

SKRIPSI

**PERANCANGAN APLIKASI SISTEM *HELPDESK* BERBASIS *WEBSITE*
PADA DIREKTORAT TEKNOLOGI INFORMASI
UNIVERSITAS HASANUDDIN MAKASSAR**



Oleh:

FITRIANTINI SUAIB

NIM: K105 82 1662 15

HADRAYANTI

NIM: K105 82 1672 15

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK ELEKTRO

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2018

**PERANCANGAN APLIKASI SISTEM *HELPDESK* BERBASIS *WEBSITE*
PADA DIREKTORAT TEKNOLOGI INFORMASI
UNIVERSITAS HASANUDDIN MAKASSAR**

Skripsi

Diajukan sebagai salah satu syarat
untuk memperoleh gelar Sarjana
Program Studi Teknik Telekomunikasi
Jurusan Teknik Elektro
Fakultas Teknik

Disusun Dan diajukan oleh :

FITRIANTINI SUAIB
NIM: K105 82 1662 15

HADRAYANTI
NIM: K105 82 1672 15

PADA

UNIVERSITAS MUHAMMADIYAH MAKASSAR

MAKASSAR

2018



UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221
Website: www.umh.ac.id, e-mail: umh@umh.ac.id
Website: <http://teknik.umh.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Teknik (ST) Program Studi Teknik Elektro Jurusan Teknik Elektro Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PERANCANGAN APLIKASI SISTEM HELPDESK BERBASIS WEBSITE PADA DIREKTORAT TEKNOLOGI INFORMASI UNIVERSITAS HASANUDDIN MAKASSAR**

Nama : 1. Fitriantini Suaib
2. Hadrayanti

Stambuk : 1. K10582 1662 15
2. K10582 1672 15

Makassar, 26 Januari 2018

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing:

Pembimbing I

Pembimbing II

Dr. Ir. Hj. Hafsah Nirwana, M.T.

Rizal Ahdiyati Duyo, S.T., M.T.

Mengetahui,

Ketua Jurusan Elektro

Dr. Umar Katu, S.T., M.T.
NBM : 990 410



UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK

GEDUNG MENARA IGRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221
 Website: www.umuh.ac.id, e-mail: umuh@umuh.ac.id
 Website: http://teknik.umuh.makassar.ac.id

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
PENGESAHAN

Skripsi atas nama Fitriantini Suaib dengan nomor induk Mahasiswa K10582 1662 15 dan Hadrayanti dengan nomor induk Mahasiswa K10582 1672 15 dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0001/SK-Y/20201/091004/2018, sebagai salah satu syarat guna memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Jurusan Teknik Elektro Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu tanggal 20 Januari 2018

Panitia Ujian Makassar, 09 Jumadil Awwal 1439 H
 26 Januari 2018 M

1. Pengawas Umum
 - a. Rektor Universitas Muhammadiyah Makassar
 Dr. H. Abdul Rahman Rahim, SE., MM. : 
 - b. Dekan Fakultas Teknik Universitas Hasanuddin
 Dr. -Ing. Ir. Wahyu H. Piarah, MSME : 
2. Penguji
 - a. Ketua : Dr. Ir. Zahir Zainuddin, M.Sc : 
 - b. Sekretaris : Andi Abd Halik Lateko Tj, S.T.,M.T : 
3. Anggota
 1. Anugrah, S.T.,M.M : 
 2. Rossy Timur Wahyuningsih, S.T.,M.T : 
 3. Rahmania, S.T.,M.T : 

Mengetahui

Pembimbing I

Pembimbing II



Dr. Ir. Hj. Hafsa Nirwana, M.T



Rizal Ahdiyati Duyo, S.T.,M.T

Dekan


Ir. Hamzah Al Imran, S.T., M.T.

NBM : 855 500

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT, karena rahmat dan hidayah-Nyalah sehingga penulis dapat menyusun skripsi ini, dan dapat kami selesaikan dengan baik.

Tugas akhir ini disusun sebagai salah satu persyaratan Akademik yang harus ditempuh dalam rangka menyelesaikan Program Studi pada Jurusan Teknik Telekomunikasi Fakultas Teknik Universitas Muhammadiyah Makassar. Adapun Judul tugas akhir kami adalah : “Perancangan Aplikasi Sistem *Helpdesk* Berbasis *Website* Pada Direktorat Teknologi Informasi Universitas Hasanuddin Makassar”

Penulis menyadari sepenuhnya bahwa di dalam penulisan skripsi ini masih terdapat kekurangan, hal ini disebabkan penulis sebagai manusia biasa tidak lepas dari kesalahan dan kekurangan, Oleh karena itu, penulis menerima dengan ikhlas dan senang hati segala koreksi serta perbaikan guna penyempurnaan tulisan ini agar kelak dapat bermanfaat.

Skripsi ini dapat terwujud berkat adanya bantuan, arahan, dan bimbingan dari berbagai pihak. Oleh karena itu dengan segala ketulusan dan kerendahan hati, kami mengucapkan terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Bapak Ir. Hamzah Al Imran, S.T., M.T. sebagai Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
2. Bapak Umar Katu, S.T.,M.T. sebagai Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Muhammadiyah Makassar.

3. Ibu Dr. Ir. Hafsah Nirwana, M.T. selaku Pembimbing I dan Bapak Rizal A Duyo, ST.,MT selaku Pembimbing II yang telah banyak meluangkan waktu dalam membimbing kami.
4. Bapak dan Ibu dosen serta staf pegawai pada Fakultas Teknik atas segala waktunya telah mendidik dan melayani penulis selama mengikuti proses belajar mengajar di Universitas Muhammadiyah Makassar.
5. Seluruh staf serta karyawan Universitas Hasanuddin Makassar atas kerjasamanya kepada penulis dalam penyusunan skripsi ini.
6. Ayahanda dan Ibunda tercinta, penulis mengucapkan terima kasih yang sebesar-besarnya atas segala limpahan kasih sayang, doa dan pengorbanannya terutama dalam bentuk materi dalam menyelesaikan kuliah.
7. Teman-teman seperjuangan kami, mahasiswa Fakultas Teknik terkhusus Angkatan Konversi Elektro dan Angkatan Nonreg 2014 yang dengan keakraban dan persaudarannya banyak membantu dalam menyelesaikan tugas akhir ini.

Semoga semua pihak tersebut di atas mendapat pahala yang berlipat ganda di sisi Allah SWT dan skripsi yang sederhana ini dapat bermanfaat bagi penulis, rekan-rekan, masyarakat serta bangsa dan negara. Amin.

Makassar, 2017

Penulis

Fitriantini Suaib¹, Hadrayanti²

¹Jurusan Teknik Elektro Fakultas Teknik Unismuh Makassar
email: fitrisuaib23@gmail.com

²Jurusan Teknik Elektro Fakultas Teknik Unismuh Makassar
email: adrayan.muh@gmail.com

ABSTRAK

Abstrak; Fitriantini dan Hadrayanti; (2017) Perancangan Aplikasi Sistem *Helpdesk* Berbasis *Website* Pada Direktorat Teknologi Informasi Universitas Hasanuddin Makassar dibimbing oleh Hafsah Nirwana (pembimbing I) dan Rizal A Duyo (pembimbing II). Universitas Hasanuddin, yang kemudian disingkat Unhas, merupakan sebuah perguruan tinggi negeri di kota Makassar. Dalam lingkup kerja Unhas, penggunaan teknologi informasi kepada semua staf sangat banyak manfaatnya. Dan sering kali ditemukan berbagai masalah dalam penggunaan peralatan tersebut, gangguan-gangguan yang dialami oleh staf Unhas apabila tidak ditangani akan menyebabkan kemunduran kerja. Oleh karena itu dibutuhkan divisi *helpdesk* yang berfungsi sebagai lapisan pertama yang harus dihubungi oleh *end user* bila mereka mendapatkan masalah/gangguan jaringan. Pelaporan tersebut dapat di tangani dengan cepat dengan aplikasi *website helpdesk*. Untuk mendukung proses tersebut penulis merancang aplikasi *helpdesk* yang memiliki fungsi penginputan keluhan staf, pencatatan proses penanganan pertanyaan dan keluhan dari staf, menyimpan dan mengolah informasi dari staf dalam *database* yang nanti nya bisa menjadi data untuk pengembangan dan perbaikan untuk mencegah gangguan terulang. Aplikasi *helpdesk* dibuat oleh penulis dengan tampilan yang menarik untuk memudahkan staf dalam pelaporan pertanyaan/keluhan secara *online*, Aplikasi *helpdesk* berbasis *web* dibuat dengan beberapa software pendukung yaitu salah satunya adalah PHP dan MySQL. PHP sebagai bahasa pemrograman yang mudah di mengerti oleh *programmer web* serta bersifat *Open Source* dan MySQL sebagai sistem manajemen *database* yang dapat diandalkan dan penggunaannya mudah untuk dipahami.

Kata kunci : *Helpdesk, Pelayanan Gangguan, PHP, MySQL*

DAFTAR ISI

HALAMAN JUDUL	i
KATA PENGANTAR	ii
ABSTRAK.....	v
DAFTAR ISI.....	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	3
1.4.1 Tujuan	3
1.4.2 Manfaat	4
BAB II TINJAUAN PUSTAKA	5
2.1 Konsep Dasar Perancangan.....	5
2.2 Pengertian Sistem	5
2.3 Definisi Aplikasi	5
2.4 Konsep Dasar <i>Helpdesk</i>	5
2.4.1 Kelebihan <i>Helpdesk</i>	7
2.4.2 Cara Kerja <i>Helpdesk</i>	7
2.5 Konsep Perancangan Aplikasi <i>Helpdesk</i>	9
2.6 Pengertian Web	9
2.6.1 Jenis-Jenis Web	10
2.6.1.1 Perbedaan Web Statis dan Web Dinamis	11
2.6.2 Cara Kerja Web	12

2.6.3	<i>Notepad ++</i>	12
2.6.3.1	Keunggulan <i>Notepad++</i> Dalam <i>Software Programming</i>	13
2.6.4	<i>Adobe Photoshop CS3 Portable</i>	14
2.6.5	<i>Apache Web Server</i>	15
2.6.6	<i>Hypertext Preprocessor (PHP)</i>	15
2.6.7	<i>MySQL</i>	17
2.7	Bagan Alir Dokumen.....	19
2.8	Pengertian Tipe Data	21
2.8.1	Contoh Tipe Data Dasar	22
2.8.2	Pengertian Variabel	25
2.8.3	Pengertian Operator	25
2.8.4	Operator Rasional	27
2.8.5	Operator Logika	28
2.9	Pemrograman Berorientasi Obyek	29
BAB III METODOLOGI PENELITIAN		32
3.1	Metode Pengumpulan Data.....	32
3.1.1	<i>Library Research</i>	32
3.1.2	<i>Field Research</i>	32
3.2	<i>Hardware</i> dan <i>Software</i> Perancangan.....	32
3.2.1	<i>Hardware</i>	32
3.2.2	<i>Software</i>	33
3.3	Metode Perancangan Sistem.....	33
3.3.1	Kerangka Berpikir	33
3.3.2	Kebutuhan Aplikasi	34
3.3.3	Desain Sistem	37
3.3.3.1	Bentuk Umum Perancangan Sistem	37
3.3.3.1.1	<i>Use Case Diagram</i>	37
3.3.3.2	Perancangan <i>Database Helpdesk</i>	38

3.3.3.2.1	<i>Class Diagram</i>	38
BAB IV HASIL DAN PEMBAHASAN		48
4.1	Halaman Utama	48
4.2	Halaman Menu Utama	50
4.3	<i>Form Ticket</i>	53
4.4	<i>Form Change Password</i>	54
4.5	Halaman <i>User</i>	54
4.5.1	Halaman <i>My Request</i>	54
4.5.2	Halaman <i>My Assigment</i>	55
4.5.3	Halaman <i>My Resolution</i>	56
4.5.4	Halaman <i>Waiting for Close</i>	56
4.5.5	Halaman <i>View All Open Ticket</i>	57
4.6	Halaman Admin	58
4.6.1	<i>User list</i>	58
4.6.2	<i>Division List</i>	60
4.7	Halaman <i>Ticket Admin</i>	62
4.7.1	Halaman <i>List All Tickets</i>	62
4.7.2	Halaman SLA Setting	63
4.8	Halaman <i>Helpdesk News</i>	64
4.9	Halaman Statistic	65
	Halaman Popular Solution	66
BAB V PENUTUP		67
5.1	Kesimpulan	67
5.2	Saran	67
DAFTAR PUSTAKA		68

DAFTAR GAMBAR

Gambar 2.1	Proses Penanganan Keluhan Staf	8
Gambar 3.1	Kerangka Berpikir	34
Gambar 3.2	Alur Kerja <i>Helpdesk</i> DTI UH	37
Gambar 3.3	Tiga <i>Role</i> Akses <i>Helpdesk</i> Dti UH	38
Gambar 3.4	Asosiasi Hirarki <i>Class</i> Diagram <i>Helpdesk</i> Dti UH	40
Gambar 4.1	<i>Form Login</i>	49
Gambar 4.2	Halaman Menu Utama	51
Gambar 4.3	<i>Form Ticket</i>	53
Gambar 4.4	<i>Change Password</i>	54
Gambar 4.5	Halaman <i>My Request</i>	55
Gambar 4.6	Halaman <i>My Assigment</i>	56
Gambar 4.7	Halaman <i>My Resolution</i>	56
Gambar 4.8	Halaman <i>My Assigment</i>	57
Gambar 4.9	Halaman <i>View All Open Ticket</i>	58
Gambar 4.10	Halaman Admin	59
Gambar 4.11	Halaman <i>Add User List</i>	60
Gambar 4.12	Halaman <i>User List</i>	60
Gambar 4.13	Halaman <i>Edit User List</i>	61

Gambar 4.15	<i>Division List</i>	62
Gambar 4.16	<i>Form Add New Division</i>	62
Gambar 4.17	Halaman <i>Ticket Admin</i>	63
Gambar 4.18	Halaman List All Tickets	63
Gambar 4.19	Halaman SLA Setting	64
Gambar 4.20	Halaman <i>Helpdesk News</i>	65
Gambar 4.21	Halaman <i>Helpdesk Statistic</i>	65
Gambar 4.22	Halaman Pivot	65
Gambar 4.23	Halaman Popular Solution	66

DAFTAR TABEL

Tabel 2.1	Simbol-simbol Bagan Alir Dokumen	19
Tabel 2.2	Integer	22
Tabel 2.3	<i>Real</i>	22
Tabel 2.4	Operator Artimatik	26
Tabel 2.5	Operator <i>Assignment</i>	27
Tabel 2.6	Operator Relasional	28
Tabel 2.7	Operator Logika	28
Tabel 2.8	Operator AND dan OR	28
Tabel 3.1	<i>Class</i> diagram staf	39
Tabel 3.2	<i>Log_user</i>	41
Tabel 3.3	<i>News</i>	41
Tabel 3.4	SLA Level Tiket	42
Tabel 3.5	SLA	43
Tabel 3.6	<i>Tickets</i>	43
Tabel 3.7	<i>User</i>	45
Tabel 3.8	<i>Log_Tickets</i>	45
Tabel 3.9	<i>Log_Divison</i>	46
Tabel 4.1	Level	50

DAFTAR LAMPIRAN

Lampiran 1	<i>Script Form Login</i>	69
Lampiran 2	<i>Script Halaman Menu Utama</i>	69
Lampiran 3	<i>Script Form Ticket</i>	70
Lampiran 4	<i>Script Change Password</i>	72
Lampiran 5	<i>Script Halaman My Request</i>	73
Lampiran 6	<i>Script Halaman My Assigment</i>	74
Lampiran 7	<i>Script Halaman My Resolution</i>	75
Lampiran 8	<i>Script Halaman My Assigment</i>	76
Lampiran 9	<i>Script Halaman View All Open Ticket</i>	79
Lampiran 10	<i>Script Halaman Admin</i>	80
Lampiran 11	<i>Script Halaman Add User List</i>	82
Lampiran 12	<i>Script Halaman User List</i>	82
Lampiran 13	<i>Script Division List</i>	83
Lampiran 14	<i>Script Form Add New Division</i>	84
Lampiran 15	<i>Script Halaman Ticket Admin</i>	86

DAFTAR NOTASI DAN SINGKATAN

TIK	Teknologi Informasi dan Komunikasi
UH	Universitas Hasanuddin Makassar
DTI	Direktorat Teknologi Infomasi adalah pusat teknologi dan pembelajaran sistem informasi, jaringan komputer di Universitas Hasanuddin Makassar
XAMPP	perangkat lunak (<i>free software</i>), sebagai <i>server</i> yang berdiri sendiri (<i>localhost</i>), yang mendukung banyak sistem operasi, <i>Apache HTTP Server</i> , <i>MySQL database</i> , dan penerjemah bahasa yang ditulis dengan bahasa pemrograman <i>PHP dan Perl</i>
PHP	<i>Hypertext Preprocessor</i> adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam <i>HTML</i>
MySQL	sebuah perangkat lunak sistem manajemen basis data <i>SQL</i> (bahasa Inggris: <i>database management system</i>)
WWW	<i>World Wide Web</i> dikenal dengan <i>WEB</i> adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet
HTML	<i>HyperText Markup Language</i> adalah sebuah bahasa <i>pemrograman</i> standar yang digunakan untuk membuat sebuah halaman <i>web</i>
Apache	server web yang dapat dijalankan di banyak sistem operasi seperti (<i>Unix, BSD, Linux, Microsoft Windows dan Novell Netware</i> serta <i>platform</i> lainnya) yang berguna untuk melayani dan memfungsikan situs <i>web</i>
INT	<i>Integer</i> merupakan tipe data dasar berupa bilangan yang tidak mengandung pecahan desimal

Char	<i>Char</i> atau <i>Varchar</i> adalah tipe data untuk menampung data yang bertipe karakter atau alfabetik(a-z)
TEXT	tipe data yang dapat menampung semua tipe data
Input	data masukan
Output	data keluaran
<i>Hardware</i>	perangkat keras
<i>Software</i>	perangkat Lunak
HDD	<i>Hard Disk Drive</i> adalah sebuah komponen perangkat keras yang menyimpan data sekunder dan berisi piringan magnetis
DDR	<i>Double-Data-Rate</i> adalah jenis memori komputer yang sangat cepat
PBO	pemrograman berorientasi obyek merupakan sebuah metode yang digunakan untuk mendapatkan solusi dari suatu masalah melalui perspektif object
OOP	<i>object oriented programming</i> metode pemrograman yang berorientasi kepada objek

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Internet merupakan singkatan dari *interconnected networking* yang berarti jaringan komputer yang saling terhubung antara satu komputer dengan komputer yang lain yang membentuk sebuah jaringan komputer di seluruh dunia, sehingga dapat saling berinteraksi, berkomunikasi, saling bertukar informasi atau tukar menukar data.

Dalam bidang pendidikan, internet memberikan berjuta-juta manfaat untuk pengguna. Salah satunya pada Universitas Hasanuddin Makassar, penggunaan teknologi informasi dan komunikasi (TIK) di universitas ini sangat banyak dampaknya, seperti membantu pelaksanaan suatu pekerjaan bahkan hampir seluruh bidang di setiap divisi.

Dalam perjalanan penggunaan TIK sering muncul permasalahan-permasalahan yang jika tidak ditangani akan mengakibatkan adanya kemunduran kinerja dari bagian/seseorang yang menggunakan peralatan TIK untuk menunjang kerja mereka. Adapun masalah yang berkaitan dari TIK ini, seperti gangguan jaringan, kerusakan alat atau perangkat dari TIK

Dimana pengaduan dari staf tersebut masih di lakukan secara manual ke bagian IT terkait, sehingga pencatatan gangguan/kerusakan tersebut masih dilakukan secara manual. Sehingga suatu universitas yang sudah sedemikian kompleksnya dalam penggunaan peralatan TIK di instansinya maka diperlukan suatu mekanisme dan unit khusus yang menangani penggunaan dan permasalahan dalam hal penggunaan TIK tersebut. Pada Universitas Hasanuddin Makassar divisi yang menangani permasalahan jaringan disebut Direktorat Teknologi Infomasi (DTI). Dan divisi DTI tersebut membutuhkan tim *helpdesk* yang tujuan utamanya adalah menangani penyelesaian terhadap permasalahan yang muncul dari penggunaan suatu peralatan TIK tersebut sehingga mampu memperlancar kerja yang terkait.

Sehingga bertolak dari latar belakang di atas dan seiring dengan berkembangnya teknologi informasi maka penulis memiliki ide untuk melakukan perancangan dan pembuatan aplikasi untuk mengelola dan mencatat permasalahan pengguna terkait, maka dibutuhkan sistem *helpdesk* yang terintegrasi dengan baik sehingga pengaksesan data pada *helpdesk* dapat di lakukan dengan mudah dan cepat guna pengukuran tingkat masalah serta pengaksesan laporan oleh Direktur. Untuk mendukung kebutuhan tersebut maka penulis mengambil judul **“PERANCANGAN APLIKASI SISTEM HELPDESK BERBASIS WEBSITE PADA DIREKTORAT TEKNOLOGI INFORMASI UNIVERSITAS HASANUDDIN MAKASSAR”**

Adapun *software* yang digunakan dalam pembuatan sistem ini adalah *Notepad++*, *Adobe Photoshop CS3 Portable* untuk mengolah gambar. XAMPP versi 3.2.2 yang mencakup: *Apache* versi 2.4.17 untuk *web server*, PHP versi 5.5.35 untuk bahasa pemrograman, dan MySQL versi 5.0.11 untuk *database* nya.

1.2 Rumusan Masalah

Berdasarkan penjabaran latar belakang penulisan maka dapat dirumuskan bahwa masalah yang saat ini di hadapi adalah:

1. Bagaimana cara memudahkan tim *helpdesk* DTI UH dalam *management* penerimaan pelaporan pertanyaan / keluhan dari staf ?
2. Bagaimana cara mensimulasikan aplikasi *helpdesk*?

1.3 Batasan Masalah

1. Aplikasi ini dibangun hanya sebatas sistem informasi *helpdesk* sederhana.
2. Aplikasi ini dikembangkan meliputi *input problem*, *update problem*, *view report problem*.
3. Aplikasi *helpdesk* menggunakan pemrograman PHP dan MySQL sebagai *database*-nya dengan dukungan *web server Apache*.
4. Aplikasi *helpdesk* berjalan pada *browser*.
5. Keamanan sistem sebatas login berupa *username* dan *password*.

1.4 Tujuan Dan Manfaat

1.4.1 Tujuan

1. Merancang sistem *helpdesk* berbasis web yang sesuai untuk kebutuhan DTI UH melalui tampilan *interface* yang *user friendly*.

2. Agar semua aktivitas anggota *helpdesk* tercatat secara sistimatis dan bisa di-*review* kembali untuk keperluan-keperluan tertentu seperti penilaian tim, masalah yang sering terjadi dan solusi untuk setiap insiden yang terjadi.

1.4.2 Manfaat

1. Untuk memudahkan tim *helpdesk* DTI UH dalam menangani pertanyaan atau pun keluhan dari staf.
2. Menghasilkan suatu informasi gangguan yang cepat dan akurat dalam membantu dan memudahkan proses *helpdesk* atau proses penanganan pertanyaan atau pun keluhan dari staf.

BAB II

TINJAUAN PUSATAKA

2.1 Konsep Dasar Perancangan

Perancangan adalah strategi untuk memecahkan dan mengembangkan solusi terbaik bagi permasalahan itu termasuk bagaimana mengorganisasi sistem kedalam subsistem – subsistem, serta alokasi subsistem – subsistem ke komponen – komponen perangkat keras, perangkat lunak, serta prosedur – prosedur (Adi Nugroho, ST. MMSI, 2002:139).

2.2 Pengertian Sistem

Sistem adalah totalitas yang efisien dan efektif, terdiri dari bagian-bagian yang berstruktur dan berinteraksi secara teratur, kualitas dan saling berhubungan satu sama lain di dalam wadah (transformasi/ konversi) yang dipengaruhi oleh aspek-aspek lingkungan guna mencapai tujuan. Menurut Drs M.A. Makkassau

2.3 Definisi Aplikasi

Aplikasi merupakan suatu kelompok file (*form, class, report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, di mana ruang lingkup dari suatu aplikasi berbeda – beda dari satu perusahaan ke perusahaan lainnya (Harip Santoso, 2000:9).

2.4 Konsep Dasar *Helpdesk*

Beberapa definisi mengenai *helpdesk* yang terdapat dalam beberapa literatur:

- a. *Helpdesk* adalah nama yang umum digunakan untuk pusat bantuan terhadap end-user. Saat ini *helpdesk* makin dilihat sebagai satu kesatuan

dengan fungsi layanan dan bertanggung jawab untuk menjembatani sumber daya untuk menyelesaikan masalah dan memenuhi kepuasan user.

- b. *Helpdesk* adalah titik pusat pelaporan masalah dan selanjutnya dikelola atau dikoordinasi. Dari sudut pandang yang lebih luas, juga dilihat sebagai bagian utama dari fungsi layanan, bertanggung jawab menjembatani sumber daya untuk menyelesaikan masalah. Pengguna *helpdesk* bisa internal atau eksternal, sehingga menjadikannya sangat penting dalam hal kelancaran jalannya organisasi dan kualitas bantuan yang ditawarkan pelanggan.
- c. *Helpdesk* adalah lapisan pertama yang harus dihubungi oleh end user bila mereka mendapatkan masalah. *Helpdesk* akan berupaya menanganinya, tapi bila gagal akan mengirimkan ke lapisan yang lebih senior. Selama itu, *helpdesk* akan menjadi koordinator dari penanganan masalah. *End user* harus selalu menghubungi *helpdesk* saat meminta bantuan ataupun menanyakan progress permintaan bantuan mereka. *End User* dilarang untuk menghubungi secara langsung lapisan *support* yang lebih dalam (mem-*bypass helpdesk*).
- d. *Helpdesk* adalah titik utama dimana *client* dari IT akan pertama kali menghubungi divisi IT saat mempunyai pertanyaan atau masalah yang berhubungan dengan IT. *Helpdesk* membawa harga diri dan wibawa divisi IT saat berhubungan dengan client sehingga *Helpdesk* sangat mempengaruhi *customer experience*. *Helpdesk* menyimpan *database* dari masalah dan solusi yang muncul dari operasional IT sehari-hari. *Helpdesk* memfasilitasi komunikasi antara user dan bagian IT lainnya, merespon

complaint, dan membuat prioritas pengerjaan masalah. Karena merupakan titik pertama hubungan ke client, staf *helpdesk* harus mempunyai pengetahuan yang luas (meskipun tidak mendalam). Hal ini diperlukan agar sebuah masalah dapat segera dikategorikan dan diberikan pada tim solusi yang benar. *Helpdesk* haruslah menjadi tempat utama client pertama kali menghubungi divisi IT. Bila tidak, penanganan masalah menjadi tidak terkoordinasi dan pengetahuan menjadi hilang setelah solusi diimplementasikan. *Client* tidak diperkenankan untuk menghubungi divisi lain karena akan mengacaukan prioritas kerja.

2.4.1 Kelebihan *Helpdesk*

Helpdesk memiliki beberapa kelebihan, antara lain :

1. *Helpdesk* dapat memberikan solusi atas pertanyaan-pertanyaan maupun *complaint* yang masuk dalam waktu yang lebih singkat.
2. *Helpdesk* dapat mengecek status permasalahan yang ada dan mengatur pembagian kerja staf.
3. *Helpdesk* dapat meningkatkan efisiensi perusahaan dalam menangani pertanyaan dan complain dari pelanggan.
4. *Helpdesk* dapat memberikan laporan kerja perkembangan kinerja para staf kepada pimpinan.

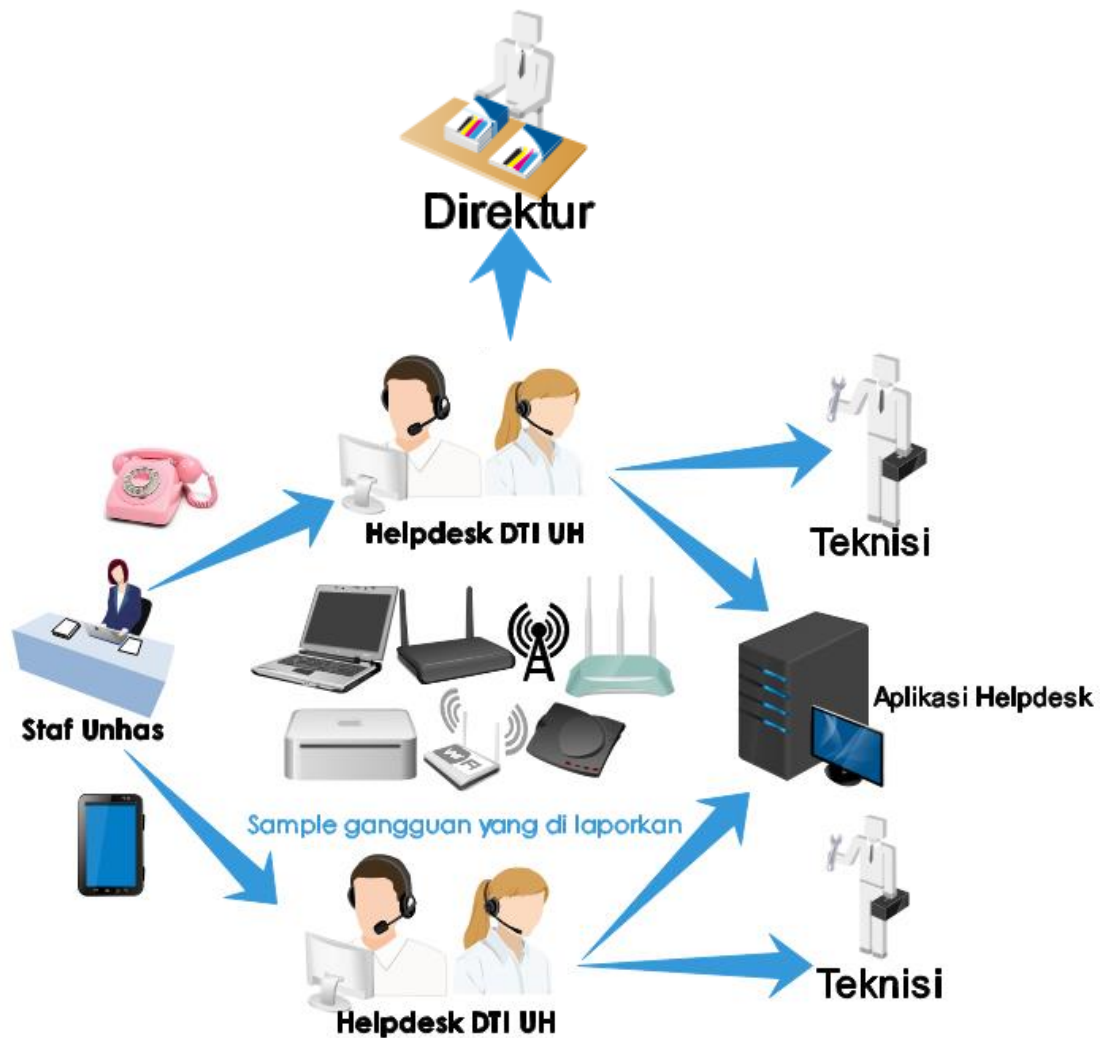
2.4.2 Cara Kerja *Helpdesk*

- Menampung, mengklasifikasikan dan memberikan prioritas terhadap *request ticket* melalui berbagai jalur seperti *E-Mail*, *Website*, atau *Telephone*.

- Melakukan *assignment request ticket* kepada unit-unit yang bertanggung jawab dan memiliki kapabilitas untuk menyelesaikan permasalahan tersebut. Jika diperlukan melakukan eskalasi *request ticket* ke tingkatan manajemen yang lebih tinggi.
- Monitoring durasi & status penyelesaian, dan melakukan *logging* terhadap seluruh tahapan tindakan yang telah diambil dan status penyelesaian *request ticket*.

Keluhan dapat disampaikan staf dengan melampirkan beberapa keterangan seperti berikut ini :

- Nama
- Nomor telepon
- E-mail
- Keluhan
- Jenis keluhan



Gambar 2.1 Proses Penanganan Keluhan Staf

2.5 Konsep Perancangan Aplikasi *Helpdesk*

Berdasarkan pengalaman penulis dalam dunia kerja sebagai *helpdesk*, dimana sering nya terjadi keluhan dan pelanggan melapor kan informasi tersebut melalui dari email, telepon, whatsapp, dan media lainnya. Kami ingin memberikan solusi terbaru dan berbeda dengan menawarkan pelaporan pertanyaan / keluhan staf melalui *website*, dimana staf dapat melaporkan dan mengetahui update pengerjaan tersebut.

2.6 Pengertian Web

World Wide Web atau WWW atau juga dikenal dengan *WEB* adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. *Web* ini menyediakan informasi bagi pemakai computer yang terhubung ke internet dari sekedar informasi “sampah” atau informasi yang tidak berguna sama sekali sampai informasi yang serius, dari informasi yang gratisan sampai informasi yang komersial. *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

2.6.1 Jenis-Jenis Web

Web dibagi menjadi 2 jenis, antara lain :

1. Pengertian Web Statis

Adalah sebuah web yang isinya tidak bisa dirubah-rubah dan tidak memiliki *database* sebagai tempat penyimpanan kontennya. Apabila ingin mengubah isi dari *website*, harus melakukan perubahan tulisan yang tertanam pada file program tersebut , sehingga penguasaan bahasa pemograman harus benar-benar dikuasai dengan baik. Contoh dari pemograman statis ini adalah HTML dan CSS
Contoh website statis adalah berisi profil perusahaan.

2. Pengertian Web Dinamis

Adalah sebuah web yang isinya dapat dirubah sewaktu-waktu tanpa melalui perubahan *code* pada file web, akan tetapi melalui halaman *administrator*. *Content* yang tampil dalam *web* dinamis tersimpan pada sebuah *database* sehingga orang awam pun bisa merubah isi *content web* tanpa penguasaan bahasa *pemograman web*.

Biasanya *web* ini tersusun dari bahasa pemograman yaitu HTML, CSS dan berhubungan dengan *database* MySQL. Contoh website dinamis adalah seperti Friendster, Multiply, dll.

2.6.1.1 Perbedaan Web Statis dan Web Dinamis

- Interaksi antara pengunjung dengan pemilik web :

Dalam *web* statis tidak dimungkinkan terjadinya interaksi antara pengunjung dengan pemilik *web*. Sementara dalam *web* dinamis terdapat interaksi antara pengunjung dengan pemilik *web* seperti memberikan komentar, transaksi *online*, forum dll

- Adanya *script language* yang digunakan :

Web statis hanya menggunakan *HTML* saja, sedangkan *web* dinamis menggunakan bahasa pemrograman web seperti *PHP* atau *ASP*.

- Penggunaan *Database* :

Web statis tidak menggunakan *database* karena tidak ada data yang perlu disimpan dan diproses. Sedangkan *web* dinamis menggunakan *database* seperti *mysql*, *oracle*, dll untuk menyimpan dan memproses data

- *Content* :

Content dalam web statis hanya diberikan oleh pemilik *web* dan jarang *diupdate*, sementara *content* dalam web dinamis bisa berasal dari pengunjung dan lebih sering *diupdate*.

2.6.2 Cara Kerja Web

Adapun cara kerja *web* adalah sebagai berikut:

- a. Informasi *web* disimpan dalam dokumen dalam bentuk halaman-halaman *web* atau *web page*.
- b. Halaman *web* tersebut disimpan dalam komputer *server web*.
- c. Sementara di pihak pemakai ada komputer yang bertindak sebagai komputer *client* dimana ditempatkan program untuk membaca halaman *web* yang ada di *server web* (*browser*).
- d. *Browser* membaca halaman *web* yang ada di *server web*.

2.6.3 Notepad++

Menurut Fitri (diakses pada tanggal 18 Februari 2017 di <https://bayudwiarta.wordpress.com/2014/10/16/fungsional-dari-notepad/>) *Notepad++* adalah suatu text editor yang berjalan pada *Operating System(OS)* Windows. *Notepad++* disini menggunakan komponen-komponen Scintilla agar dapat menampilkan dan menyunting text dan berkas source code berbagai bahasa pemrograman. *Notepad++* didistribusikan sebagai Free Software (gratis) Proyek ini dilayani oleh *Sourceforge.net* dengan telah diunduh lebih dari 27 juta kali dan dua kali memenangkan penghargaan *SourceForge Community Choice Award for Best*

Developer Tool. Pengertian lain Notepad++ adalah standar *programmer* editor. Sebenarnya selain notepad++, juga masih banyak editor yang lain. Anda boleh saja menggunakan sembarang editor, namun disarankan editor yang dipakai adalah editor yang mempunyai 3 fitur utama yaitu *Syntax Higlighting*, *Code folding*, dan *Line numbering*. *Notepad++* dipakai untuk editing file secara cepat (misal Bugfix) atau mengedit satu dua baris yang error.

2.6.3.1 Keunggulan Notepad++ Dalam Software Web Programming

- Simple, Ringan dan Cepat dibandingkan dengan text editor lainnya, notepad++ tidak perlu menunggu *loading opening library*, terlebih seperti pada software adobe dreamweaver dan eclipse apa lagi untuk PC / Laptop yang memiliki *specification* yang rendah.
- *Bracket Matching* atau bisa dibilang mengumpulkan yang sesuai (berpasangan),biasanya digunakan pada saat menuliskan syntax percabangan,perulangan dan bagian utama program. Fungsi ini berguna untuk kita jika menuliskan *block* program yang panjang dan terkadang kita lupa untuk menutup *block* program yang telah kita buat
- *Syntax Higlighting* tampilan *source code*,disini kita bisa melihat warna pada setiap fungsi dari *syntax*. Jadi kita tidak bingung menggolongkan kegunaan syntax yang kita tulis dan dapat dibaca dengan mudah. Contoh tulisan yang berwarna hijau biasanya terdapat pada statment jika kita menuliskan komentar pada sebuah program.

- *Syntax Folding* atau melipat *source code*, ini hampir sama seperti bracket matching sebelumnya. Jika bracket *matching* digunakan untuk menunjukkan awal dan akhir suatu block program. *Syntax Folding* disini tidak jauh beda tetapi, digunakan untuk menyembunyikan *block* program tertentu agar terlihat lebih ringkas tampilannya agar *programmer* tidak perlu melihat seluruh syntaxnya apa lagi jika sudah sampai lebih 1000 baris lebih.
- *Quick Color Picker++* fungsi ini berguna pada saat kita menuliskan kode warna pada html atau pun CSS tetapi tidak harus menuliskan kodenya terlebih jika kita lupa kode pada warna yang kita inginkan, biasanya akan muncul kotak dengan banyak aneka warna yang kita tinggal pilih, setelah memilih nanti kode warna tersebut akan muncul.
- *FingerText*, biasa digunakan untuk menuliskan bahasa PHP di notepad++, fungsi ini berfungsi untuk memudahkan pengetikan syntax dengan kata tertentu sebagai pemicu/trigger dan menggantikannya dengan menekan tombol TAB. Contoh kata g akan berubah menjadi \$_GET.

2.6.4 Adobe Photoshop CS3 Portable

Adobe Photoshop adalah *software* pengolah gambar yang sangat powerfull dengan segala fasilitasnya. Hasil gambar olah dengan Adobe Photoshop ini banyak dilihat di berbagai website, brosur, koran, majalah, dan media lainnya.

2.6.5 Apache Web Server

Apache adalah sebuah HTTP *server*. *Apache* dibangun pertama kalinya berdasarkan pada kode – kode dan ide – ide yang terdapat pada HTTP *server* yang terkenal pada saat itu yaitu NCSA httpd 1.3, pada awal 1995. Pada saat ini *Apache* termasuk *web server* yang paling banyak digunakan, hal ini terjadi bukan semata – mata karena *Apache* diedarkan secara gratis. Namun juga karena kemampuannya, secara fungsional, tingkat efisiensinya dan kecepatannya yang dapat dibandingkan dengan *server – server* HTTP UNIX lainnya. Selain diedarkan secara gratis dan *open source*, pengembangan *Apache* juga dilakukan oleh banyak *programmer* secara sukarela.

2.6.6 Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) menawarkan solusi yang lebih luwes. Dengan PHP, *developer* tidak perlu lagi berurusan dengan dua buah *file* terpisah seperti itu. *Browser web* mengacu secara langsung ke *file* yang dituju, yang lalu dibaca oleh *server* sebagaimana *file* HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser web*, *server web* memeriksa isi *file* dan menentukan apakah ada kode di dalam *file* tersebut yang harus dieksekusi. Bila ada, kode – kode tersebut akan dieksekusi. Hasilnya dimasukkan ke dalam dokumen yang sama. *Server web* bekerja secara langsung terhadap *file* yang bersangkutan, tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode dieksekusi di *server* (oleh karena itu disebut *server – side script*). PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan – kelebihannya, yaitu :

1. *Script* (kode program) terintegrasi dengan *file Hypertext Markup Language (HTML)*, sehingga *developer* biasa berkonsentrasi langsung pada penampilan dokumen webnya.
2. PHP diterbitkan secara gratis (*Open Source*).
3. PHP dapat berjalan pada sistem operasi yang berbeda.
4. Tidak ada proses *compiling* dan *linking*.
5. Berorientasi obyek (*Object Oriented*).
6. Sintaksis pemrogramannya mudah dipelajari, sangat menyerupai C dan Perl.
7. Integrasi yang sangat luas ke berbagai *server database*.

Menulis *web* yang terhubung ke *database* menjadi sangat sederhana.

Database yang didukung oleh : PHP, Oracle, Sybase, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, Informix, dBase, UNIX dbm.

PHP tidak terbatas untuk hanya menghasilkan keluaran HTML. Ia juga bisa digunakan untuk menghasilkan GIF, atau bahkan sumber *audio* yang dinamis dengan bantuan *Javascript*.

Dalam cara kerjanya PHP, yaitu pertama *client web browser* atau pengguna memakai komputer kemudian pengguna tersebut menjalankan *file* PHP itu di *web browser* atau yang biasa di sebut *Browser* saja dan kemudian *file* PHP itu di kirim ke *web server*. *Web server* mengirimkannya lagi ke *Engine* PHP atau mesin PHP dan di dalam mesin PHP itu diproses dan setelah diproses oleh mesin PHP maka akan berbentuk *file* HTML, dan *file* HTML ini akan dikirimkan ke *web server* dan *web server* akan memberikan ke pengguna.

2.6.7 MySQL

MySQL adalah suatu pemrograman / sistem manajemen *database* (kumpulan data yang terstruktur) yang menggunakan basis bahasa *Structured Query Language (SQL)*. *SQL* merupakan bahasa *query* yang terstandarisasi secara international untuk mengakses *database relational*. *MySQL* merupakan sistem manajemen *database* yang dapat diandalkan dan penggunaannya mudah untuk dipahami. *MySQL* didesain untuk menangani *database* yang besar dengan cepat, memiliki tingkat keamanan dan konektifitas yang tinggi.

Kehandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query *MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*. Selain itu *MySQL* juga memiliki beberapa keistimewaan, antara lain :

- *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac Os X Server*, *Solaris*, *Amiga*, dan masih banyak lagi.

- *Open Source*

MySQL didistribusikan secara open source (gratis), dibawah lisensi *GPL* sehingga dapat digunakan secara cuma-cuma.

- *Multiuser*

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

- *Performance tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

- *Column types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

- *Command dan functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

- *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta *password* terenkripsi.

- *Scalability dan limits*

MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

- *Connectivity*

MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

- *Interface*

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

- *Clients dan tools*

MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data.

- Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel.

2.7 Bagan Alir Dokumen


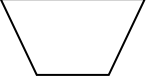


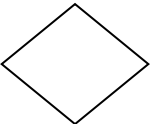


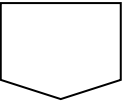
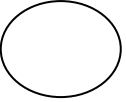
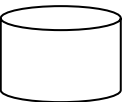
Bagan Alir dokumen (*Document Flowchart*) atau disebut juga bagan alir formulir (*Form Flowchart*) atau *Paperwork Flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

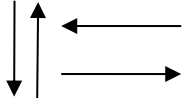

Suatu proses sistem apabila diuraikan menggunakan bagan diagram alir harus dimulai dan diakhiri dengan menggunakan simbol kegiatan awal dan akhir serta didalamnya terdapat aliran proses dan simbol-simbol proses yang terkait dengan kegiatan suatu program

Simbol-simbol yang biasa digunakan dalam bagan alir terlihat seperti pada Tabel di bawah ini.

Tabel 2.1 Simbol-simbol Bagan Alir Dokumen

No	Simbol	Keterangan
----	--------	------------

1		Simbol Dokumen, menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer.
2		Simbol Manual, menunjukkan kegiatan manual atau menyatakan tindakan atau proses yang tidak dilakukan oleh computer.
3		Simbol Terminal, menunjukkan kegiatan awal atau akhir suatu program.
4		Simbol Proses, menunjukkan kegiatan proses dari operasi program computer.
5		Simbol <i>decision</i> atau logika, menyatakan suatu kondisi yang akan menyebabkan dua kemungkinan jawaban, ya/tidak, benar/salah, true/false.
6		Simbol <i>Input-Output</i> , menyatakan proses <i>input</i> atau proses <i>output</i> tanpa tergantung dari jenis media <i>input</i> atau <i>output</i> .
7		Simbol Proses, digunakan untuk menunjukan suatu operasi yang rinciannya ditunjukan di tempat lain.
8		Simbol <i>OffLine Conector</i> , menyatakan sambungan suatu proses ke proses lainnya dalam halaman atau lembar yang berbeda.
9		Simbol <i>Conector</i> , menunjukkan penghubung dari suatu proses ke proses lainnya dari halaman/lembar yang sama
10		Menunjukkan <i>input</i> atau <i>output</i> menggunakan harddisk (penyimpanan/pengambilan data ke/dari disket)

11		Simbol Arus Data atau <i>Flow</i> , menyatakan jalannya suatu proses
12		File non-komputer yang diarsip untuk angka (<i>numeric</i>)

(Sumber : Jogiyanto H. M (2008 : 796)

2.8 Pengertian Tipe Data

Sebuah bahasa pemrograman tidak terlepas pada Tipe Data, karena memiliki fungsi yang sangat penting yaitu untuk menyatakan jenis nilai yang dimiliki oleh sebuah variabel. Selain itu, bahasa pemrograman juga membutuhkan Variabel yang berfungsi untuk menyimpan nilai baik itu huruf, karakter atau angka dan Operator yang biasanya di identikan dengan perhitungan yang bermain pada angka di tipe data integer atau yang memiliki nilai angka bulat.

Tipe data adalah himpunan nilai yang dapat dimiliki oleh sebuah data. Tipe data menentukan apakah sebuah nilai dapat dimiliki sebuah data atau tidak, serta operasi apa yang dapat dilakukan pada data tersebut. Contoh tipe data dalam dunia nyata adalah bilangan bulat.

Dalam sebuah program, setiap variabel dan konstanta memiliki tipe data yang harus dideklarasikan di awal program. Pendeklarasi tipe data tersebut bertujuan untuk menentukan besarnya tempat dalam memori yang akan digunakan untuk menyimpan data pada tersebut saat program dijalankan.

Tipe data dasar adalah tipe data yang dapat langsung digunakan. Secara umum terdapat 2 tipe data dasar, yaitu numerik dan kategorik. Tipe data

numerik terdiri atas angka/ kumpulan angka serta dapat mengalami operasi perhitungan, sedangkan tipe data kategorik dapat berupa angka maupun huruf namun tidak dapat mengalami operasi perhitungan.

2.8.1 Contoh Tipe Data Dasar :

Berikut adalah contoh dari tipe data, antara lain :

1. *Integer* (bilangan bulat)

Integer adalah tipe data dasar berupa bilangan yang tidak mengandung pecahan desimal. Tipe data ini juga memiliki urutan, sehingga dapat dibandingkan satu dengan lainnya.

Contoh integer:

- 2
- 5
- -10
- 135
- 2008

Tabel 2.2 Integer

TUPE	UKURAN	NILAI
Short	8 bit	-127 sampai 127
Int	16 bit	-32768 sampai 32767
Long	32 bit	-2147483648 sampai 2147483648

Secara teoritis, tipe data integer tidak memiliki batasan, yaitu dari minus tak hingga hingga plus tak hingga. Namun dalam pemrograman yang menggunakan bahasa pemrograman C++, secara umum dikenal beberapa macam tipe data integer, yaitu:

2. *Real* (bilangan riil)

Real adalah tipe data dasar berupa bilangan yang memiliki pecahan desimal. Dalam pemrograman, nilai dengan tipe data ini harus ditulis

dengan sebuah titik sebagai pemisah bilangan utuh dan bilangan pecahannya.

Tabel 2.3 Real

TIPE	UKURAN	NILAI
Float	32 bit	2.9×10^{-39} sampai 1.7×10^{38}
Double	48 bit	5.0×10^{-324} sampai 1.7×10^{308}

Tipe data ini digunakan untuk perhitungan yang melibatkan bilangan pecahan, seperti perhitungan kosinus, akar persamaan, dan sebagainya. Tipe data ini juga memiliki urutan, sehingga dapat dibandingkan satu dengan lainnya.

Contoh real:

- 0.5
- 0.17
- -3.465
- 92.0
- 4.3000+E9

Secara teoritis, tipe data real juga tidak memiliki batasan, yaitu dari minus tak hingga hingga plus tak hingga. Namun dalam pemrograman, secara umum dikenal beberapa macam tipe data real, yaitu:

Nilai pada tabel diatas berbeda dengan nilai yang ada pada tabel tipe data integer, pada tabel diatas nilai untuk tipe data merupakan tingkat ketelitian untuk masing-masing tipe data, bukan berdasarkan rentang nilai.

3. *Char* (karakter)

Char adalah tipe data dasar yang terdiri atas satu buah angka, huruf, tanda baca atau karakter khusus. Untuk menyimpan sebuah karakter, diperlukan 1 byte atau 8 bit tempat didalam memori.

Dalam sebuah program, penulisan tipe data char diawali dan diakhiri dengan tanda kutip ganda. Selain itu, terdapat sebuah karakter kosong yang disebut dengan null atau nil dan dituliskan sebagai "".

Contoh char :

- "5"
- "A"
- "?"
- "+"
- "\$"

Perhatikan bahwa 5 adalah integer sedangkan "5" adalah char.

4. *String*

String adalah tipe data dasar yang berupa kumpulan karakter dengan panjang tertentu. Meskipun berupa kumpulan karakter, karena tipe data string sering digunakan dalam pemrograman, string dianggap sebagai tipe data dasar.

Untuk penyimpanan string didalam memori, dibutuhkan 1 byte untuk tiap karakternya. Serupa dengan penulisan karakter, penulisan sebuah string juga harus diawali dan diakhiri dengan tanda petik ganda. String juga mengenal null yang dituliskan dengan "".

Contoh string:

- "BANDUNG"
- "Politeknik Telkom Bandung"
- "ABC3456"
- "Lucu"
- "30202001"
- "z"

Perhatikan bahwa sebuah karakter tunggal ("z") juga merupakan string.

2.8.2 Pengertian Variabel

Variabel atau peubah adalah obyek yang nilainya dapat berubah-ubah dalam sebuah program. Pada saat sebuah variabel dideklarasikan, program "memesan" tempat dengan ukuran tertentu (sesuai tipe datanya) pada memori untuk menyimpan nilai dari variabel tersebut.

Pemrogram dapat memberikan nama pada sebuah variabel untuk mempermudah pemanggilan variabel tersebut di dalam program. Pada saat mendeklarasikan sebuah variabel, pemrogram harus menyebutkan nama variabel dan tipe data dari variabel tersebut. Dalam bentuk flowchart, deklarasi variabel digambarkan sebagai sebuah proses.

Misalnya sebagai berikut:

x : integer

nama : string

tinggiBadan: real

jenisKelamin : char

status : boolean

2.8.3 Pengertian Operator

Operator adalah pengendali operasi yang akan dilakukan pada beberapa operan sehingga membentuk sebuah ekspresi. Secara umum, dalam sebuah ekspresi terdapat sebuah operator yang diapit dua operan.

Contohnya pada ekspresi:

$x + y$

x dan y adalah operan, sedangkan

"+" adalah operatornya

Terdapat tiga macam operator yang biasa digunakan dalam pemrograman, yaitu:

1. Operator Aritmatik

Operator ini membentuk perhitungan aritmatik. Kedua operan dari operasi aritmatik ini dapat berupa nilai integer atau real.

Operator yang termasuk tipe ini adalah:

Tabel 2.4 Operator Artimatik

LAMBANG	DESKRIPSI	CONTOH
+	Penjumlahan	$x = y+z$
-	Pengurangan	$x = y-z$
*	Perkalian	$x = y*z$
/	Pembagian	$x = y/z$
%	Modulo	$x = y\%z$

2. Operator Assignment

Dalam pemrograman bahasa C++, Operator ini digunakan memasukan nilai kedalam sebuah variabel, tanpa menghilangkan atau mengosongkan nilai variabel sebelumnya. Contoh penggunaan operator ini adalah sebagai berikut :

Tabel 2.5 Operator Assignment

LAMBANG	DESKRIPSI	CONTOH
+=	Menambahkan	x += 1
--	Mengurangkan	x -= 1
*=	Mengalikan	x *= 2
/=	Membagikan	x /= 2
%=	Mem-mod	x %= 2

3. Increase and Decrease

Penulisan ini dilambangkan dengan ++ (Increase) dan -- (decrease). Operator ini berfungsi untuk menaikkan atau menurunkan satu satuan nilai pada sebuah variabel. Contoh penggunaannya adalah pada contoh dibawah ini :

..

a++;

a += 1;

a = a + 1;

...

2.8.4 Operator Relasional

Operator ini membandingkan dua operan dan hasilnya berupa nilai boolean (BENAR atau SALAH). Operasi relasional dapat dilakukan pada dua nilai dengan

tipe data yang sama: tipe data integer, riil, char, string, maupun boolean. Berikut ini adalah operator relasional:

Tabel 2.6 Operator Relasional

LAMBANG	DESKRIPSI	CONTOH
<code>==</code>	Sama dengan	<code>x == y</code>
<code>!=</code>	Tidak sama dengan	<code>x != y</code>
<code>></code>	Lebih dari	<code>x > y</code>
<code><</code>	Kurang dari	<code>x < y</code>
<code>>=</code>	Lebih dari atau sama dengan	<code>x >= y</code>
<code><=</code>	Kurang dari atau sama dengan	<code>x <= y</code>

2.8.5 Operator Logika

Operator logika adalah operator yang digunakan untuk mengkombinasikan hasil ekspresi yang mengandung operator relasional. Tiga macam operator logika adalah:

Tabel 2.7 Operator Logika

LAMBANG	DESKRIPSI	CONTOH
<code>&&</code>	And / dan	<code>x > 7 && x = y</code>
<code> </code>	Or / atau	<code>x != y x > 3</code>
<code>!</code>	Not / tidak	<code>! (x > y)</code>

Output dari penggunaan operator AND dan OR adalah sebagai berikut:

Tabel 2.8 Operator AND dan OR

EKSPRESI 1	EKSPRESI 2	CONTOH	
		AND	OR
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

Pola yang mudah untuk mengingat output kedua operator logika tersebut adalah:

True AND True = True, False OR False = False.

2.9 Pemrograman Berorientasi Obyek

OOP/PBO merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. *Object Oriented Programming* yang berarti pula Pemrograman Berorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970.

Pemrograman Berorientasi Objek (*Object Oriented Programming/OOP*) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan Object yang lain. *Pemrograman* berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum, yaitu:

- ***Attribute***

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas, Atribut dapat memiliki hak akses *private*, *public* maupun *protected*. Sebuah atribut yang dinyatakan sebagai *private* hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung.

Sebuah atribut yang dinyatakan sebagai *public* dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya. Sebuah atribut yang dinyatakan sebagai *protected* tidak dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

Atribut juga biasa disebut *state/ properties/ field* merupakan bagian yang dimiliki oleh sebuah *class* yang tidak melakukan operasi, tetapi kepadanya dilakukan operasi sehingga dapat merubah nilai dari *attribute* tersebut.

- ***Method***

Definisi dan Gambaran Umum *Method*

Method adalah fungsi atau prosedur yang dibuat oleh seorang programmer didalam suatu *Class*. Dengan kata lain, *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada pemrograman prosedural.

Pada sebuah *method* di dalam sebuah kelas juga memiliki izin akses seperti halnya atribut pada kelas, izin akses itu antara lain *private*, *public* dan

protected yang memiliki arti sama pada izin akses atribut yang telah dibahas sebelumnya. Sebuah kelas boleh memiliki lebih dari satu method dengan nama yang sama asalkan memiliki parameter masukan yang berbeda sehingga kompiler atau interpreter dapat mengenali method mana yang dipanggil.

- ***Class***

Merupakan model yang berisi kumpulan attribute dan method dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh class manusia memiliki attribute berat, tinggi, usia kemudian memiliki method makan, minum, tidur. Method dalam sebuah class dapat merubah attribute yang dimiliki oleh class tersebut. Sebuah class merupakan dasar dari modularitas dan struktur dalam pemrograman berorientasi object.

- ***Object***

Merupakan perwujudan dari *class*, setiap *object* akan mempunyai attribute dan *method* yang dimiliki oleh *class*-nya, contohnya: amir, ahmad, yani merupakan object dari *class* manusia. Setiap *object* dapat berinteraksi dengan *object* lainnya meskipun berasal dari *class* yang berbeda.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

3.1.1 Library Research

Library research atau penelitian kepustakaan yaitu mengumpulkan data dengan jalan membaca buku-buku yang berkaitan dengan penelitian, mengutip pendapat-pendapat para ahli dari buku-buku bacaan yang ada kaitannya dengan pembahasan penelitian ini, dan mengumpulkan artikel dari internet yang berhubungan dengan penelitian.

3.1.2 Field Research

Field research atau penelitian lapangan yaitu mengumpulkan data yang diperoleh melalui observasi dan survey langsung.

3.2 Hardware dan Software Perancangan

Dalam proses perancangan ini, diperlukan alat dan bahan yang dapat mendukung keberhasilan perancangan. Alat dan bahan perancangan yang digunakan adalah

3.2.1 Hardware:

- a. Notebook Dell Inspiron N4050 dengan spesifikasi :
 - 1) Intel® Pentium® CPU B940 @2.00 GHz (2CPUs)
 - 2) Intel® HD Graphics
 - 3) RAM 4 GB DDR 3

4) 466 GB HDD

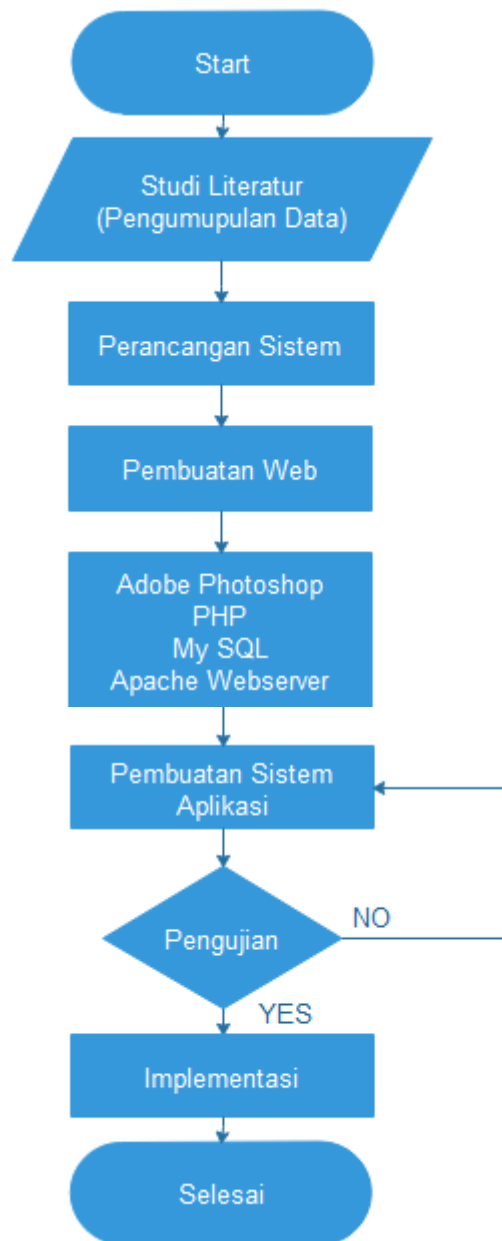
3.2.2 Software, terdiri atas :

1. *Micrsoft Windows 7 Ultimate 32 bit: Sistem Operasi*
2. *Adobe Photoshop CS3 : Web dan Image Editor*
3. *Notepad ++*
4. *PHP : Web editor*
5. *MySQL : Database server*
6. *Apache : Web Server*
7. *Mozilla firefox : Web browser*
8. *XAMPP : Web Server*

3.3 Metode Perancangan Sistem

3.3.1 Kerangka Berpikir

Untuk lebih memperjelas kerangka berpikir dalam laporan tugas akhir Kami, maka berikut ini digambarkan kerangka pikir dalam bentuk skema.



Gambar 3.1 Kerangka Berpikir

3.3.2 Kebutuhan Aplikasi

Kebutuhan ini diperoleh dengan melakukan komunikasi dengan pengguna.

Kebutuhan aplikasi sistem *helpdesk* secara *online* ini adalah sebagai berikut :

- a. **Website pengaduan online untuk staf Universitas Hasanuddin (UH)**

1. Staf UH melaporkan gangguan dengan berkomunikasi melalui *email*, *telephone*, aplikasi *chat* seperti *whatsapp* dengan tim *helpdesk*.
2. Tim *helpdesk* akan merespon dan menginput dalam aplikasi sistem *helpdesk*.

b. Aplikasi sistem *helpdesk* ketika menerima laporan gangguan

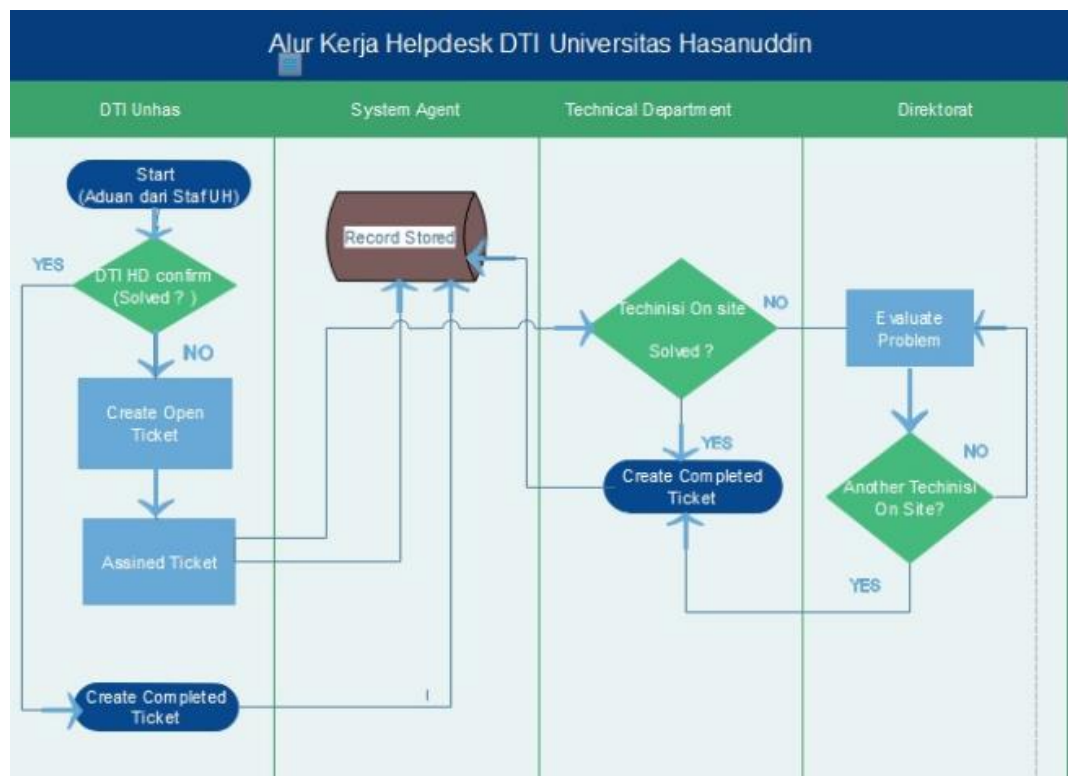
1. *Helpdesk* menerima pelaporan gangguan dan menginput pada aplikasi sistem *helpdesk*.
2. *Helpdesk* membantu menyelesaikan *problem* dan meng-*update problem* jika telah terselesaikan.
3. Jika gangguan tidak dapat diselesaikan oleh *helpdesk* maka *helpdesk* dapat mengirim *work order problem* kepada teknisi.
4. Teknisi menerima *work order problem* dari *helpdesk* dan segera menyelesaikan tugas tersebut.
5. Teknisi mengirim *problem solution* kepada *helpdesk* jika *problem* tersebut terselesaikan.
6. *Helpdesk* dapat meng-*update* kembali ke aplikasi *helpdesk* bahwa *complaint* tersebut telah terselesaikan.
7. Direktur Direktorat Teknologi Informasi UH dapat melihat laporan *helpdesk* kapan saja.

Sistem *helpdesk* berbasis web dikembangkan atas dasar fungsi operasi dari tim *helpdesk*, teknisi yang berunit kerja di Direktorat Teknologi Informasi dan Staf

unhas dalam mengelola dan melaporkan gangguan. Tim *helpdesk* dan teknisi adalah pengguna sistem yang menyimpan gangguan.

Untuk mempermudah Perancangan Aplikasi Sistem *Helpdesk* Berbasis *Website* Pada Direktorat Teknologi Informasi Universitas Hasanuddin Makassar ini, kami bagi menjadi 3 bagian yaitu :

1. User **Helpdesk** dapat membuat, meng-assign atau men-resolve ticket.
2. User **Teknisi** hanya dapat mengisi solusi dan melihat semua *ticket* yang ada.
3. Direktur/Manager **Direktorat Teknologi Informasi UH** dapat mengedit semua ticket serta melihat report helpdesk system.



Gambar 3.2 Alur Kerja *Helpdesk* DTI UH

3.3.3 Desain Sistem

3.3.3.1 Bentuk Umum Perancangan Sistem

Aplikasi *helpdesk* sistem ini akan dilengkapi dengan website statis atau biasa disebut *landing page* yang berisi informasi kinerja dari *helpdesk* dan *tools live chat* dari *Tawk*.

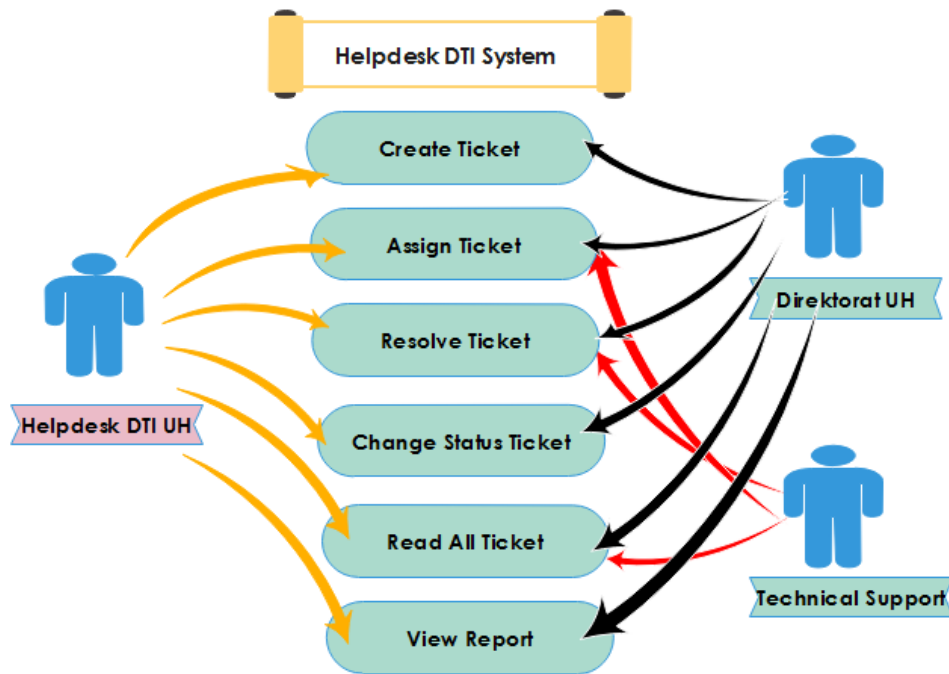
Secara keseluruhan sistem ini dirancang menggunakan metode pemrograman berorientasi obyek (PBO) atau kadang lebih dikenal dengan *object oriented programming* (OOP) dengan *Hypertext Preprocessor* (PHP) sebagai bahasa pemrogramannya yang dimana semua data dan fungsi dibungkus dalam class-class atau object.

Perancangan sistem menjelaskan mengenai aplikasi *helpdesk online* dalam menginterpretasikan masalah aplikasi desain *web* PHP dengan *database* MySQL. Perancangan ini terdiri dari :

3.3.3.1.1 *Use Case* Diagram

Use case merupakan gambaran skenario dari interaksi antara user dengan sistem. Sebuah diagram *use case* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi. Dalam aplikasi *Helpdesk* Sistem ini penulis akan melibatkan 4 aktor yaitu staf, *helpdesk*, teknisi dan direktorat. Jadi akan sediakan 3 role akses.

3 role use case diagram :



Gambar 3.3 Tiga Role Akses Helpdesk DTI UH

3.3.3.2 Perancangan Database Helpdesk

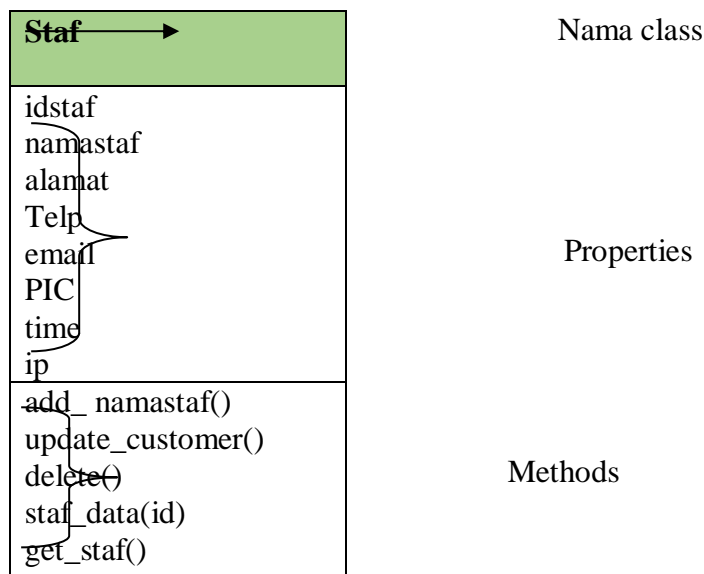
3.3.3.2.1 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class* diagram menggambarkan klasifikasi object-object yang terdapat dalam sistem serta hubungan antar object tersebut. Notasi class diagram terdiri atas tiga kotak, yaitu nama class, properties dan methods.

Properties dari sebuah *class* menggambarkan karakteristik atau sifat dari *object*. *properties* sebuah *class* merupakan data atau variabel dalam suatu program. Sementara *methods* menggambarkan perilaku atau operasi yang dapat dilakukan

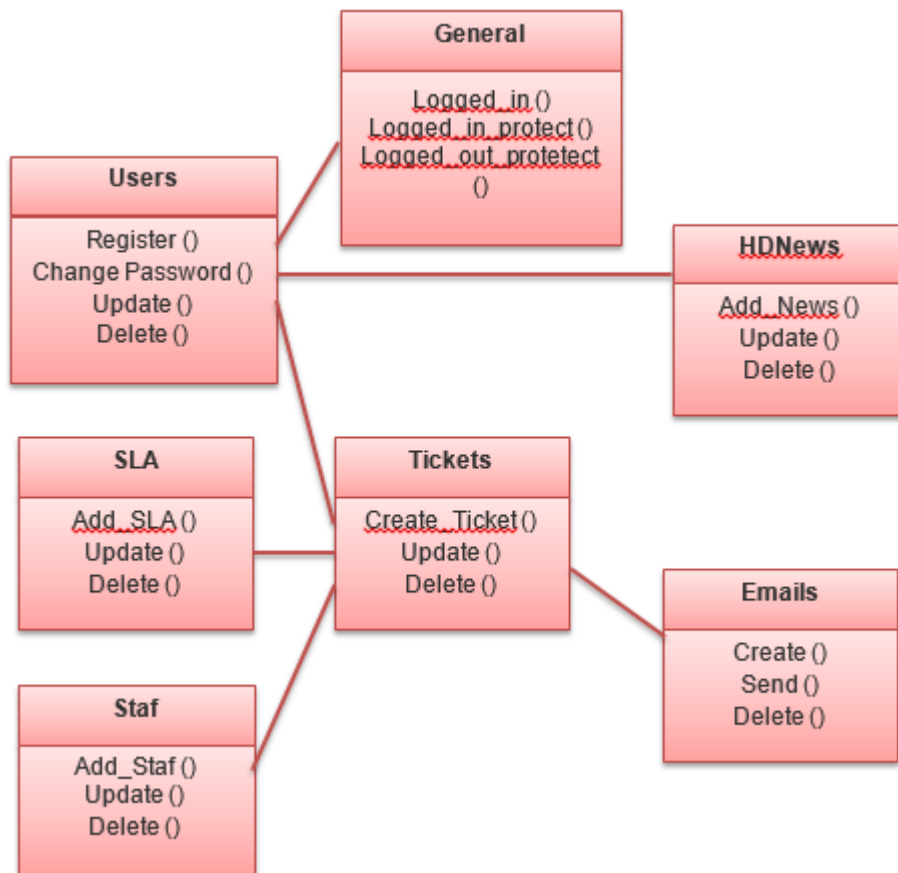
oleh *object* tersebut. *Methods* dalam program merupakan kumpulan *functions* dalam sebuah *class*.

Tabel 3.1 *Class* diagram staf



Tabel diatas merupakan *class* diagram staf yang terdiri dari 8 *properties* dan 5 *methods*. Ada 7 *class* yang akan kita buat dalam aplikasi *helpdesk*, yaitu class *General*, class *Users*, class *HDNews*, class *SLA*, class *Tickets*, class *Emails*, class *Staf*.

Masing-masing *class* tersebut memiliki hubungan (asosiasi) hingga membentuk hirarki *class* diagram pada diagram berikut :



Gambar 3.4 Asosiasi Hirarki *Class Diagram Helpdesk DTI UH*

Misalnya asosiasi antara *Tickets* dan *SLA*, dimana 1 ticket memiliki 1 *SLA*, dimana *SLA* dari 1 ticket akan di hubungkan dengan nomor ticketnya.

Hirarki *class diagram* dalam database ini terdiri dari :

- Tabel *general* yang merupakan halaman log in dari *helpdesk* saat meng-*input* gangguan.
- Tabel *log_users* digunakan untuk menyimpan informasi data penggunaan sistem. Tabel ini juga menyimpan data level dari user yang di gunakan sebagai rule akses otoritas halaman web. Berikut strukturnya :

Tabel 3.2 *Log_user*

Field	Type	Lenght	Index	A_I (Auto Increment)
iduser	INT	11	PRIMARY	✓
time	INT	11		
ip	VARCHAR	20		
browser	VARCHAR	100		
log	TEXT			

- Tabel *news* digunakan untuk menyimpan informasi berita dari tim helpdesk, misalnya jika terjadi gangguan internet, maka tim lain akan tahu sebelum mereka membuat *ticket* terkait masalah gangguan internet. Berikut struktur tabelnya.

Tabel 3.3 News

Field	Type	Lenght	Index	A_I (Auto Increment)
id	INT	11	PRIMARY	✓
newsdate	INT	11		
title	VARCHAR	80		
detail	TEXT			
createdby	VARCHAR	50		
createdon	INT	11		
expired	INT	11		

- Tabel *sla* digunakan untuk menyimpan informasi SLA terkait *priority ticket helpdesk*. Sebuah ticket umumnya memiliki salah satu dari empat jenis

priority yaitu *Low*, *Medium*, *High*, dan *Critical*, dimana setiap jenis priority memiliki waktu penyelesaian masalah (*resolution time*) yang berbeda. Perhatikan tabel SLA pada gambar.

Tabel 3.4 SLA Level Tiket

SLA Level	SLA	Response Time	Resolution Time	SLA Warning Time
1	Critical	1 Hours	6 Hours	4 Hours
2	High	1 Hours	24 Hours	20 Hours
3	Medium	1 Hours	72 Hours	50 Hours
4	Low	1 Hours	360 Hours	270 Hours

- *Response Time* adalah waktu yang dibutuhkan untuk menanggapi masalah yang diterima oleh team *helpdesk* dari staf.
- *Resolution Time* adalah waktu yang dibutuhkan untuk menyelesaikan masalah atau ticket yang masuk ke dalam *system*. Jika *ticket* itu memiliki *priority High*. Maka SLA untuk ticket itu adalah 24 jam, artinya *ticket* itu harus di resolved dalam waktu 24 jam *sejak ticket* di-*create*.
- *SLA Warning Time* digunakan oleh *system* untuk mengirimkan email notifikasi kepada *assignee* atau teknisi bahwa *ticketnya* sudah mendekati batas waktu SLA.

Berikut struktur tabel sla :

Tabel 3.5 SLA

Field	Type	Lenght	Index	A_I (Auto Increment)
slaid	INT	11	PRIMARY	✓
namasla	VARCHAR	30		
responsetime	INT	11		
resolutiontime	INT	11		
slawarning	INT	11		

- Tabel *tickets* digunakan untuk menyimpan informasi data *ticket*. Berikut struktur tabelnya :

Tabel 3.6 Tickets

Field	Type	Lenght	Index	A_I (Auto Increment)
id	INT	11	PRIMARY	✓
ticketnumber	VARCHAR	20		
sla	VARCHAR	10		
idstaf	VARCHAR	10		
reporteddate	INT	11		
reportedby	VARCHAR	50		
telp	VARCHAR	20		
email	VARCHAR	50		
problemsumary	VARCHAR	80		
problemdetail	TEXT	50		

ticketstatus	VARCHAR	21		
assignee	VARCHAR	50		
assigneddate	INT	11		
pendingby	VARCHAR	50		
pendingdate	INT	11		
resolution	TEXT			
resolvedby	VARCHAR	50		
resolveddate	INT	11		
closedby	VARCHAR	50		
closeddate	INT	11		
documentedby	INT	11		
documenteddate	INT	11		

Keterangan :

Field id dan ticketnumber pada dasarnya sama, dimana ticketnumber berisi data nomor ticket dengan format “id/SR/Bulan/Tahun”, contoh nomor ticket adalah 20/SR/Aug/2013. Format nomor ticket bisa diganti sesuai selera dan tidak mempengaruhi sistem, karena yang diambil sebagai key adalah field id, bukan field ticketnumber.

- Tabel *User* digunakan untuk menyimpan informasi data penggunaan sistem. Tabel ini juga menyimpan data level dari *user* yang di gunakan sebagai rule akses otoritas halaman web. Berikut struktur tabelnya.

Tabel 3.7 User

Field	Type	Lenght	Index	A_I (Auto Increment)
id	INT	11	PRIMARY	✓
username	VARCHAR	30		
password	VARCHAR	50		
level	VARCHAR	25		
fullname	VARCHAR	100		
email	VARCHAR	70		
Telp	VARCHAR	15		
Email_code	VARCHAR	100		
Time	INT	11		
Confirmed	INT	11		
ip	VARCHAR	32		

- Tabel *log_tickets* digunakan untuk menyimpan informasi perubahan *ticket* untuk audit *trail*, sehingga bisa diketahui siapa yg mengubah *ticket*, kapan perubahannya dan apa yang diubah. Berikut struktur tabelnya :

Tabel 3.8 Log_Tickets

Field	Type	Lenght	Index	A_I (Auto Increment)
id	INT	11	PRIMARY	✓
sla	VARCHAR	10		
reporteddate	INT	11		
reportedby	VARCHAR	50		
Telp	VARCHAR	20		
email	VARCHAR	50		

problemsumary	VARCHAR	80		
problemdetail	TEXT			
ticketstatus	VARCHAR	20		
assignee	VARCHAR	50		
assigneddate	INT	11		
pendingby	VARCHAR	50		
pendingdate	INT	11		
resolution	TEXT			
resolvedby	VARCHAR	50		
resolveddate	INT	11		
closedby	VARCHAR	50		
closeddate	INT	11		
changes	VARCHAR	80		
changeby	INT	11		
changedate	INT	11		

- Tabel *log_divisi* digunakan untuk menyimpan data informasi divisi unhas saat melaporkan gangguan, seperti nama, email, no telp.

Tabel 3.9 Log_staf

Field	Type	Lenght	Index	A I (Auto Increment)
Iddivisi	INT	11	PRIMARY	✓
namadivisi	VARCHAR	50		
Alamat	TEXT			
Telp	VARCHAR	20		
email	VARCHAR	50		

PIC	VARCHAR	50		
time	INT	11		
lp	VARCHAR	32		

BAB IV

HASIL DAN PEMBAHASAN


4.1 Halaman Utama

Layout halaman *login* terdiri dari judul aplikasi, logo lembaga, *form login* dan bagian *footer*.

Pada saat halaman *login* dibuka (*onload*), maka kursor *mouse* akan langsung berada di *field* User Name, sehingga *user* tidak perlu mengarahkan *mousenya* ke *field* User Name.

Error message ditempatkan pada bagian bawah *form login* dengan teks berwarna merah. Ada 4 pesan *error* yang akan ditampilkan:

- “Sorry, but we need your username and password. “Error ini ditampilkan jika user tidak mengisi *field username* atau *password*.”
- “Sorry, that username does not exist. Please try again. Error ini ditampilkan jika *username* yang di isi tidak terdaftar dalam *Helpdesk system*.”
- “Sorry, that username/password is invalid. Please try again.” Error ini ditampilkan jika *user* salah memasukkan *password*.”
- “Sorry, your account is locked. Please contact Administrator.” Error ini akan ditampilkan jika status *user* dalam kondisi disable (di *locked admin*).



Login Helpdesk
Direktorat Teknologi Informasi

Please Log in

User Name :

Password :

Login

Gambar 4.1 Form Login

Setiap *user* berhasil *login* dan *logout*, maka akan tersimpan dalam database.

Adapun tampilan dari halaman login dapat dilihat pada gambar 4.1

Adapun *Helpdesk system* ini memiliki 3 Role Level, yaitu:

1. Admin/*helpdesk*
2. Direktur/Manager
3. *User*/teknisi

User dengan *role level* Admin memiliki hak untuk mengakses semua menu, sementara *user* dengan *role level* Direktur memiliki hak untuk mengakses menu statistic dan *role level User* hanya dapat mengakses menu *My Ticket* dan menu *Knowledge Base*.

Berikut matriks *Role Level* dan menunya:

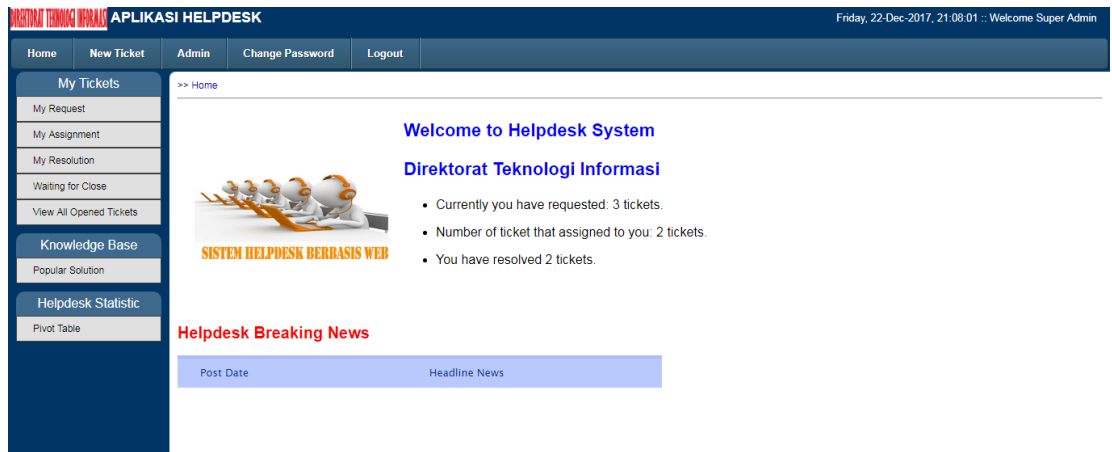
Tabel 4.1 Level

Akses Menu	User	Manager	Admin
New Ticket	Y	Y	Y
My Ticket	Y	Y	Y
Knowledge Base	Y	Y	Y
Helpdesk	T	T	Y
Statistic	T	T	Y
Data Master	T	T	Y
Ticket Admin	T	T	Y
System	T	T	Y

Tabel 4.1 diatas untuk “Y” menyatakan “Ya”, yang berarti aktif atau level *user* dapat mengakses menu tersebut dan untuk “T” menyatakan “Tidak” yang berarti tidak aktif atau tidak dapat mengakses menu tersebut.

4.2 Halaman Menu Utama

User akan masuk ke halaman utama (*home*) jika berhasil ke dalam *system*. Pada halaman utama ini terdapat *Frame Content* yang isinya berubah secara dinamis sesuai dengan menu yang dipilih oleh *user*. Berikut tampilan halaman utama:



Gambar 4.2 Halaman Menu Utama

- **Home**

Jika menu dipilih, maka *Frame Navigasi* (*navigator.php*) akan di reload begitu juga dengan *Frame Content* (*content.php*).

- **New Ticket**

Menu ini digunakan oleh *user* untuk membuat ticket baru. *Form ticket* (*ticketnew.php*) akan di-load di *Frame Content* menggantikan file.

- **Admin**

Menu Admin hanya terlihat oleh *user* dengan level 'Admin'. Jika menu ini dipilih, maka *Frame Navigasi* akan di-load dengan file *adminmenu.php* untuk menggantikan *navigator.php*

- **Change Password**

Menu ini digunakan oleh *user* untuk menggamti *password* loginnya. *Form Change Password* (*changepwd.php*) akan di-load dengan file *adminmenu.php*

- ***Logout***

Menu ini digunakan oleh user untuk keluar dari aplikasi *helpdesk system*. Setiap *user* melakukan *login* dan *logout* akan di catat di database pada *tabel user_log*.

Menu ***Home*** terdiri dari beberapa sub-menu, yaitu :

- ***My Ticket***

My Request, My Assigment, My Resolution, Waiting for Close.

- ***Knowledge Base***

Populer Solution.

- ***Helpdesk Statistic***

Pivot Table.

Menu ***Admin*** terdiri dari beberapa submenu, yaitu:

- ***Data Master***

User List, Divisi List.

- ***Ticket Admin***

List All Tickets, SLA Setting, dan Helpdesk News.

- ***System***

User Log, Email Log, dan Email Queue.

Adapun tampilan dari halaman utama (*home*) ini dapat dilihat pada gambar 4.2. Pada menu home, frame content berisi ringkasan jumlah ticket yang dimiliki oleh user yang login. Pada gambar tersebut terlihat user admin dengan keterangan ticket yang dimiliki.

Pada bagian bawahnya terdapat *view Helpdesk Breaking News* yang berisi berita terbaru dari team *helpdesk*. Tujuan dari *view* ini untuk memberitahukan kejadian umum atau masalah yang sedang terjadi, seperti misalnya gangguan internet global yang mengakibatkan banyak *user* tidak dapat mengakses email atau internet. Dengan demikian *user* lain yang *login* ke aplikasi *helpdesk* akan mengetahui issue umum yang sedang terjadi.

4.3 Form Ticket

User membuat *ticket* baru melalui menu *New Ticket*. *Form ticket* akan tampil dan *user* dapat mengisi dan men-submit-nya, seperti pada gambar 4.4. *User* dapat mengisi data-data pada form.

The screenshot shows the 'New Ticket' form in the 'APLIKASI HELPDESK' application. The form is titled 'New Ticket' and contains the following fields and options:

- Division***: A dropdown menu.
- Division Problem**: A text input field.
- Ticket No.**: A text input field with the placeholder 'New Ticket Number'.
- Reported Date***: A date picker showing '22-Dec-2017'.
- Reported By***: A text input field.
- Urgency (SLA)***: A dropdown menu.
- Problem Summary***: A text input field.
- Problem Detail***: A text area for detailed description.
- Assign to***: A dropdown menu.
- Telephone**: A text input field.
- Email**: A text input field.

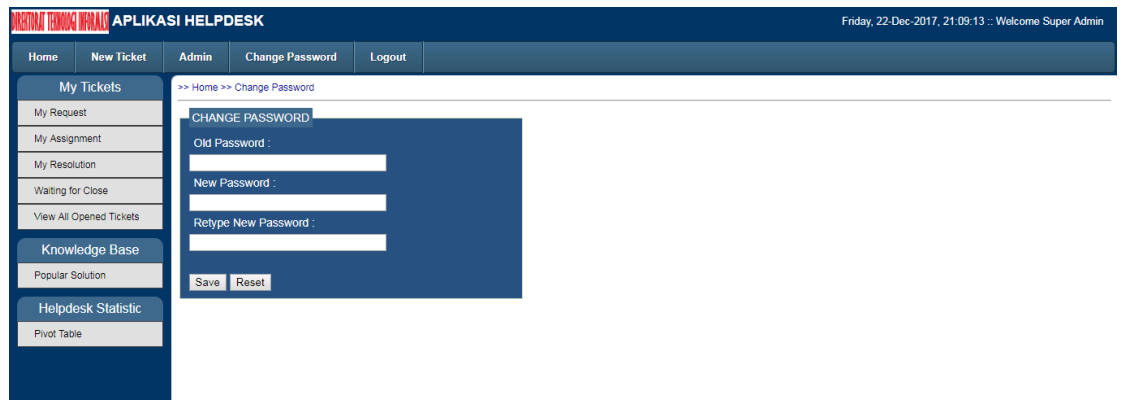
At the bottom of the form, there are two buttons: 'Submit' and 'Reset'.

Gambar 4.3 *Form Ticket*

Data yang tampil secara otomatis menggunakan teknik AJAX (Asynchronous Javascript and XML) jQuery dan JSON, berikut penggalan kode javascript dalam file “ticketnew.php”

4.4 Form Change Password

User dapat mengubah melalui menu *Change Password*. Halaman untuk change password dapat dilihat pada gambar 4.5.



Gambar 4.4 *Change Password*

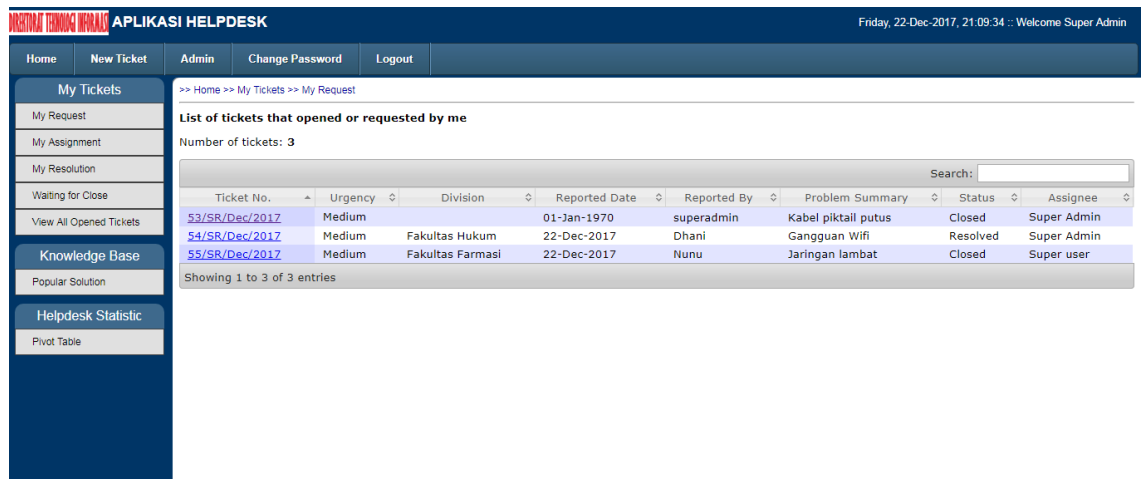
Untuk dapat mengubah *password*, *user* harus memasukkan *password* lamanya lalu memasukkan *password* yang baru sebanyak 2 kali, lalu klik tombol *Save* untuk menyimpan hasil pengubahannya.

4.5 Halaman *User*

Pada menu *Home* terdapat submenu *My Ticket*, *Knowledge Base* dan *Helpdesk* Statistik hanya dapat dilihat oleh *user* dengan level Admin dan Manager.

4.5.1 Halaman *My Request*

Menu *My Request* berisi *frame content* dari file *myticketbyrequester.php* yang berisi informasi pengaduan atau ticket yang diinput oleh *user*. Pada gambar 4.6 menampilkan dari halaman menu *My Request*.



APLIKASI HELPDESK Friday, 22-Dec-2017, 21:09:34 :: Welcome Super Admin

Home New Ticket Admin Change Password Logout

My Tickets >> Home >> My Tickets >> My Request

List of tickets that opened or requested by me

Number of tickets: 3

Ticket No.	Urgency	Division	Reported Date	Reported By	Problem Summary	Status	Assignee
53/SR/Dec/2017	Medium		01-Jan-1970	superadmin	Kabel piktail putus	Closed	Super Admin
54/SR/Dec/2017	Medium	Fakultas Hukum	22-Dec-2017	Dhani	Gangguan Wifi	Resolved	Super Admin
55/SR/Dec/2017	Medium	Fakultas Farmasi	22-Dec-2017	Nunu	Jaringan lambat	Closed	Super user

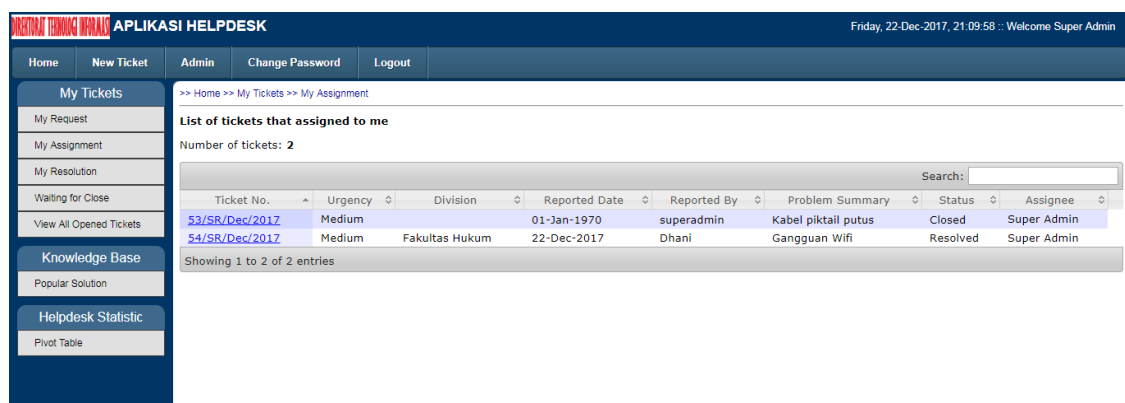
Showing 1 to 3 of 3 entries

Gambar 4.5 Halaman *My Request*

Daftar pengaduan ditampilkan dalam tabel menggunakan *library* jQuery datatables. Ada 3 file yang diperlukan oleh datatables, yaitu `jquery.dataTables.js`, `datatable.css` dan `jquery-ui.css`

4.5.2 Halaman *My Assigment*

Menu *My Assigment* hampir sama dengan menu *My Request*, menu ini menampilkan frame content yang berisi informasi gangguan atau *ticket* yang dinput user. Misalnya jika *user login*, maka menu *My Assigment* akan menampilkan *ticket* yang berada pada user tersebut. Berikut tampilan halaman *My Assigment*.



APLIKASI HELPDESK Friday, 22-Dec-2017, 21:09:58 :: Welcome Super Admin

Home New Ticket Admin Change Password Logout

My Tickets >> Home >> My Tickets >> My Assigment

List of tickets that assigned to me

Number of tickets: 2

Ticket No.	Urgency	Division	Reported Date	Reported By	Problem Summary	Status	Assignee
53/SR/Dec/2017	Medium		01-Jan-1970	superadmin	Kabel piktail putus	Closed	Super Admin
54/SR/Dec/2017	Medium	Fakultas Hukum	22-Dec-2017	Dhani	Gangguan Wifi	Resolved	Super Admin

Showing 1 to 2 of 2 entries

Gambar 4.6 Halaman *My Assignment*.

4.5.3 Halaman *My Resolution*

Halaman *My Resolution* berisi informasi tiket-tiket yang di-*resolved* oleh *user* yang *login*. Menu ini selain menampilkan tiket yang di resolved juga dapat di-edit sebelum statusnya menjadi *closed*. Adapun tampilannya sebagai berikut:

Gambar 4.7 Halaman *My Resolution*

APLIKASI HELPDESK Friday, 22-Dec-2017, 21:12:33 : Welcome Super Admin

Home New Ticket Admin Change Password Logout

My Tickets
My Request
My Assignment
My Resolution
Waiting for Close
View All Opened Tickets

Knowledge Base
Popular Solution
Helpdesk Statistic
Pivot Table

>> Home >> My Tickets >> My Resolution

List of tickets that resolved by me
Number of tickets: 2

Search:

Ticket No.	Urgency	Division	Reported Date	Reported By	Problem Summary	Status	Assignee
53/SR/Dec/2017	Medium		01-Jan-1970	superadmin	Kabel piktail putus	Closed	Super Admin
54/SR/Dec/2017	Medium	Fakultas Hukum	22-Dec-2017	Dhani	Gangguan Wifi	Resolved	Super Admin

Showing 1 to 2 of 2 entries

4.5.4 Halaman *Waiting for Close*

Jika menu *waiting for close* diklik, maka akan membuka *file myticketwaitforclosed.php* dan menampilkan pada *Frame Content*. Halaman ini berisi informasi tiket-tiket yang sudah di resolved namun belum di closed.

APLIKASI HELPDESK Friday, 22-Dec-2017, 21:12:58 : Welcome Super Admin

Home New Ticket Admin Change Password Logout

My Tickets
My Request
My Assignment
My Resolution
Waiting for Close
View All Opened Tickets

Knowledge Base
Popular Solution
Helpdesk Statistic
Pivot Table

>> Home >> My Tickets >> Waiting for Closed

List of tickets that resolved by me and waiting for Closed
Number of tickets: 1

Search:

Ticket No.	Urgency	Division	Reported Date	Reported By	Problem Summary	Status	Assignee
54/SR/Dec/2017	Medium	Fakultas Hukum	22-Dec-2017	Dhani	Gangguan Wifi	Resolved	Super Admin

Showing 1 to 1 of 1 entries

Gambar 4.8 Halaman *My Assignment*

4.5.5 Halaman *View All Open Ticket*

Jika Anda klik menu *view all open tiket*, maka akan membuka file *ticketlistuser.php* dan *menampilkan* data di *frame content*. Halaman *viewall open ticket* yang masih open atau belum di-*resolved*. Teknisi bisa mengambil tiket teknisi lain untuk di *resolved* tanpa harus menunggu tim *Helpdesk* untuk mengganti assignment, sehingga permohonan tiket bisa segera diselesaikan sebelum batas waktu yang ditentukan.

Adapun *tampilan* halaman *view all open ticket* dapat dilihat gambar berikut:

Setiap tiket akan diberikan warna sesuai dengan SLA-nya. Tiket yang baru dibuat akan berwarna hijau. Kemudian jika tiket tersebut sudah 75% dari batas waktu SLA-nya, maka akan berwarna kuning dan jika SLA tiket (misalkan dengan urgency medium) sudah lewat dari 72 jam sejak tiket dibuat dan belum juga di-*resolved*, maka akan berwarna merah.

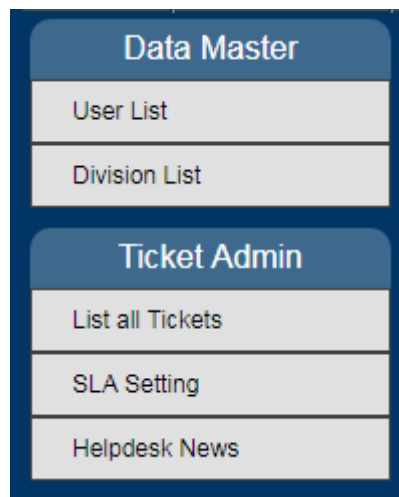
The screenshot shows the 'APLIKASI HELPDESK' interface. The top navigation bar includes 'Home', 'New Ticket', 'Admin', 'Change Password', and 'Logout'. The left sidebar contains 'My Tickets', 'Knowledge Base', and 'Helpdesk Statistic'. The main content area displays the 'List of Opened Tickets' with 7 entries. The tickets are color-coded by urgency: Medium (red), Critical (yellow), and High (green).

Ticket No.	Urgency	Division	Reported Date	Reported By	Problem Summary	Status	Assignee
46/SR/Dec/2017	Medium	Fakultas Kedokteran Gigi	02-Dec-2017 00:00	Sofyan	dafasgsg	Assigned	Fitri Suaib
48/SR/Dec/2017	Critical		02-Dec-2017 00:00	pitto	dadad	Resolved	Administrator
49/SR/Dec/2017	High	Tes	22-Dec-2017 00:00	pak udin	kabel putus	Assigned	Staff Helpdesk
50/SR/Dec/2017	Medium	Fakultas Hukum	22-Dec-2017 00:00	asdas	dadad	Assigned	Hadra
51/SR/Dec/2017	High	Fakultas Ilmu Budaya	22-Dec-2017 00:00	indra	bbbb	Assigned	Admin1
52/SR/Dec/2017	Medium	Fakultas Farmasi	22-Dec-2017 00:00	cc	dadadadacc	Assigned	Admin1
54/SR/Dec/2017	Medium	Fakultas Hukum	22-Dec-2017 00:00	Dhani	Gangguan Wifi	Resolved	Super Admin

Gambar 4.9 Halaman *View All Open Ticket*

4.6 Halaman Admin

Apabila menu Admin yang ada di *header* di-klik, maka akan tampil dua menu dan submenu didalamnya. Berikut gambarnya :



Gambar 4.10 Halaman Admin

4.6.1 User list

Pada halaman ini, jika menu *user list* di-klik, maka akan membuka file *userlist.php* dan akan menampilkan pada *Frame Content*.

Halaman *user list* menampilkan informasi user yang terdaftar di aplikasi helpdesk. Disini admin dapat melakukan *create*, *update* dan *delete user*, menentukan level user dan memblokir user. Untuk tampilan halaman *user list* dapat di lihat pada gambar 4.11

Untuk menambah *user* baru, admin dapat meng-klik tombol *Add New User*, maka akan tampil *form Add New User*.

Gambar 4.11 Halaman *Add User List*

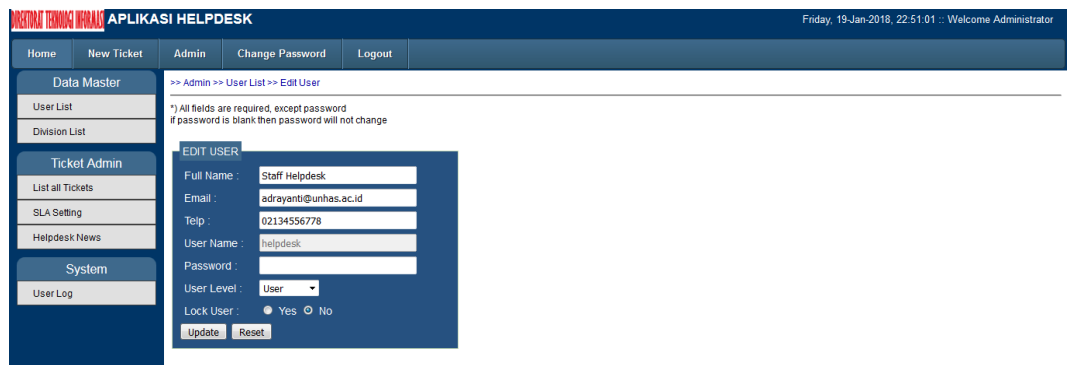
Semua *field* dalam form tersebut harus di-isi dan field password minimal 3 karakter. User level ada 3 pilihan, yaitu: *User*, *Direktur*, dan *Admin*.

User Name	Level	Full Name	Email	Telepon	Last Update	Locked	
admin	Admin	Administrator	hendra@kampushendra.com	08159123113	24-Jul-2013 16:17	No	del
adra	User	adrayan	adrayan.muh@gmail.com	085256755356	23-Feb-2017 10:14	No	del
hadra	User	Hadra	hadrayanti.muhammad@gmail.com	085256755356	28-Nov-2017 09:43	No	del
helpdesk	User	Staff Helpdesk	helpdesk@kampushendra.com	02134556778	08-Aug-2013 12:26	No	del
khadra	Manager	khadra	khadra.muhammad@gmail.com	085256755356	01-Mar-2017 14:26	No	del
manager1	Manager	Manager 1	manager1@kampushendra.com	123456789012345	08-Aug-2013 10:49	No	del
Pitto	User	Fitri Suaib	fitrisuaib@gmail.com	085256755356	02-Dec-2017 08:13	No	del

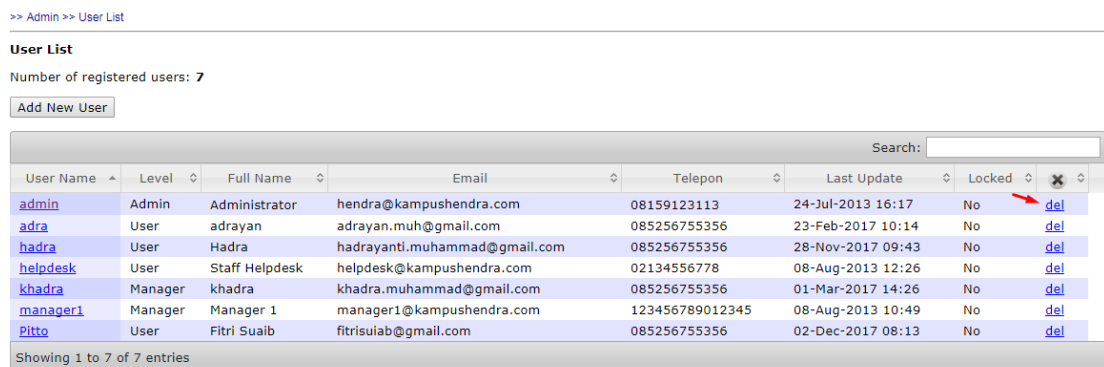
Showing 1 to 7 of 7 entries

Gambar 4.12 Halaman *User List*

untuk edit *user*, admin dapat mengklik pada user name dihalaman *user list* pada gambar berikut.

Gambar 4.13 Halaman *Edit User List*

sedangkan untuk menghapus user, admin dapat mengklik link del pada data yang ingin dihapus di halaman user list.

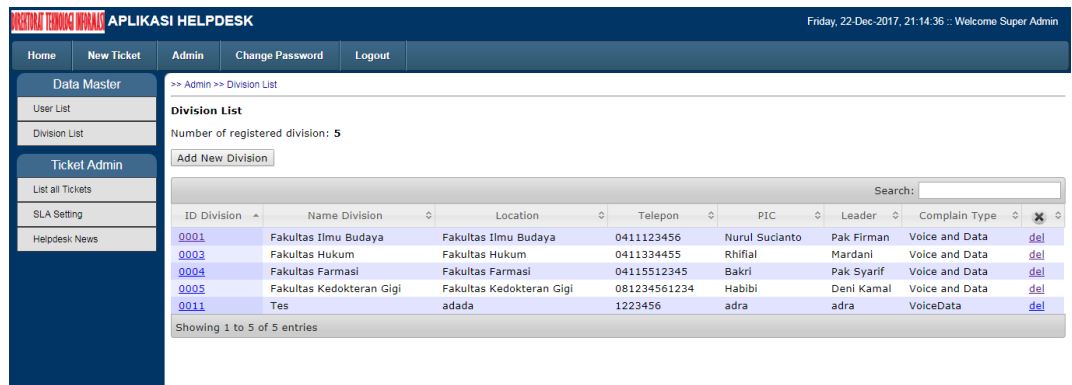


Gambar 4.14 Halaman Delete User List

4.6.2 *Division List*

Jika menu *division list* di-klik, maka akan membuka file *divisilist.php* dan akan menampilkan *Frame Content*.

Halaman *division list* menampilkan informasi *division list* yang akan di *support* didalam aplikasi *helpdesk*. Disini admin dapat *add new divison*, *update* dan *delete divison*.



APLIKASI HELPDESK Friday, 22-Dec-2017, 21:14:36 - Welcome Super Admin

Home New Ticket Admin Change Password Logout

Data Master
User List
Division List

Ticket Admin
List all Tickets
SLA Setting
Helpdesk News

>> Admin >> Division List

Division List
Number of registered division: 5

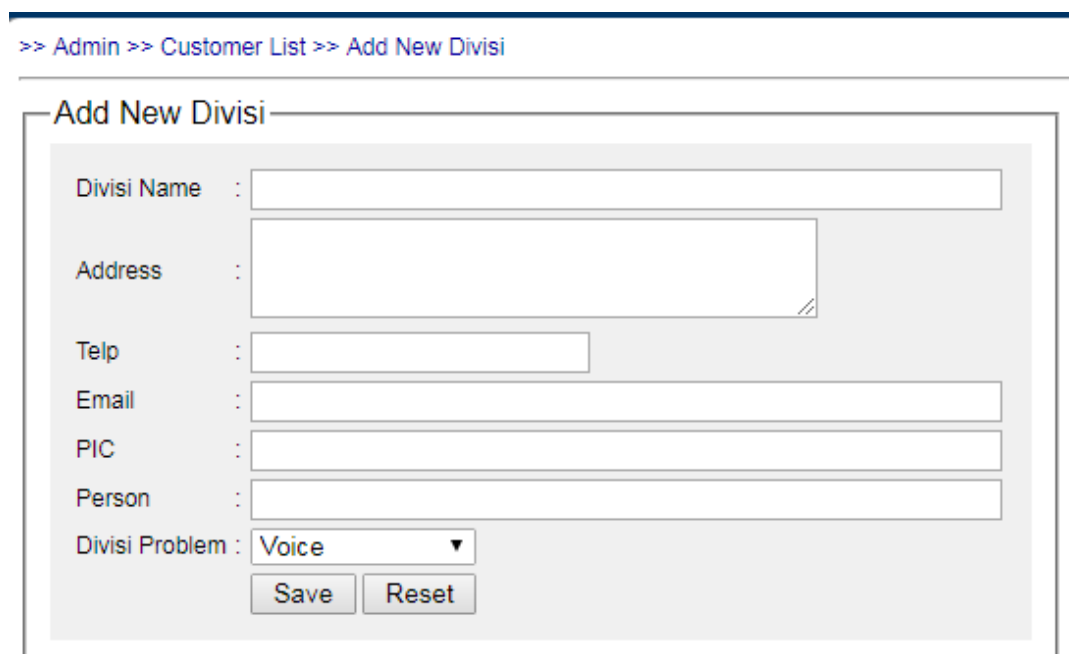
Add New Division

ID Division	Name Division	Location	Telepon	PIC	Leader	Complain Type	
0001	Fakultas Ilmu Budaya	Fakultas Ilmu Budaya	0411123456	Nurul Suclanto	Pak Firman	Voice and Data	del
0003	Fakultas Hukum	Fakultas Hukum	0411334455	Rhifial	Mardani	Voice and Data	del
0004	Fakultas Farmasi	Fakultas Farmasi	04115512345	Bakri	Pak Syarif	Voice and Data	del
0005	Fakultas Kedokteran Gigi	Fakultas Kedokteran Gigi	081234561234	Habibi	Deni Kamal	Voice and Data	del
0011	Tes	adada	1223456	adra	adra	VoiceData	del

Showing 1 to 5 of 5 entries

Gambar 4.15 *Division List*

Untuk menambah divisi baru, admin dapat mengklik tombol *Add New Division*, maka akan tampil form *Add New Division* seperti berikut :



>> Admin >> Customer List >> Add New Divisi

Add New Divisi

Divisi Name :

Address :

Telp :

Email :

PIC :

Person :

Divisi Problem : Voice ▼

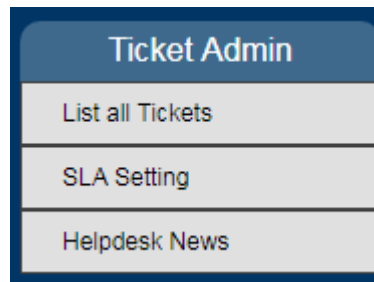
Save Reset

Gambar 4.16 *Form Add New Division*

Untuk edit division, admin dapat mengklik pada link *ID Division* di halaman *division list*.

4.7 Halaman *Ticket Admin*

Menu ticket admin terdiri dari tiga submenu, yaitu list all ticket, SLA setting, dan helpdesk news. Berikut gambarnya :



Gambar 4.17 Halaman *Ticket Admin*

4.7.1 Halaman *List All Tickets*

Jika admin mengklik menu *list all ticket*, akan membuka file `ticketlist.php` dan menampilkan pada *Frame Content*.

Berbeda dengan halaman *view all opened ticket* di halaman *home user*, dimana pada halaman admin ini terdapat fitur *delete* (`del`) pada daftar *ticket*, artinya admin memiliki hak untuk menghapus *ticket*. Berikut tampilan halaman list all ticket :

Ticket No.	Urgency	SLA Goal Time	Division	Reported Date	Documented Date	Problem Summary	Status	Assignee	
46/SR/Dec/2017	Medium	05-Dec-2017 08:34:35	Fakultas Kedokteran Gigi	02-Dec-2017 00:00:00	02-Dec-2017 08:34:35	dafsasgsg	Assigned	Fitri Suaib	del
48/SR/Dec/2017	Critical	02-Dec-2017 14:35:52		02-Dec-2017 00:00:00	02-Dec-2017 08:35:52	dadad	Resolved	Administrator	del
49/SR/Dec/2017	Critical	22-Dec-2017 23:19:02	Tes	22-Dec-2017 00:00:00	22-Dec-2017 17:19:02	kabel putus	Assigned	Staff Helpdesk	del
50/SR/Dec/2017	Medium	25-Dec-2017 17:19:33	Fakultas Hukum	22-Dec-2017 00:00:00	22-Dec-2017 17:19:33	dadad	Assigned	Hadra	del
51/SR/Dec/2017	High	23-Dec-2017 17:30:23	Fakultas Ilmu Budaya	22-Dec-2017 00:00:00	22-Dec-2017 17:30:23	bbbb	Assigned	Admin1	del

Gambar 4.18 Halaman *List All Tickets*

4.7.2 Halaman *SLA Setting*

Jika menu *SLA setting* di-klik, maka akan membuka file `slalist.php` dan menampilkan pada *frame content*.

Halaman SLA setting informasi level *urgency/priority/severity* (umumnya: *critical, high, medium,low*). Disamping itu juga terdapat informasi *response time, resolution time, dan SLA warning time* dari setiap levelnya.

Berikut tampilan SLA setting :

>> Admin >> SLA Setting

SLA Setting

Add New SLA

Search:

SLA Level ^	SLA ⇅	Response Time ⇅	Resolution Time ⇅	SLA Warning Time ⇅	✕ ⇅
1	Critical	1 Hours	6 Hours	4 Hours	del
2	High	1 Hours	24 Hours	20 Hours	del
3	Medium	1 Hours	72 Hours	50 Hours	del
4	Low	1 Hours	360 Hours	270 Hours	del

Showing 1 to 4 of 4 entries

Gambar 4.19 Halaman SLA Setting

Keterangan :

- *Response Time* : lamanya waktu yang disepakati untuk sebuah *ticket* dari status “*new*” menjadi “*assigned*”. Jika set *Response Time* = 1 jam, artinya sebuah *ticket* harus berubah statusnya dari “*new*” ke status “*assigned*” dalam waktu 1 jam. Dalam aplikasi *helpdesk* ini, status sementara “*new*” tidak dipakai, jadi ketika *helpdesk* membuat *ticket* baru, statusnya langsung “*assigned*”
- *Resolution Time*: lamanya waktu yang disepakati untuk sebuah *ticket* dari status “*assigned*” menjadi “*resolved*”. Misalnya untuk *ticket* “*critical*”, jika kita set *resolution time* = 6 jam maka artinya sebuah *ticket* harus berubah statusnya dari “*assigned*” ke status

“resolved” dalam waktu 6 jam. Perubahan status tersebut dilakukan oleh *assignee* (biasanya teknisi yang melakukan *ticket* tersebut).

- Untuk *ticket critical*, sistem mengimrimkan email kepada *assignee* 4 jam setelah *ticket* di create. Waktu tersebut bisa kita *setting* di *field* “SLA warning time”

4.8 Halaman *Helpdesk News*

Pada halaman *helpdesk news*, akan membuka file *hdnews.php* dan menampilkannya pada *frame content*.

Halaman *helpdesk news* seperti sekilas info yang berisi informasi tentang issue global yang mempengaruhi banyak sistem, seperti internet down atau aplikasi down. Tujuannya agar semua *user*, terutama tim *support* mengetahui berita tersebut, sehingga mencegah akan banyaknya *ticket* yang masuk ke *helpdesk*.

Tampilan halaman *helpdesk news* seperti gambar berikut :

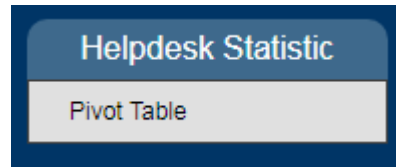
The screenshot shows a web interface for 'Helpdesk News'. At the top, there is a breadcrumb trail '>> Admin >> Helpdesk News' and a title 'Helpdesk News' with an 'Add News' button. Below this is a search bar and a table with columns: News Date, Headline News, Created By, Created on, News Expired, and a delete icon. The table contains one row with the following data: 02-Dec-2017, Selamat Datang, Administrator, 02-Dec-2017, 02-Dec-2017, and del. At the bottom of the table, it says 'Showing 1 to 1 of 1 entries'.

News Date	Headline News	Created By	Created on	News Expired	
02-Dec-2017	Selamat Datang	Administrator	02-Dec-2017	02-Dec-2017	del

Gambar 4.20 Halaman *Helpdesk News*

4.9 Halaman *Statistic*

Untuk halaman *statistic*, terdiri dari menu pivot table, berikut tampilan gambarnya :



Gambar 4.21 Halaman Helpdesk Statistic

Jika menu pivot table di-klik, maka akan membuka file `hdpivot.php` terletak di folder pivot dan akan menampilkannya pada *Frame Content*. Pada halaman pivot table akan menampilkan table pivot yang berisi rekapitulasi data jumlah *ticket* yang ada didalam sistem. Halaman ini hanya tersedia untuk user dengan role direktur dan admin.

Tampilan menu pivot table dibuat dengan bantuan free library Javascipt bernama `pivot.js`.

 A screenshot of a web application interface showing a pivot table. The page title is 'APLIKASI HELPDESK' and the date is 'Friday, 22-Dec-2017, 21:17:53'. The page has a navigation menu with 'Home', 'New Ticket', 'Admin', 'Change Password', and 'Logout'. The main content area is titled 'Helpdesk Statistic Pivot Table'. It features a table with columns for 'Company', 'Status', 'Assigned', 'Closed', 'Resolved', and 'Totals'. The data is as follows:

Company	Status			Totals
	Assigned	Closed	Resolved	
Fakultas Farmasi	1	1		2
Fakultas Hukum	1		1	2
Fakultas Ilmu Budaya	1			1
Fakultas Kedokteran Gigi	1			1
Tes	1			1
Totals	5	2	2	9

Gambar 4.22 Halaman Pivot

4.10 Halaman *Popular Solution*

Pada halaman *popular solution* terdapat pada submenu *knowledge base*, jika halaman ini di-klik maka akan membuka file *popularsolutin.php* dan menampilkannya pada *Frame Content*.

Halaman *popular solution* berisi informasi ticket-ticket yang sudah *closed*. User dapat secara langsung melihat problem solution pada table yang ada pada halaman tersebut tanpa harus mengisi periode. Berikut tampilan halaman popular solution.

The screenshot shows the 'Popular Solution' page in a helpdesk application. The page has a dark blue header with the text 'APLIKASI HELPDESK' and 'Friday, 19-Jan-2018, 22:47:03 :: Welcome Administrator'. Below the header is a navigation menu with 'Home', 'New Ticket', 'Admin', 'Change Password', and 'Logout'. On the left side, there is a sidebar with 'My Tickets' (My Request, My Assignment, My Resolution, Waiting for Close, View All Opened Tickets) and 'Knowledge Base' (Popular Solution, Helpdesk Statistic, Pivot Table). The main content area shows the breadcrumb '>> Knowledge Based >> Popular Solution' and the title 'Popular Solution By Product Category'. Below the title, it displays 'Voice : 2 tickets', 'Data : 2 tickets', and 'Voice and Data : 2 tickets'. A search bar is present above a table with the following data:

Division	Reported Date	Problem Summary	Problem Detail	Resolution	Resolved By
Fakultas Ekonomi & Bisnis	01-Jan-1970	Kabel piktail putus		oke solved	superadmin
Fakultas Farmasi	22-Dec-2017	Jaringan lambat	Link internet di fakultas farmasi lambat	link lambat karena trafik full	superuser

Showing 1 to 2 of 2 entries

Gambar 4.23 Halaman *Popular Solution*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan uraian pada bab-bab sebelumnya dan dari hasil perancangan program yang dibuat serta pengukuran sistem, maka penulis dapat mengambil kesimpulan sebagai berikut :

1. Aplikasi yang telah dibuat telah sesuai dengan rancangan sistem helpdesk berbasis web yang bersifat user friendly dimana setiap staf UH dapat dengan mudah melaporkan aduannya.
2. Aplikasi *helpdesk* yang telah dibuat dapat mempersingkat waktu pengolahan data *helpdesk* dan membantu DTI UH dalam menghasilkan laporan gangguan dengan cepat dan akurat
3. Berdasarkan pengujian *browser*, aplikasi ini dapat running dengan baik pada *browser IE, chrome dan firefox*.

5.2 Saran

Apabila ada yang berminat untuk mengembangkan aplikasi ini, maka disarankan hal-hal sebagai berikut:

1. Aplikasi sistem *helpdesk* berbasis website dapat dikembangkan dengan membuat notifikasi sukses via email secara otomatis.
2. Aplikasi sistem *helpdesk* dibuat secara sederhana maka dari itu pengembang dapat memodifikasi tampilan yang lebih eye catching atau menarik untuk dilihat sehingga pengguna betah dalam bekerja.

DAFTAR PUSTAKA

- Al-Fatta, Hanif (2007). *Analisis dan Perancangan Sistem Informasi: Untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta : Andi Offset
- Cassandra, Cadelina. 2015. Pengembangan Model Sistem Informasi Aplikasi Helpdesk Online PT. MUstika Memadata. Jakarta Barat: Jurnal: Vol 6 No.2:161-137
- Hidayat, Heri. 2011. Cara Instan Menguasai Pemrograman Website. Jakarta Barat : Agogos Publishing (2011)
- Kadir, Abdul. 2012. Pemrograman Database MySQL. Yogyakarta: Mediakom.
- Makkasau. 1983. Metode Analisa Sistem (Bandung : Sinar Baru, 1983)
- Nugroho, Bunafit. 2007. *PHP Profesional Pengembangan Data Array dalam Aplikasi Web*. Yogyakarta
- Rosa & M. Shalahuddin (2011). Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung : Penerbit Modula
- Rico. 2016. Analisis dan Perancangan Sistem Informasi IT-Helpdesk. Jambi: Jurnal Ilmiah Media SISFO. Vol. 10, No.2:297-299.
- Ulfa, Aini. (2009). *Aplikasi Helpdesk Online Dengan Menggunakan PHP dan Ajax*. Jakarta : Universitas Gunadarma
- <https://bayudwiarta.wordpress.com/2014/10/16/fungsional-dari-notepad/> diakses pada tanggal 18 februari 2017, pukul 21.00
- <http://westborneoblogger.blogspot.co.id/2015/08/pengertian-dan-jenis-jenis-tipe-data.html> di akses pada tanggal 13 november 2017, pukul 20.00

L

A

M

P

I

R

A

N

LAMPIRAN

Lampiran 1 Skrip Halaman Login

```

<head>
<title>Helpdesk Direktorat Teknologi Informasi</title>
<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">
<link rel="icon" href="images/favicon.ico" type="image/x-icon">
<link rel="stylesheet" type="text/css" href="css/style.css" />
</head>
<body onload="document.login.username.focus();">
<br/><br/>
<h1 class="loginform">Login Helpdesk<p>Direktorat Teknologi
Informasi</p></h1>
<div class="loginform" align="center">
<table class="loginform">
<tr><td>

</td>
<td>

echo '<p class="errmsg">' . implode('<p><p class="errmsg">', $errors)
.</p>';
}
?>
<div class="footer">
</div>
</body>
</html>

```

Lampiran 2 Script Halaman Menu Utama

```

<html>
<head>
<title>Helpdesk System</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="shortcut icon" href="images/favicon.png" type="image/x-
icon">
<link rel="icon" href="images/favicon.png" type="image/x-icon">
</head>

```

```

<frameset name="mainfs" rows="85,*" frameborder="0" border="0"
framespacing="0">
    <frame src="header.php" name="topFrame" scrolling="NO"
noresize />
    <frameset name="contentfs" cols="200,*" frameborder="0"
border="0" framespacing="0">
        <frame src="navigator.php" name="leftFrame"
scrolling="NO" noresize />
        <frame src="content.php" name="contentFrame" />
    </frameset>
</frameset><noframes></noframes>
<body>
</body>
</html>

```

Lampiran 3 Script Form Ticket

```

<body>
    <div class="breadcrumb"> >> Home >> New Ticket</div>
    <hr/>
    <form name="ticketform" method="post" action=""
onsubmit="return cekData();">
    <fieldset style="display: inline-block;">
    <legend> New Ticket </legend>
    <div class="breadcrumb">*) Field Required</div>
    <table class="formtable">
        <tr>
            <td width="120"> Division* </td><td> : </td>
            <td> <select name="idcustomer" id="idcustomer">
                <option></option>
            <?php
                $customers = $customers->get_customers();
                foreach ($customers as $customerval)
                {
                    echo '<option value=' . $
            <td> Division Problem </td><td> : </td>
            <td><label id="customerproduct"></label>
            </td>
        </tr>
    </table><br/>
    <table class="formtable">

```

```

        <tr>
            <td>Ticket No:</td><td> : </td>
            <td>New Ticket Number
            <input type="hidden" size='20' name='ticketnumber'
value="">
            </td>
        </tr>
        <tr>
            <td> Urgency (SLA)*</td><td> : </td>
            <td><select name="sla">
                <?php
                    $sla = $slas->get_sla();
                    echo '<option
        <tr>
            <td> Problem Detail* </td><td> : </td>
            <td> <textarea name="problemdetail" rows="3"
cols="38"></textarea> </td>
            </tr>
                foreach ($Asignees as $user)
                {
                    echo '<option value=' . $user['id'] . '>'
. $user['fullname'] . '</option>';
                }
            ?>
            </select> </td>
        </tr>
        <tr>
            <td> Telephone </td><td> : </td>
            <td> <input type='text' size='20' name='telp'
maxlength="20"> </td>
            </tr>
        <tr>
            <td> Email </td><td> : </td>
            <td> <input type='email' size='50' name='email'
maxlength="50"> </td>
            </tr>
    </form>

```



```

        <?php
        if(empty($errors) === false){
            echo ' <p class=errmsg>' . implode('</p><p
class=errmsg>', $errors) . '</p>';
        }
        ?>
</body>
</html>

```

Lampiran 4 *Script Change Password*

```

<body>
    <div class="breadcrumb"> >> Home >> Change Password</div>
    <hr/>
    <form name="userform" class="form" method="post" action=""
onsubmit="return cekData();">
    <fieldset>
    <legend> Change Password </legend>
    <label>Old Password :</label> <input type='password' size='30'
name='oldpassword'> <br/>
    <label>New Password :</label> <input type='password' size='30'
name
        <input type="reset" name="reset" value="Reset">
    </fieldset>
    </form>
    <br/>
    <?php
    if(empty($errors) === false){
        echo ' <p class=errmsg>' . implode('</p><p
class=errmsg>', $errors) . '</p>';
    }
    ?>
</body>
</html>

```

Lampiran 5 *Script Halaman My Request*

```

<html>
<head>
    <title>My Tickets</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
    <style type="text/css">

```

```

        body{ background-
image:url('images/corner.jpg');background-repeat:no-repeat;background-
attachment:fixed;}
        .breadcrumb{ font-size:12px;color:#0000A0;font-family:
Arial, Helvetica, sans-serif;}
,
        "bPaginate": false,
        "jQueryUI": true
    });
})
</script>

<th>Ticket No.</th>
        <th>Urgency</th>
<th>Division</th>
<th>Reported Date</th>
        <th>Reported By</th>
<th>Problem Summary</th>
        <th>Status</th>
        <th>Assignee</th>
</tr>
</thead>
<tbody>
    <?php
        foreach ($tickets as $ticket) {
            $sla = $slas->sla_data($ticket['sla']);
            $customer = $customers-
>customer_data($ticket['idcustomer']);
            $user = $users->userdata($ticket['assignee']);
            echo
                <tr><td><a
href=ticketedituser.php?id='.$ticket['id'].
>'.$ticket['ticketnumber'].'</a></td>'.
                <td>'.$sla['namasla'].'</td>'.
                <td>'.$customer['namacustomer'].'</td>'.
                <td>'.date('d-M-
Y',$ticket['reporteddate']).</td>'.
                <td>'.$ticket['reportedby'].'</td>'.
                <td>'.$ticket['problemsummary'].'</td>'.
                <td>'.$ticket['ticketstatus'].'</td>'.
                <td>'.$user['fullname'].'</td></tr>;

```



```

        <th>Status</th>
        $user = $users->userdata($ticket['assignee']);
        echo '<tr><td><a
            <td>'. $ticket['reportedby']. '</td>'.
            <td>'. $ticket['problemsummary']. '</td>'.
            <td>'. $ticket['ticketstatus']. '</td>'.
            <td>'. $user['fullname']. '</td></tr>';
        }
    ?>
</tbody>
</table>
<p>&nbsp;</p>
</body>
</html>

```

Lampiran 7 *Script Halaman My Resolution*

```

<html>
<head>
    <title>My Tickets</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
    <style type="text/css">
        body{ background-
image:url('images/corner.jpg');background-repeat:no-repeat;background-
attachment:fixed;}
    charset="utf-8">
    $(document).ready(function(){
        $('#datatables').dataTable({
            "sScrollY": "400px",
            "sScrollX": "100%",
            "bScrollCollapse": true,
            "bPaginate": false,
            "bJQueryUI": true
        });
    })
</script>

</head>
<body>
    <div class="breadcrumb"> >> Home >> My Tickets >> My
Resolution</div>

```

```

    <hr/>
    <h1>List of tickets that resolved by me</h1>
    <p>Number of tickets: <strong><?php echo $tickets_count;
?></strong> </p>
    <table id="datatables" class="display">
    <thead>
    <tr>
    <th>Ticket No.</th>
        <th>Urgency</th>
    <th>Division</th>
    <th>Reported Date</th>
        <th>Reported By</th>
    <th>Problem Summary</th>
        <th>Status</th>
        <th>Assignee</th>
    </tr>
    </thead>
    <tbody>
}
        ?>
    </tbody>
    </table>
    <p>&nbsp;</p>
</body>
</html>

```

Lampiran 8 *Script Halaman View All Open Ticket*

```

html>
<head>
    <title>Update Ticket</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
    <style type="text/css">
        body{ background-
image:url('images/corner.jpg');background-repeat:no-repeat;background-
attachment:fixed;font-family: Arial, Helvetica, sans-serif;}
        .breadcrumb{ font-size:12px;color:#0000A0;}
        .formtable {text-align:left; font-size:12px;color:#000000;
background-color:#f0f0f0;padding:10px;width:600px; }
        .errmsg { font-size:10pt;color:#ff0000;text-align:left;}

```

```

</style>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript">
function cekData()
{
    if (ticketform.resolution.value == "")
    {
        alert("Resolution must be filled!");
        ticketform.resolution.focus();
        return false;
    }
    if (ticketform.ticketstatus.value == "Closed")

        echo $customer['namacustomer'];
    ?> </td>
</tr>
<tr>
    <td>Complain Type</td><td> : </td>
    <td> <?php echo $customer['customerproduct']; ?>
</td>
</tr>
<tr>
    <td> Warranty Period</td><td> : </td>
    <td>
    <?php
        $project=$projects-
>get_project_customer($customer['idcustomer']);
        echo $project['warrantyperiod'].' Year';
    ?> </td>
</tr>
<tr>
    <td> Contract Period</td><td> : </td>
    <td>
    <?php
        echo $project['contractperiod'].' Month';
    ?>
    </td>
</tr>
</table>
<br/>
<table class="formtable">
    <tr>

```

```

</td><td> : </td>
        <td> <?php echo $ticket['problemsummary']; ?>
        <input type="hidden" name="problemsummary"
value="<?php echo $ticket['problemsummary']; ?>"> </td>
    </tr>
    <tr valign="top">
        <td> Problem Detail </td><td> : </td>
        <td> <?php echo nl2br($ticket['problemdetail']); ?>
    </td>
    </tr>
    <tr>
        <td> Telephone </td><td> : </td>
        <td> <?php echo $ticket['telp']; ?>
        <input type='hidden' name='telp' value='<?php echo
$ticket['telp']; ?>'> </td>
    </tr>
    <tr>
        <td> Email </td><td> : </td>
        <td> <?php echo $ticket['email']; ?>
        <input type='hidden' name='email' value='<?php
echo $ticket['email']; ?>'> </td>
    </tr>
</table>
<br/>
<table class="formtable">
    <tr>
        <td width="120"> Assign to </td><td> : </td>
        <td> <?php $userassignee = $users-
>userdata($ticket['assignee']); echo $userassignee['fullname']; ?>
        <input type='hidden' name='idassignee'
value='<?php echo $ticket['assignee']; ?>'> </td>
    </tr>
    <tr>
        <td> Status* </td><td> : </td>
        <td> <input type="hidden" name="oldticketstatus"
value="<?php echo $ticket['ticketstatus']; ?>">
        <select name="ticketstatus">
            <?php

```

```

                                echo ' <option value=' .
$ticket['ticketstatus']. ' selected="selected">'. $ticket['ticketstatus'] .
'</option>';
                                ?>
                                <option value="Assigned"> Assigned
</option>
                                <option value="Resolved"> Resolved
</option>
                                <option value="Pending"> Pending
</option>
                                <option value="Closed"> Closed </option>
                                </select>
                                </td>
                                </tr>
                                <tr valign="top">
                                <td> Resolution* </td><td> : </td>
                                <td> <textarea name="resolution" rows="3"
cols="38"><?php echo $ticket['resolution']; ?></textarea> </td>
                                </tr>
                                <tr>

```

Lampiran 9 Script Halaman Admin

```

<html>
<head>
    <title>Helpdesk System</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
    <link rel="stylesheet type="text/css" href="css/style.css">
</head>
<body bgcolor="#003566">
<div id="leftmenu">
<div id="headleftmenu">Data Master</div>
<ul>
    <li><a href="userlist.php" target="contentFrame">User
List</a></li>
    <li><a href="customerlist.php" target="contentFrame">Division
List</a></li>
</ul>
</div>

```



```

<div id="leftmenu">
<div id="headleftmenu">Ticket Admin</div>
<ul>
  <li><a href="ticketlist.php" target="contentFrame">List all
Tickets</a></li>
  <li><a href="slalist.php" target="contentFrame">SLA
Setting</a></li>
  <li><a href="hdnews.php" target="contentFrame">Helpdesk
News</a></li>
</ul>
</div>
<div id="leftmenu">
</div>
</body>
</html>

```

Lampiran 10 *Script Halaman Add User List*

```

<html>
<head>
  <title>Add New User</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
  <link href="css/styleform.css" rel="stylesheet" type="text/css" />
  <style type="text/css">
    body{ background-
image:url('images/corner.jpg');background-repeat:no-repeat;background-
attachment:fixed;font-family: Arial, Helvetica, sans-serif;}
  </style>
  <script type="text/javascript">
function cekData()
{
  if (userform.fullname.value == "")
  {
    alert("User full name must be filled!");
    userform.fullname.focus();
    return false;
  }
  if (userform.email.value == "")
  {
    alert("User email address must be filled!");
    userform.email.focus();
    return false;
  }
  var filter = new RegExp(

```

```

    "^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@" +
    "[A-Za-z0-9-]+(\\.[A-Za-z0-9-]+)*(\\.[A-Za-
z]{2,})$");
    if (!filter.test(userform.email.value))
    {
        alert("Please enter a valid email address!");
        userform.email.focus();
        return false;
    }
    if (userform.telp.value == "")
    {
        alert("User telephone must be filled!");
        userform.telp.focus();
        return false;
    }
    if (/^D/.test(userform.telp.value))
    {
        alert("User telephone must be filled with numbers only!");
        userform.telp.focus();
        return false;
    }
    <label>Telp :</label> <input type='text' size='30' name='telp'>
<br/>
    <label>User Name :</label> <input type='text' size='30'
name='username'> <br/>
    <label>Password :</label> <input type='text' size='30'
name='password'> <br/>
    <label>User Level :</label>
    <select name="level">
        <option value="User" selected="selected"> User
</option>
        <option value="Manager"> Manager </option>
        <option value="Admin"> Admin </option>
    </select><br/>
    <label>Lock User :</label>
    <input type="radio" name="locked" value="0" /> <font
color="#ffffff"> Yes </font>
    <input type="radio" name="locked" value="1" CHECKED
/> <font color="#ffffff"> No </font>
    <label>&nbsp;</label><br/>
    <input type="submit" name="submit" value="Save">
    <input type="reset" name="reset" value="Reset">
</fieldset>

```

```

        </form>
        <?php
            if(empty($errors) === false){
                echo ' <p class=errmsg>' . implode('</p><p
class=errmsg>', $errors) . '</p>';
            }
        ?>
    </body>
</html>

```

Lampiran 11 *Script Halaman User List*

```

<html>
<head>
    <title>User List</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
    <style type="text/css">
        body{ background-
image:url('images/corner.jpg');background-repeat:no-repeat;background-
attachment:fixed;}
        .breadcrumb{ font-size:12px;color:#0000A0;font-family:
Arial, Helvetica, sans-serif;}
    <tr>
        <th>User Name</th>
            <th>Level</th>
        <th>Full Name</th>
        <th>Email</th>
            <th>Telepon</th>
        <th>Last Update</th>
            <th>Locked</th>
            <th></th>
    </tr>
</thead>
<tbody>
    <?php
        foreach ($members as $member) {
            if ($member['confirmed']==1)
                {$locked='No';}
            else

```

```

        {$locked='Yes';}
        echo                                '<tr><td><a
href=useredit.php?id='.$member['id'].
'>'.$member['username'].'</a></td>'.
        '<td>'.$member['level'].'</td>'.
        '<td>'.$member['fullname'].'</td>'.
        '<td>'.$member['email'].'</td>'.
        '<td>'.$member['Telp'].'</td>'.
        '<td>'.date('d-M-Y                H:i',
$member['time']).</td>'.
        '<td>'.$locked.</td>'.
        '<td><a
href=userdel.php?id='.$member['id'].          '      onclick="return
delete_confirm();">del</a></td></tr>';
        }
        ?>
</tbody>
</table>
<p>&nbsp;</p>
</body>
</html>

```

Lampiran 12 *Script Halaman Edit User List*

```

<html>
<head>
    <title>Update User</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">

    if (userform.telp.value == "")
    {
        alert("User telephone must be filled!");
        userform.telp.focus();
        return false;
    }
    if(/\D/.test(userform.telp.value))
    { alert("User telephone must be filled with numbers only!");
      userform.telp.focus();
      return false;
    }

    else
        return true;

```

```

    }
  </script>
</head>
<body>
  <div class="breadcrumb"> >> Admin >> User List >> Edit
  User</div>
  <hr/>
  <div id="notes"> *) All fields are required, except password <br/>
  if password is blank then password will not change
  </div><br/>
  <form name="userform" class="form" method="post" action=""
  onsubmit="return cekData();">
    <fieldset style="display: inline-block;">
      <legend> Edit User </legend>
      <label>Full Name :</label> <input type='text' size='30'
  name='fullname' value='<?php echo $member['fullname']; ?>'>
      <label>Email :</label> <input type='text' size='30' name='email'
  value='<?php echo $member['email']; ?>'> </td>
      <label>Telp :</label> <input type='text' size='30' name='telp'
  value='<?php echo $member['Telp']; ?>'> </td>
      <label>User Name :</label> <input type='text' size='30'
  name='username' value='<?php echo $member['username']; ?>' disabled>
      </td>

      <input type='submit' name='submit' value=' Update '>
      <input type='reset' name='reset' value=' Reset '>
    </fieldset>
  </form>
</body>
</html>

```

Lampiran 13 Script Division List

```

<html>
<head>
  <title>Division List</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-
  8859-1">
  <style type="text/css">
    body{ background-
  image:url('images/corner.jpg');background-repeat:no-repeat;background-
  attachment:fixed;}

```

```

        .breadcrumb{ font-size:12px;color:#0000A0;font-family:
        Arial, Helvetica, sans-serif;}
    </style>
    <link rel="stylesheet" type="text/css" href="css/datatable.css" />
    <link rel="stylesheet" type="text/css" href="css/jquery-ui.css" />
    <script src="js/jquery.js" type="text/javascript"></script>
    <script
        src="js/jquery.dataTables.js"
    type="text/javascript"></script>
    <script type="text/javascript" charset="utf-8">
    $(document).ready(function(){
        $('#datatables').dataTable({
            "sScrollY": "300px",
            "sScrollX": "100%",
            "bScrollCollapse": true,
            "bPaginate": false,
            "bJQueryUI": true
        });
    })

```

```

<p>Number of registered division: <strong><?php echo $member_count;
?></strong> </p>

```

```

    <p><button onclick="window.location='customeradd.php';">Add
New Division</button></p>

```

```

    <table id="datatables" class="display">
    <thead>
    <tr>
    <th>ID Division</th>
        <th>Name Division</th>
    <th>Location</th>
    <th>Telepon</th>
        <th>PIC</th>
    <th>Leader</th>
        <th>Complain Type</th>
    <th></th>
    </tr>
    </thead>
    <tbody>
        <?php
        foreach ($members as $member)

```

```

        {      if (@$member['confirmed']==='1')
                {$locked='No';}
            else
                {$locked='Yes';}
                $disp_id_customer=sprintf("%04s",
$member['idcustomer']);
                echo                                '<tr><td><a
href=customeredit.php?id='.$member['idcustomer'].
'>'.$disp_id_customer.'</a></td>'.
                '<td>'.$member['namacustomer'].'</td>'.

</table>
        <p>&nbsp;</p>
</body>
</html>

```

Lampiran 14 Script Form Add New Division

```

<html>
<head>
    <title>Add New Divisi</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
    <style type="text/css">
        body{background-
image:url('images/corner.jpg');background-repeat:no-repeat;background-
attachment:fixed;font-family: Arial, Helvetica, sans-serif;}
        .breadcrumb{font-size:12px;color:#0000A0;}
        .formtable {font-size:12px;color:#000000; background-
color:#f0f0f0;padding:10px;width:500px;}
        .errmsg {font-size:10pt;color:#ff0000;text-align:left;}
    </style>

    alert("Division telephone must be filled with numbers only!");
        customerform.telp.focus();
        return false;
    }
    if (customerform.email.value == "")
        {      alert("Division email must be filled!");
            customerform.email.focus();
            return false;

```

```

</script>
</head>
<body>
    <div class="breadcrumb"> >> Admin >> Division List >> Add
New Divisi</div>
    <hr/>
    <form name="customerform" method="post" action=""
onsubmit="return cekData();">
    <fieldset style="display: inline-block;">
    <legend> Add New Divisi </legend>
    <table class="formtable">
        <tr align="left">
            <td> Divisi Name </td><td> : </td>
            <td> <input type='text' size='50'
name='namacustomer' maxlength="50"> </td>
        </tr>
        <tr align="left">
            <td> Address </td><td> : </td>
            <td> <textarea name="alamat" rows="3"
cols="38"></textarea> </td>
        </tr>
        <tr align="left">
            <td> Telp </td><td> : </td>
            <td> <input type='text' size='20' name='telp'
maxlength="20"> </td>
        </tr>
    </table>
    </body>
</html>

```

Lampiran 15 Script Halaman Ticket Admin

```

<html>
<head>
    <title>List of Tickets</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
    <style type="text/css">
        body{ background-
image:url('images/corner.jpg');background-repeat:no-repeat;background-
attachment:fixed;}

```



```

        .breadcrumb{ font-size:12px;color:#0000A0;font-family:
Arial, Helvetica, sans-serif;}
    </style>
    <link rel="stylesheet" type="text/css" href="css/datatable.css" />
    <link rel="stylesheet" type="text/css" href="css/jquery-ui.css" />
    <script src="js/jquery.js" type="text/javascript"></script>
    <script
        src="js/jquery.dataTables.js"
type="text/javascript"></script>
    <script type="text/javascript" charset="utf-8">
    $(document).ready(function(){
        $('#datatables').dataTable({
            "s
    <th>Ticket No.</th>
        <th>Urgency</th>
        <th>SLA Goal Time</th>
    <th>Division</th>
    <th>Reported Date</th>
        <th>Documented Date</th>
    <th>Problem Summary</th>
        <th>Status</th>
        <th>Assignee</th>
        <th></th>
    </tr>
    </thead>
    <tbody>
        <?php
        foreach ($ticket_list as $ticket) {
            $sla = $slas->sla_data($ticket['sla']);
            $documenteddate=$ticket['documenteddate'];
            $resolutiontime=$sla['resolutiontime'];
            $
            <td>'.$customer['namacustomer'].'</td>'.
                <td>'.date('d-M-Y
H:i:s',$ticket['reporteddate']).</td>'.
                <td>'.date('d-M-Y
H:i:s',$ticket['documenteddate']).</td>'.
                <td>'.$ticket['problemsummary'].'</td>'.
                <td>'.$ticket['ticketstatus'].'</td>'.

```