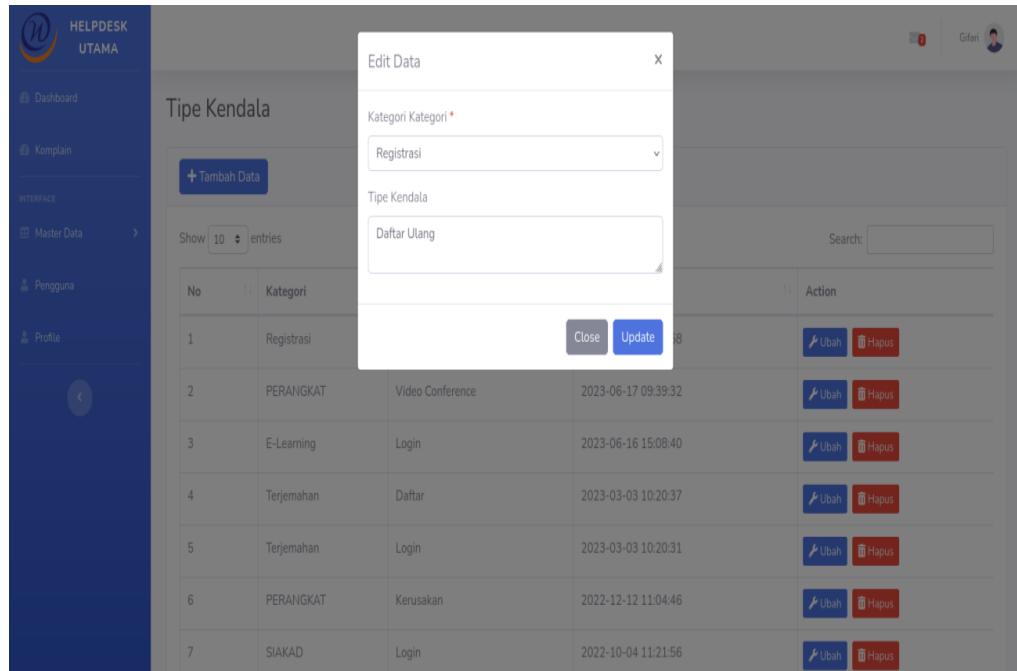
Gambar 4.15 Tampilan Menghapus Kategori

## 10. Tampilan Membuat Tipe Kendala



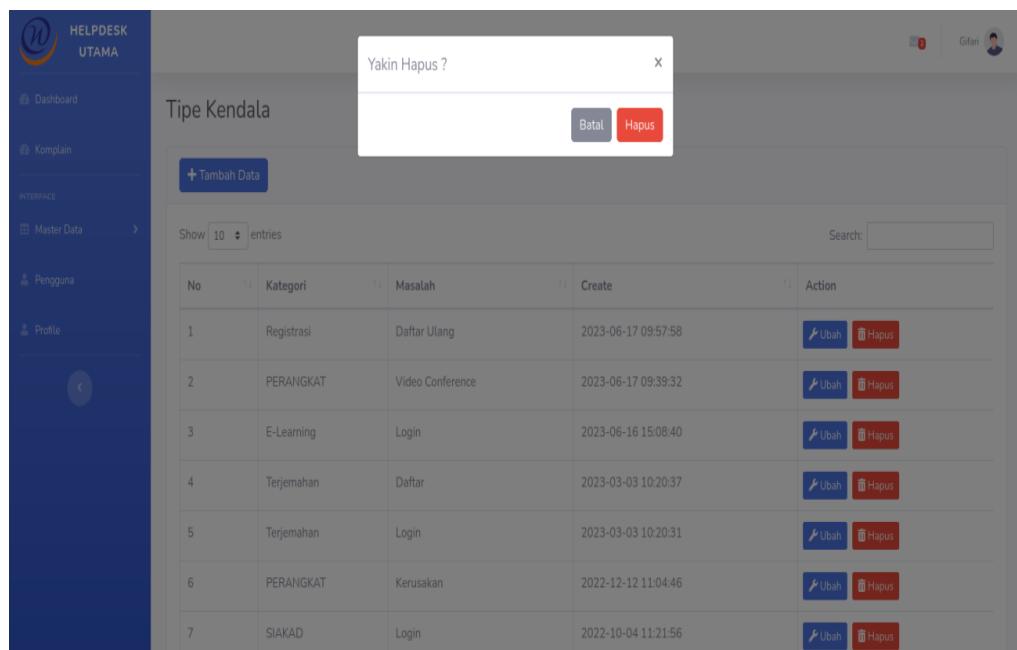
Gambar 4.16 Tampilan Membuat Tipe Kendala

## 11. Tampilan Mengubah Tipe Kendala



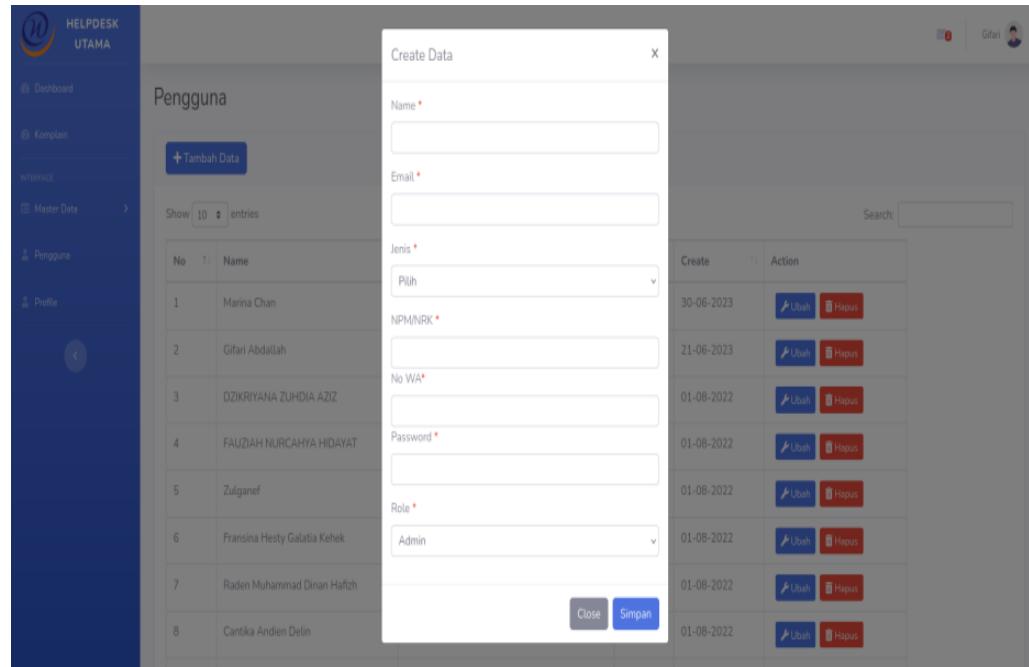
Gambar 4.17 Tampilan Mengubah Tipe Kendala

## 12. Tampilan Menghapus Tipe Kendala



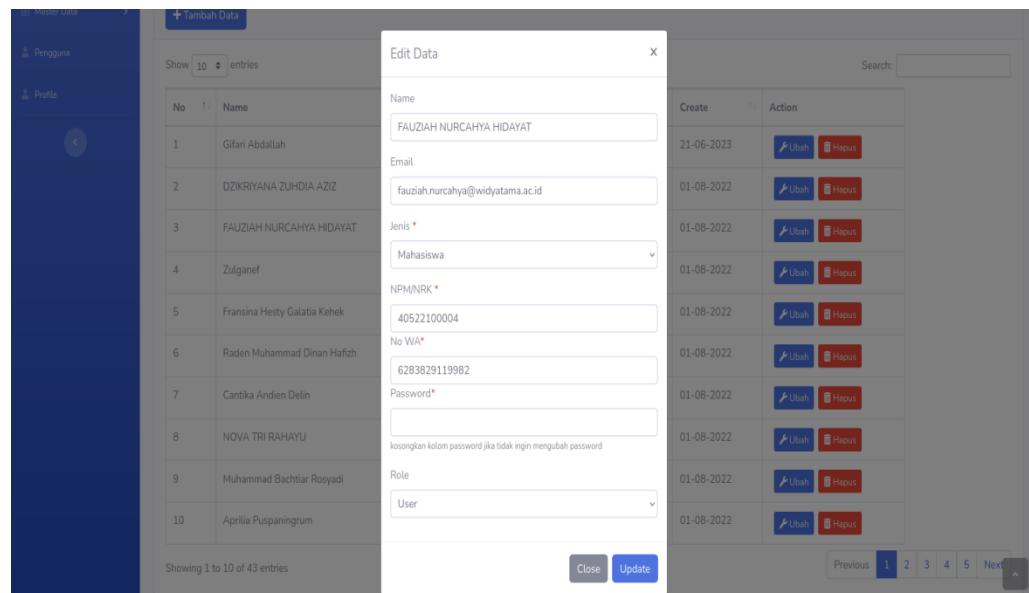
Gambar 4.18 Tampilan Menghapus Tipe Kendala

### 13. Tampilan Membuat Pengguna



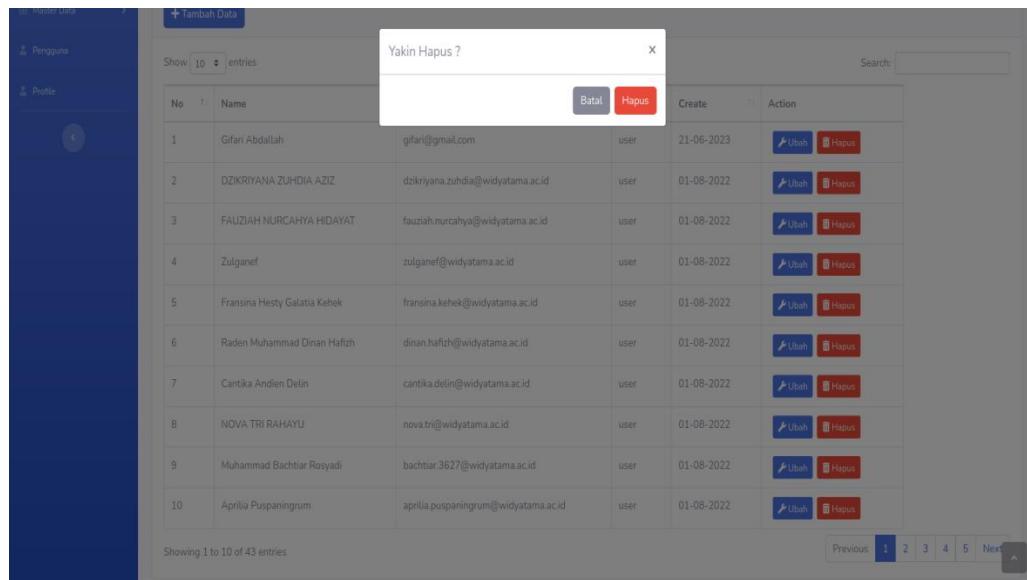
Gambar 4.19 Tampilan Membuat Pengguna

### 14. Tampilan Mengubah Pengguna



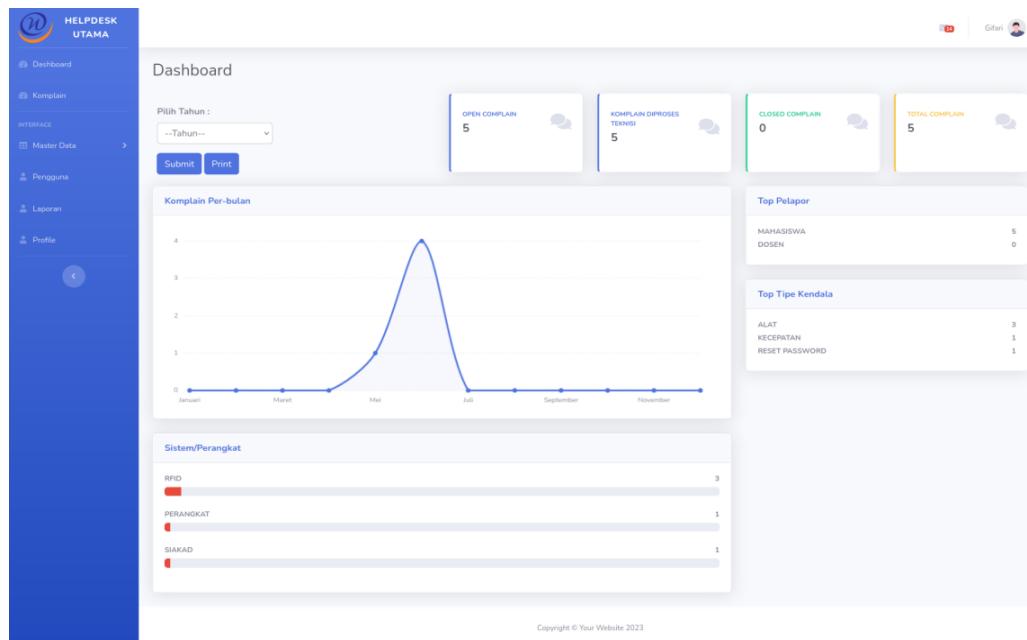
Gambar 4.20 Tampilan Mengubah Pengguna

## 15. Tampilan Menghapus Pengguna



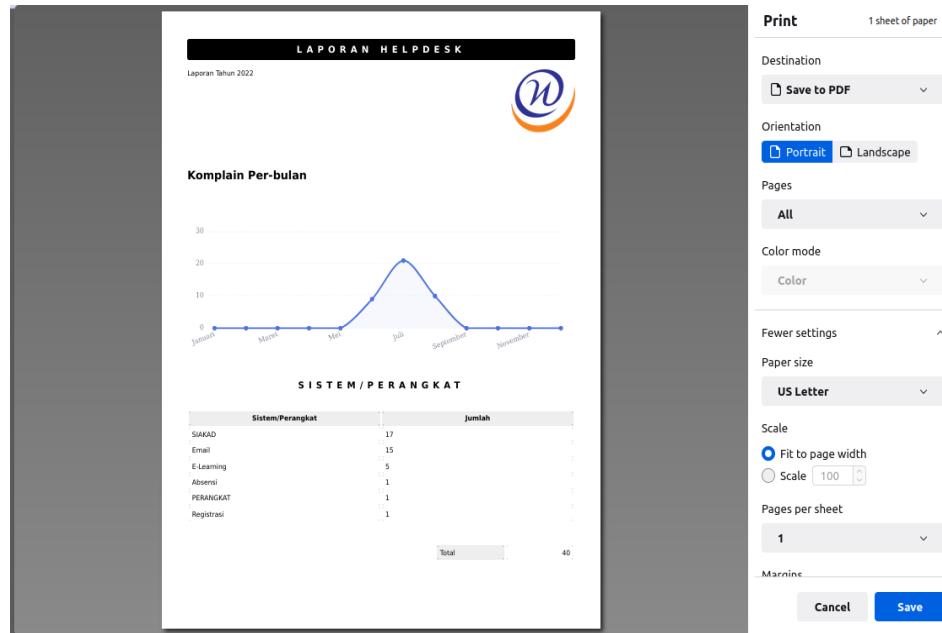
Gambar 4.21 Tampilan Menghapus Pengguna

## 16. Tampilan Melihat Laporan



Gambar 4.22 Tampilan Melihat Laporan

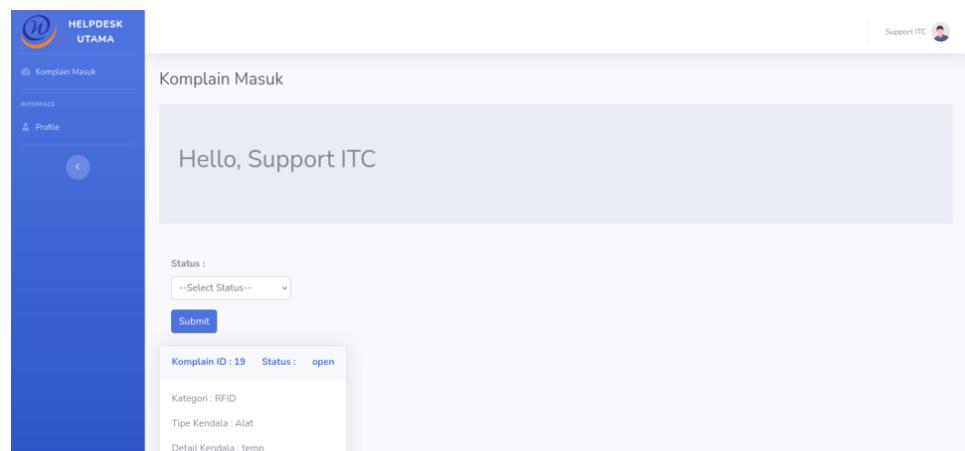
## 17. Tampilan Mencetak Laporan



Gambar 4.23 Tampilan Mencetak Laporan

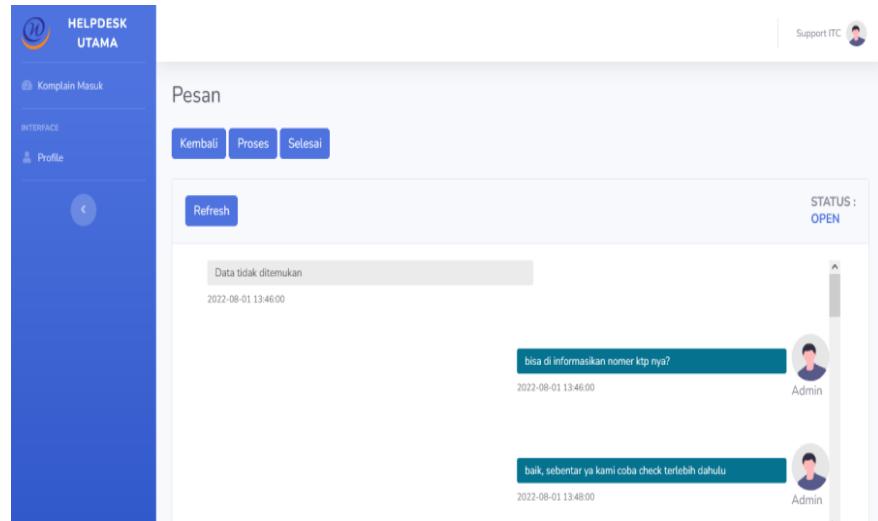
## B. Tampilan Teknisi

### 18. Tampilan Komplain Masuk



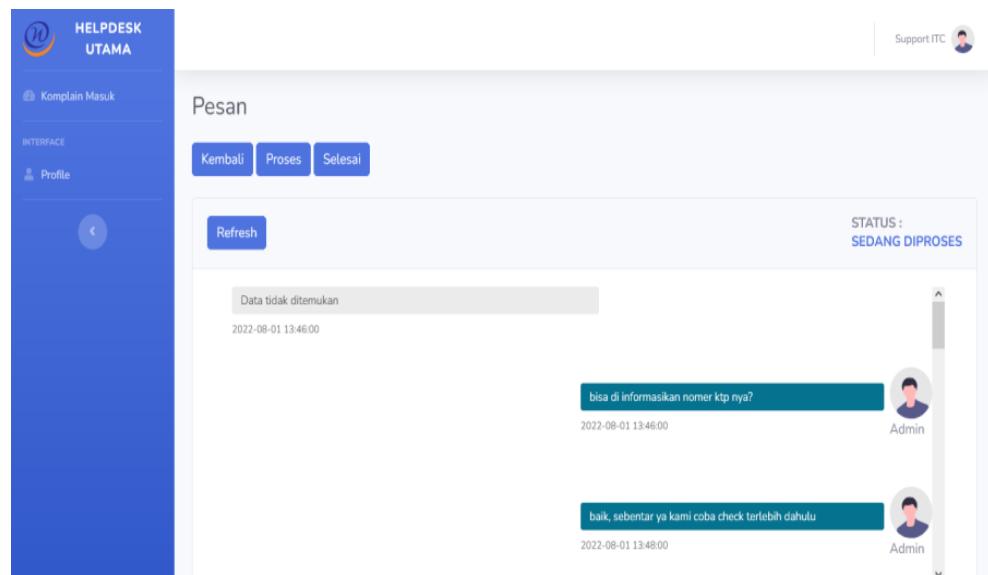
Gambar 4.24 Tampilan Komplain Masuk

### 19. Tampilan Melihat Pesan



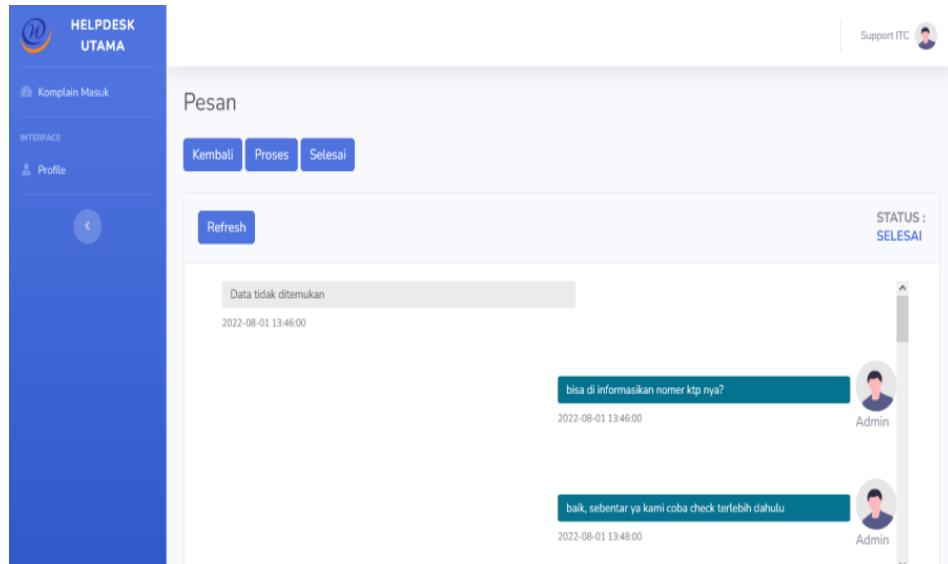
Gambar 4.25 Tampilan Melihat Pesan

### 20. Tampilan Memproses Komplain



Gambar 4.26 Tampilan Memproses Komplain

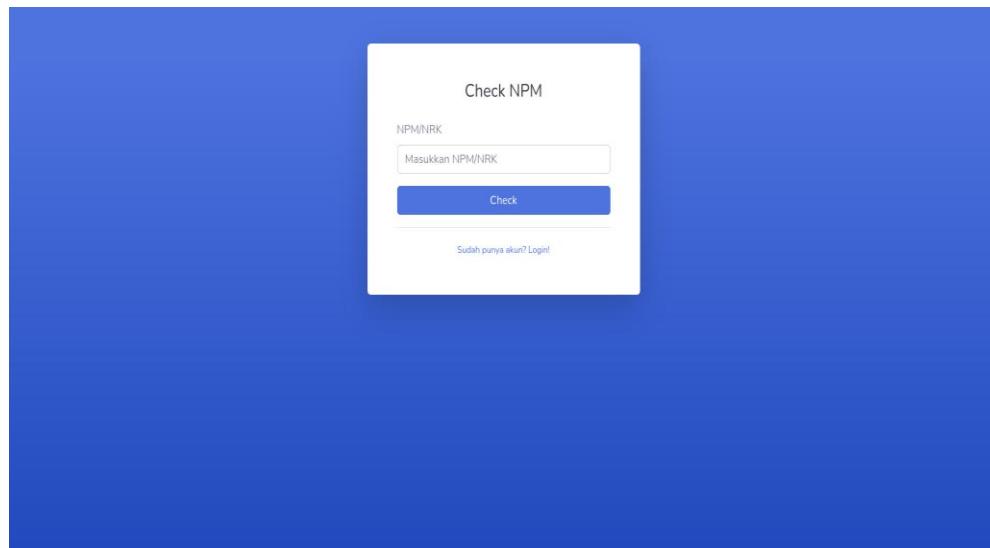
## 21. Tampilan Menyelesaikan Komplain



Gambar 4.27 Tampilan Menyelesaikan Komplain

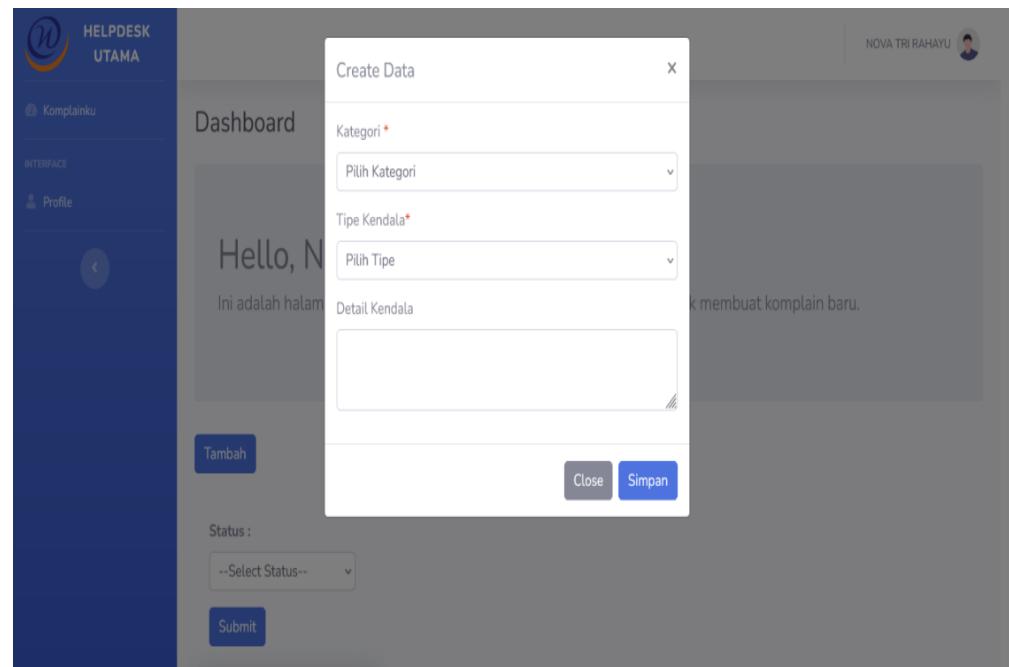
## C. Tampilan Pengguna

### 22. Tampilan Melakukan Pendaftaran



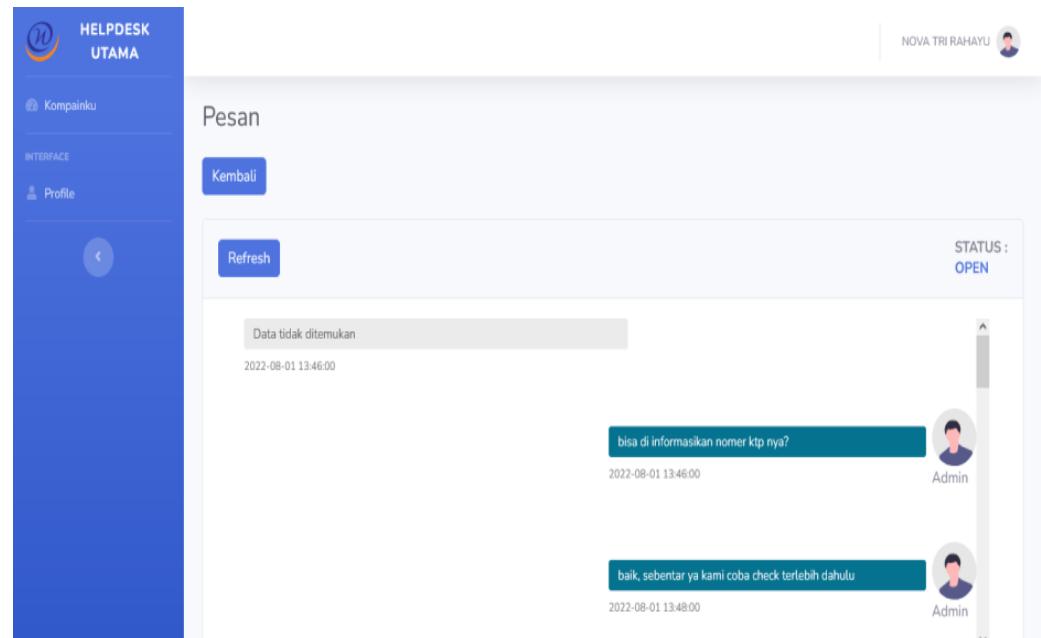
Gambar 4.28 Tampilan Melakukan Pendaftaran

### 23. Tampilan Membuat Pengajuan Komplain



Gambar 4.29 Tampilan Membuat Pengajuan Komplain

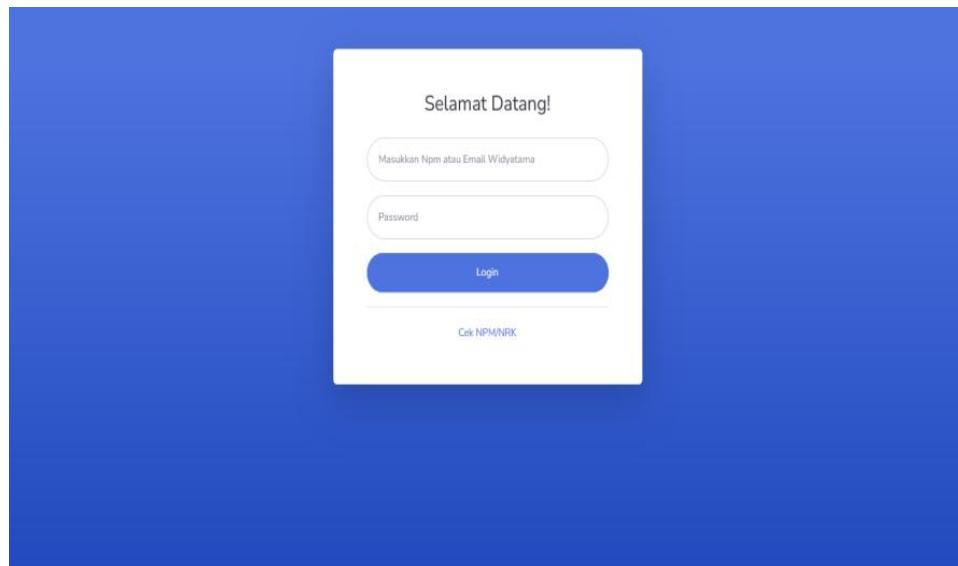
### 24. Tampilan Membalas Pesan



Gambar 4.30 Tampilan Membalas Pesan

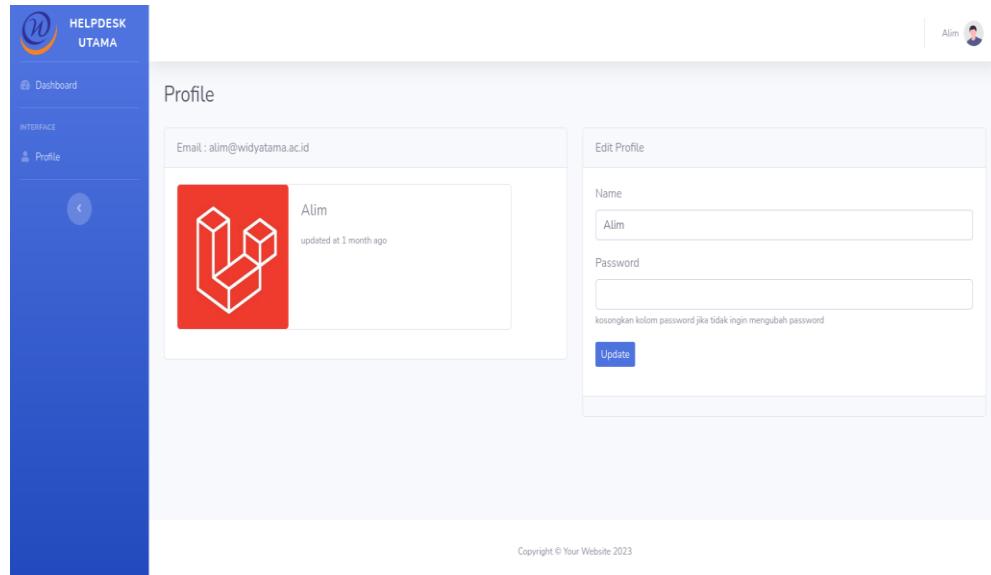
## B. Tampilan Semua Pengguna

### 17. Tampilan *Login*



Gambar 4.31 Tampilan *Login*

### 17. Tampilan Mengubah *Password*



Gambar 4.32 Tampilan Mengubah *Password*

#### 4.5 Pengujian

Aplikasi Layanan *Helpdesk* berbasis *website* pada Universitas Widyaatama ini telah diuji menggunakan pengujian *black box* berikut hasil pengujianya:

1. Pengujian *Admin*

Tabel 4.4 Pengujian *Blackbox* Menerima Pengajuan Komplain

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan menu Komplain	Menekan menu Komplain	Sistem menampilkan data-data komplain	Sesuai Harapan	Valid

Tabel 4.5 Pengujian *Blackbox* Mengubah Komplain

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol kunci ( <i>edit</i> ) pada salah satu data ditabel	Menekan tombol kunci	Sistem menampilkan halaman edit data komplain sesuai data yang dipilih	Sesuai Harapan	Valid
2	Mengubah form edit komplain lalu menekan tombol simpan	Mengubah isi form -Melapor (isi) -Kategori (isi) -Tipe Kendala (isi) -Kendala (Isi)	Sistem menyimpan komplain yang sudah diubah pada basis data	Sesuai Harapan	Valid

Tabel 4.6 Pengujian Blackbox Menghapus Komplain

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol sampah ( <i>delete</i> ) pada salah satu data ditabel	Menekan tombol sampah	Sistem menampilkan notifikasi berupa pertanyaan akan dihapus, jika hapus maka data dihapus, dan jika tidak maka klik <i>close</i>	Sesuai Harapan	Valid

Tabel 4.7 Pengujian Blackbox Merespon Pesan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol surat (pesan) pada salah satu data ditabel	Menekan tombol surat	Sistem menampilkan halaman pesan berdasarkan komplain yang dipilih	Sesuai Harapan	Valid
2	Mengisi form pesan lalu menekan tombol simpan	Mengisi form pesan	Sistem menyimpan pesan yang <i>diinput</i> dan menampilkan pesan yang sudah dibuat	Sesuai Harapan	Valid
3	Mengosongkan semua data di form pesan	Mengosongkan semua data di form pesan	Sistem menolak menyimpan dan tampilkan pesan kolom tidak boleh kosong	Sesuai Harapan	Valid

Tabel 4.8 Pengujian *Blackbox* Mengubah Status

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol <i>close/Open</i>	Menekan tombol <i>close/Open</i>	Status Komplain berubah	Sesuai Harapan	Valid

Tabel 4.9 Pengujian *Blackbox* Mengirim Komplain Ke Teknisi

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol kapal kertas(kirim ke telegram) pada salah satu data ditabel	Menekan tombol kapal kertas	Sistem mengirimkan komplain yang dipilih ke teknisi, dan mengirim notifikasi telegram	Sesuai Harapan	Valid

Tabel 4.10 Pengujian *Blackbox* Membuat Kategori

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan menu master data, lalu menekan menu kategori	Menekan menu kategori	Sistem menampilkan data-data kategori	Sesuai Harapan	Valid
2	Menekan tombol tambah data	Menekan tombol tambah data	Menampilkan halaman tambah kategori	Sesuai Harapan	Valid
3	Mengisi form kategori lalu menekan tombol	-Nama (isi) -Catatan (isi)	Sistem menyimpan kategori dan Menampilkan	Sesuai Harapan	Valid

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
	simpan		pesan data sudah berhasil dibuat		
4	Mengosongkan semua data di form kategori	-Nama (kosong) -Catatan (kosong)	Sistem menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid
5	Tidak memasukan data salah satu kolom form.	Form kategori -Nama (isi) -Catatan (kosong)	Sistem menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid

Tabel 4. 11 Pengujian *Blackbox* Mengubah Kategori

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol kunci (ubah) pada salah satu data ditabel	Menekan tombol kunci	Menampilkan halaman edit data kategori sesuai data yang dipilih	Sesuai Harapan	Valid
2	Mengubah form edit kategori lalu menekan tombol simpan	Mengubah isi form -Name -Catatan	Sistem menyimpan kategori yang sudah diubah pada basis data	Sesuai Harapan	Valid

Tabel 4.12 Pengujian *Blackbox* Menghapus Kategori

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol sampah ( <i>delete</i> ) pada salah satu data ditabel	Menekan tombol sampah	Sistem menampilkan notifikasi berupa pertanyaan yakin akan dihapus, jika hapus maka data dihapus, dan jika tidak maka klik <i>close</i>	Sesuai Harapan	Valid

Tabel 4.13 Pengujian *Blackbox* Membuat Tipe Kendala

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan menu master data, lalu menekan menu tipe kendala	Menekan menu tipe kendala	Sistem menampilkan data-data tipe kendala	Sesuai Harapan	Valid
2	Menekan tombol tambah data	Menekan tombol tambah data	Sistem menampilkan halaman tambah tipe kendala	Sesuai Harapan	Valid
3	Mengisi form tipe kendala lalu menekan tombol simpan	Nama (isi) Catatan (isi)	Sistem menyimpan tipe kendala dan Menampilkan pesan data sudah berhasil dibuat	Sesuai Harapan	Valid
4	Mengosongkan	Kategori	Sistem	Sesuai	Valid

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
	semua data di form tipe kendala	(kosong) Tipe Kendala (kosong)	menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Harapan	
5	Tidak memasukan data salah satu kolom form.	Form tipe kendala Kategori (isi) Tipe Kendala (kosong)	Sistem menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid

Tabel 4.14 Pengujian *Blackbox* Mengubah Tipe Kendala

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol kunci (ubah) pada salah satu data ditabel	Menekan tombol kunci	Menampilkan halaman edit data tipe kendala sesuai data yang dipilih	Sesuai Harapan	Valid
2	Mengubah form edit tipe kendala lalu menekan tombol simpan	Mengubah isi form - Kategori - Tipe Kendala	Sistem menyimpan tipe kendala yang sudah diubah	Sesuai Harapan	Valid

Tabel 4.15 Pengujian *Blackbox* Menghapus Tipe Kendala

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol sampah ( <i>delete</i> ) pada salah satu data ditabel	Menekan tombol sampah	Sistem menampilkan notifikasi berupa pertanyaan, jika hapus maka data dihapus, dan jika tidak maka klik <i>close</i>	Sesuai Harapan	Valid

Tabel 4.16 Pengujian *Blackbox* Membuat Pengguna

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan menu pengguna	Menekan menu pengguna	Menampilkan data-data pengguna	Sesuai Harapan	Valid
2	Menekan tombol tambah data	Menekan tombol tambah data	Menampilkan halaman tambah pengguna	Sesuai Harapan	Valid
3	Mengisi form pengguna lalu menekan tombol simpan	Name (isi) Email (isi) Pelapor (isi) NPM/NRK (isi) No WA (isi) Password (isi) <i>Role</i>	Sistem menyimpan pengguna dan Menampilkan pesan data sudah berhasil dibuat	Sesuai Harapan	Valid

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
		(isi)			
4	Mengosongkan semua data di form pengguna	Nama (kosong) Catatan (kosong)	Sistem menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid
5	Tidak memasukan data disalah satu kolom form.	Form pengguna Nama (isi) Catatan (kosong)	Sistem menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid

Tabel 4.17 Pengujian *Blackbox* Mengubah Pengguna

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol kunci (ubah) pada salah satu data ditabel	Menekan tombol kunci	Sistem menampilkan halaman edit data pengguna sesuai data yang dipilih	Sesuai Harapan	Valid
2	Mengubah form edit pengguna lalu menekan tombol simpan	Mengubah isi form Name Catatan	Sistem menyimpan pengguna yang sudah diubah pada basis data	Sesuai Harapan	Valid

Tabel 4.18 Pengujian Blackbox Menghapus Pengguna

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol sampah ( <i>delete</i> ) pada salah satu data ditabel	Menekan tombol sampah	Sistem menampilkan notifikasi berupa pertanyaan yakin akan dihapus, jika hapus maka data dihapus, dan jika tidak maka klik <i>close</i>	Sesuai Harapan	Valid

Tabel 4.19 Pengujian Blackbox Melihat Laporan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan menu <i>dashboard</i>	Menekan menu <i>dashboard</i>	Menampilkan data-data <i>dashboard</i>	Sesuai Harapan	Valid
2	Memilih tahun pada bagian <i>select box</i> , pilih tahun dan menekan tombol submit	<i>Select box</i> (Isi)	Sistem menampilkan data semua diagram berdasarkan tahun yang dipilih	Sesuai Harapan	Valid
3	Tidak memilih tahun pada bagian <i>select box</i>	<i>Select box</i> (Kosong)	Sistem menampilkan data semua diagram berdasarkan tahun ini (sekarang)	Sesuai Harapan	Valid

Tabel 4.20 Pengujian Blackbox Mencetak Laporan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Memilih tahun pada bagian <i>select box</i> , pilih tahun dan menekan tombol print	<i>Select box</i> (Isi)	Sistem menampilkan lalu mencetak data semua diagram berdasarkan tahun yang dipilih	Sesuai Harapan	Valid
2	Tidak memilih tahun pada bagian <i>select box</i> , pilih tahun dan menekan tombol print	<i>Select box</i> (Kosong)	Sistem menampilkan lalu mencetak data semua diagram berdasarkan tahun ini (sekarang)	Sesuai Harapan	Valid

## 2. Pengujian Teknisi

Tabel 4.21 Pengujian Blackbox Menerima Kiriman Komplain

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan menu komplain masuk	Menekan menu komplain masuk	Menampilkan data komplain masuk	Sesuai Harapan	Valid

Tabel 4.22 Pengujian *Blackbox* Melihat Pesan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol pesan pada komplain yang masuk ke menu komplain masuk	Menekan tombol pesan	Menampilkan data-data pesan berdasarkan komplain yang dipilih	Sesuai Harapan	Valid

Tabel 4.23 Pengujian *Blackbox* Memproses Komplain

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol proses pada halaman pesan yang dipilih	Menekan tombol proses	Status Komplain berubah menjadi “Sedang diproses”	Sesuai Harapan	Valid

Tabel 4.24 Pengujian *Blackbox* Menyelesaikan Komplain

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol proses pada halaman pesan yang dipilih	Menekan tombol Selesai	Status Komplain berubah menjadi “Selesai”	Sesuai Harapan	Valid

### 3. Pengujian Pengguna

Tabel 4.25 Pengujian *Blackbox* Melakukan Pendaftaran

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Memasukan data NPM/NRK yang sudah terdaftar pada kolom cek	NPM/NRK (isi)	Sistem menampilkan pesan status pengguna jika pengguna sudah terdaftar maka akan muncul notifikasi “ <i>User</i> sudah ada silahkan <i>login</i> ”	Sesuai Harapan	Valid
2	Memasukan data NPM/NRK yang sudah belum terdaftar pada kolom cek	NPM/NRK (isi)	Sistem menampilkan form daftar dan menampilkan pesan “ <i>User</i> tidak ditemukan, Silahkan Registrasi”	Sesuai Harapan	Valid
3	Memasukan data NPM/NRK yang sudah terdaftar tetapi belum memiliki <i>password</i> pada kolom cek	NPM/NRK (isi)	Sistem menampilkan form aktifasi	Sesuai Harapan	Valid
4	Tidak memasukan data salah satu kolom form.	NPM/NRK (kosong)	Sistem menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid
5	Mengisi bio data dan <i>password</i> dengan benar	Name (isi) Npm Email	Sistem menampilkan pesan data	Sesuai Harapan	Valid

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
		(isi) Whatsapp (isi) Password (isi) Confirm Password (isi)	berhasil dibuat		
6	Tidak memasukan data salah satu kolom form.	Name (isi) Npm Email (Kosong) Whatsapp (isi) Password (isi) Confirm Password (isi)	Sistem menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid

Tabel 4.26 Pengujian Blackbox Membuat Pengajuan Komplain

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol tambah pada halaman	Menekan tombol tambah	Memunculkan form tambah komplain	Sesuai Harapan	Valid
2	Menekan tombol pesan pada komplain	Menekan tombol pesan	Sistem menampilkan data pesan berdasarkan komplain yang dipilih	Sesuai Harapan	Valid
3	Mengisi form tambah komplain dengan benar	Kategori (isi) Tipe Kendala (isi)	Sistem menampilkan pesan data berhasil dibuat	Sesuai Harapan	Valid

		Detail Kendala (isi)			
4	Tidak memasukan data disalah satu kolom form.	Kategori (kosong) Tipe Kendala (isi) Detail Kendala (isi)	Sistem menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid

Tabel 4.27 Pengujian *Blackbox* Membalas Pesan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol pesan pada komplain dihalaman komplainku	Menekan tombol pesan	Sistem menampilkan data pesan berdasarkan komplain yang dipilih	Sesuai Harapan	Valid
2	Mengisi form pesan lalu menekan tombol simpan	Mengisi form pesan	Sistem menyimpan pesan yang <i>diinput dan Menampilkan</i> data pesan yang sudah berhasil dibuat	Sesuai Harapan	Valid
3	Mengosongkan semua data di form pesan	Mengosongkan semua data di form pesan	Sistem menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid

#### 4. Pengujian Semua Pengguna

Tabel 4.28 Pengujian *Blackbox Login*

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengosongkan semua data Email/NPM/NRK dan password, lalu langsung klik button login	Email/NPM/ NRK : (kosong) Password : (kosong)	Sistem menolak akses login masuk dan diharuskan mengisi semua form	Sesuai Harapan	Valid
2	Hanya mengisi Email/NPM/NRK dan password tidak disi tetapi klik button login	Email/NPM/ NRK : (Isi) Password : (kosong)	Sistem menolak akses login masuk dan diharuskan mengisi semua form	Sesuai Harapan	Valid
3	Mengisi salah satu dengan data yang benar, dan untuk password menggunakan data yang salah	Email/NPM/ NRK : (Isi) Password : (xxxxx)	Sistem menolak akses login masuk dan menampilkan pesan “Login gagal periksa kembali email atau username”	Sesuai Harapan	Valid
4	Mengisi semua dengan data yang benar	Email/NPM/ NRK : (isi) Password : (xxxxx)	Sistem menolak akses login masuk, Dan diarahkan ke halaman utama	Sesuai Harapan	Valid

Tabel 4.29 Pengujian *Blackbox* Mengubah *Password*

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan menu <i>profile</i>	Menekan menu <i>profile</i>	Sistem menampilkan halaman <i>profile</i>	Sesuai Harapan	Valid
2	Mengubah form <i>profile</i> lalu menekan tombol simpan	Name (isi) Password (isi)	Sistem menyimpan <i>profile</i> dan Menampilkan pesan <i>profile</i> sudah berhasil di <i>update</i>	Sesuai Harapan	Valid
3	Mengosongkan semua data di form pengguna	Nama (kosong) Password (kosong)	Sistem menolak menyimpan dan menampilkan pesan kolom tersebut tidak boleh kosong	Sesuai Harapan	Valid
4	Tidak memasukan data disalah satu kolom form.	Form pengguna Nama (isi) Password (kosong)	Sistem menyimpan <i>profile</i> dan Menampilkan pesan <i>profile</i> sudah berhasil di <i>update</i>	Sesuai Harapan	Valid

Berdasarkan dari pengujian *blackbox* yang dilakukan, dimana pengujiannya terdiri dari 4 (empat) pengujian untuk “menerima pengajuan komplain”, 2 (dua) pengujian untuk mengubah komplain, 1 (satu) pengujian untuk menghapus komplain, 2 (dua) pengujian untuk merespon pesan, 1 (satu) pengujian untuk mengubah status, 1 (satu) pengujian untuk mengirim komplain ke teknisi, 5 (lima) pengujian untuk membuat kategori, 2 (dua) pengujian untuk mengubah kategori, 1

(satu) pengujian untuk menghapus kategori, 5 (lima) pengujian untuk membuat tipe kendala, 2 (dua) pengujian untuk mengubah tipe kendala, 1 (satu) pengujian untuk menghapus tipe kendala, 5 (lima) pengujian untuk membuat pengguna, 2 (dua) pengujian untuk mengubah pengguna, 1 (satu) pengujian untuk menghapus pengguna, 3 (tiga) pengujian untuk melihat laporan, 2 (dua) pengujian untuk mencetak laporan, 1 (satu) pengujian untuk menerima kiriman komplain, 1 (satu) pengujian untuk melihat pesan, 1 (satu) pengujian untuk memproses komplain, 1 (satu) pengujian untuk menyelesaikan komplain, 6 (enam) pengujian untuk halaman melakukan pendaftaran, 4 (empat) pengujian untuk membuat pengajuan komplain, 3 (tiga) pengujian membalas pesan, 4 (empat) pengujian untuk halaman *login*, 4 (empat) pengujian untuk *mengubah password*. Sehingga total keseluruhan pengujian adalah sebanyak 62 pengujian dan semua hasil yang diharapkan sesuai.

## **BAB V**

### **PENUTUP**

Pada bab penutup ini berisi mengenai kesimpulan dan saran dari penulis berkaitan dengan aplikasi yang telah dibangun.

#### **5.1 Simpulan**

Berdasarkan perancangan dan pembangunan “PENGEMBANGAN APLIKASI PENGELOLAAN LAYANAN *HELPDESK* BERBASIS WEB DI UNIVERSITAS WIDYATAMA” maka dapat diambil kesimpulan sebagai berikut :

1. Aplikasi pengelolaan layanan *helpdesk* yang dibuat dapat mempermudah mahasiswa dan dosen dalam melakukan pelaporan komplain secara daring karena laporan dilakukan dari satu aplikasi yang membuat mahasiswa dapat mengecek status sudah ditangani atau belum, dan aplikasi tersebut dapat diakses dari mana saja menggunakan perangkat yang terhubung ke *internet*.
2. Aplikasi pengelolaan layanan *helpdesk* yang dibuat dapat memudahkan tim *helpdesk* Widyatama dalam melakukan pelayanan dan pengelolaan data komplain baik pencatatan atau penghitungan, dimana informasi tersebut terdapat pada menu utama/*dashboard*, serta melalui *intergrasi* aplikasi telegram memudahkan dalam penyampaian sistem yang bermasalah kepada staf

PTI.

## 5.2 Saran

Penulis menyadari bahwa pada “PENGEMBANGAN APLIKASI PENGELOLAAN LAYANAN *HELPDESK* BERBASIS WEB DI UNIVERSITAS WIDYATAMA” ini masih belum sempurna, sehingga perlu penyempurnaan secara berkesinambungan sesuai dengan kebutuhan. Oleh karena itu penulis memberi saran sebagai berikut :

1. Aplikasi ini hanya dapat melakukan pendataan sederhana, diperlukan pengolahan lebih lanjut untuk pembuatan laporan yang lebih detail.
2. Penambahan *reset password*, untuk aplikasi saat ini belum adanya *reset password* sehingga saat ini, jika mahasiswa/dosen lupa *password* perlu menghubungi admin untuk mengubah *password*.
3. Penambahan intergrasi data ke *database* akademik, sehingga mahasiswa/dosen tidak perlu mengisi semua data yang sudah ada di akademik.
4. Penambahan intergrasi tambahan dengan aplikasi *chat* seperti whatsapp dan telegram, sehingga mahasiswa/dosen dapat melakukan pelaporan selain melalui *website*.

## DAFTAR PUSTAKA

- Abdulloh, Rohi. 2018. “7 in 1 Pemrograman Web Tingkat Lanjut.” *7 in 1 Pemrograman Web Tingkat Lanjut 7–8*.
- Astuti, Iftadi. 2016. Analisis dan Perancangan Sistem Kerja. Yogyakarta:Deepublish.
- Azmi Farhan. 2022. “Rancang Bangun Aplikasi Pengelolaan Layanan Ticketing Helpdesk Berbasis Web (Studi Kasus: Pt. Ussi Bandung).”
- Bunardi, B., Naga, D. S., & Arisandi, D. (2019). Pengembangan Aplikasi E-Commerce Produk Lokal Dan Data Kependudukan Pada Desa Giritengah, Borobudur. *Computatio : Journal of Computer Science and Information Systems*, 3(1), 77. <https://doi.org/10.24912/computatio.v3i1.4274>
- Elgamar. 2020. BUKU AJAR KONSEP DASAR PEMROGRAMAN WEBSITE DENGAN PHP. Ahlimedia Book.
- Fatmasari, Fatmasari, and Siti Sauda. 2020. “Pemodelan Unified Modeling Language Sistem Informasi Enterprise Resource Planning.” *Jurnal Media Informatika Budidarma* 4(2):429. doi: 10.30865/mib.v4i2.2022.
- Helmund, Elly. 2021. “Optimasi Basis Data Oracle Menggunakan Complex View Studi Kasus : PT. Berkat Optimis Sejahtera (PT.BOS) Pangkalpinang.” *Jurnal Informatika* 7(1):80–86.
- Kai, Harry Novianto, Sherwin R. U. A. Sompie, Alwin M. Sambul, Teknik Elektro, Universitas Sam, Ratulangi Manado, and Jl Kampus Bahu-unrat Manado. 2018. “AplikasiLayananPengangkutanSampah.” 13(4):1–12.
- Moch Zawaruddin Abdullah, Mungki Astiningrum, Yuri Ariyanto, Dwi Puspitasari, and Atiqah Nurul Asri. 2021. “Rancang Bangun Sistem Informasi Akuntansi Berbasis Website Menggunakan Framework Laravel.” *Jurnal Pengabdian Polinema Kepada Masyarakat* 8(1):74–80. doi: 10.33795/jppkm.v8i1.64.
- Mustopa, Ali. 2017. “SISTEM INFORMASI IT-HELPDESK PADA UNIVERSITAS AMIKOM YOGYAKARTA BERBASIS WEB.” *JIKO (Jurnal Informatika Dan Komputer)* 2(2):93. doi: 10.26798/jiko.2017.v2i2.71.

- Putra, Dede Wira Trise, and Rahmi Andriani. 2019. “Unified Modelling Language (UML) Dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD.” *Jurnal TeknoIf* 7(1):32. doi: 10.21063/jtif.2019.v7.1.32-39.
- R. S. Pressman, Rekayasa Perangkat Lunak (Pendekatan Praktisi), Yogyakarta: Penerbit ANDI, 2012.
- Rachmawati, U. A., Adam, S., & Alwi, S. H. (2019). Pembangunan Helpdesk Ticketing System Berbasis Web (Studi Kasus: Universitas Yarsi). *Jurnal Teknologi Informasi YARSI*, 6(1), 19–24. <https://doi.org/10.33476/jtiy.v6i1.597>
- Raharjo, Budi. 2021. *P Y Yayasan Prima Agus Teknik Sistem Manajemen Database*.
- Rizal, Nanda Muhamad. 2020. “Penerapan Customer Relationship Management (Crm ) Berbasis Website ( Studi Kasus Di Toko Online Shop Fauxhawk Store Bandung ).” 1–49.
- Sadeli, Muhammad. 2014. “Aplikasi Bisnis Dengan Php Dan MySQL.” *Palembang: Maxikom* 4.
- Setya Budi, Achmad, Ariz Nur Annafi, Diky Syafitra, Dimas Danang, Aji Wahyu Islam, Fikri Agustiawan, Imam Balda, Rapli Muhammad Ardiansah, Rayhan Bilal Alghifary, Rescue Asmara, Rizky Ardiansyah, and Galuh Saputri. 2022. “Pengenalan Dan Penggunaan Dbms (Database Management System) Di Smk Tunas Media Kota Depok.” *Abdi Jurnal Publikasi* 1(2):52–57.
- Situmorang, Harold. 2019. “Sistem Informasi Pengelolahan Data Alumni Berbasis Web (Studi Pada Fakultas Sain, Teknologi Dan Informasi) Universitas Sari Mutiara Indonesia.” *Jurnal Mahajana Informasi* 4(1):34–48.
- Suawa, Pascallino Julian, Novie R. Pioh, and Welly Waworundeng. 2021. “Manajemen Pengelolaan Dana Revitalisasi Danau Tondano Oleh Pemerintah Kabupaten Minahasa(Studi Kasus Di Balai Wilayah Sungai Sulawesi).” *Jurnal Governance* 1(2):2088–2815.
- Suryana, Nurdin. 2018. “PERANCANGAN SISTEM E-SERVICE MENGGUNAKAN FRAMEWORK CODEIGNITER DI PT. INDOFOOD FRITOLAY MAKMUR.” Universitas Raharja Tangerang.
- Susilawati, Tuti, Fanny Yuliansyah, Muhammad Romzi, and Rintan Aryani. 2020. “Membangun Website Toko Online Pempek Nthree Menggunakan Php Dan Mysql.” *Jurnal Teknik Informatika Mahakarya (JTIM)* 3(1):35–44.

- Triaulia, Eli, Zulham Zulham, and Ibnu Rusydi. 2021. "Sistem Informasi Pendataan Hasil Penjualan Bbm Pada Spbu Kso 14.202.1119 Sutomo Medan." *Device : Journal of Information System, Computer Science and Information Technology* 1(2). doi: 10.46576/device.v1i2.1196.
- Uminingsih, Muhamad Nur Ichsanudin, Muhammad Yusuf, and Suraya Suraya. 2022. "Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula." *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer* 1(2):1–8. doi: 10.55123/storage.v1i2.270.
- Wahyudi, Rizki, Ema Utami, and M. Rudyant. Arief. 2016. "Sistem Pakar E-Tourism Pada Dinas Pariwisata D.I.Y Menggunakan Metode Forward Chaining." *Jurnal Ilmiah DASI* 17(2):67–75.
- Wardhani, Rika Novita, Meinarini Catur Utami, and Ibnu Yahya Saputra. 2020. "SISTEM INFORMASI HELPDESK TICKETING PADA PT. BANK MEGA Tbk." *Jurnal Ilmiah Matrik* 22(2):201–7. doi: 10.33557/jurnalmatrik.v22i2.868.

## **LAMPIRAN-LAMPIRAN**

### **1. Surat Pernyataan telah Melakukan Wawancara**

**Surat Bukti Pelaksanaan Survei dan Wawancara**

Yang bertandatangan dibawah ini :

Nama : Gifari Abdallah  
NIP : 361942007  
Program Studi : Teknik Informatika  
Universitas : Sekolah Tinggi Manajemen Informatika Dan Komputer Indonesia Mandiri

Telah melakukan survei wawancara dengan :

Nama : Fandry Nastaliyantono  
Jabatan : Staf Pusat Teknologi Informasi  
Perusahaan : Universitas Widyatama  
Alamat : Jl. Cikutra No.204 A

Dalam rangka melaksanakan studi untuk pengumpulan data-data yang dianggap perlu dan dibutuhkan dalam penyusunan skripsi Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika Dan Komputer Indonesia Mandiri Bandung.

Mengetahui,

  
(Fandry Nastaliyantono)

Bandung, 9 Juni 2023

  
(Gifari Abdallah)