

TUGAS AKHIR
HELPDESK SYSTEM INTERNAL PADA
LPP TVRI SULUT



Oleh

Zulkifly U. Tueno
NIM : 11 024 028

Dosen Pembimbing

Sulastri Eksan, ST
NIP. 19790720 200604 2 003

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI MANADO
JURUSAN TEKNIK ELEKTRO
PROGRAM STUDI INFORMATIKA
2015

LEMBAR PENGESAHAN

HELPDESK SYSTEM INTERNAL PADA
LPP TVRI SULUT

Oleh
Zulkifly U. Tueno
NIM : 11 024 028

*Tugas Akhir ini telah diterima dan disahkan sebagai persyaratan untuk
menyelesaikan Pendidikan Diploma IV Teknik Elektro
Bidang Keahlian Teknik Informatika
Politeknik Negeri Manado*

Manado, 2015

Ketua Panitia Tugas Akhir,

Dosen Pembimbing,

Fanny Jouke Doringin, ST, MT
NIP. 19670430 199203 1 003

Sulastri Eksan, ST
NIP. 19790720 200604 2 003

Mengetahui
Ketua Jurusan Teknik Elektro,

Ir Jusuf Luther Mappadang, MT
NIP.196106011990031002

PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini:

Nama : Zulkifly U. Tueno

Nim : 11 024 028

Program Studi : Teknik Informatika

Menyatakan dengan sebenarnya bahwa Tugas Akhir yang saya tulis ini benar-benar merupakan hasil karya pengembangan saya sendiri, bukan merupakan pengambilalihan tulisan atau pemikiran orang lain. Apabila kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan Tugas Akhir ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Manado, 27 Agustus 2015

Yang menyatakan

Zulkifly U. Tueno

ABSTRAK

Helpdesk system sekarang telah banyak digunakan oleh perusahaan – perusahaan besar untuk menunjang pekerjaan selama berada di kantor. Pada umumnya di setiap perusahaan masih mengandalkan cara manual dalam berkomunikasi antara tiap – tiap pegawai ataupun divisi, dengan mendatangi divisi atau pegawai lain, hal ini tentunya sangat menguras waktu dan tenaga apalagi jika perusahaan tersebut sudah bergerak dalam skala yang besar yang memiliki kantor dengan lantai berpuluh – puluh tingkat. Dengan menggunakan helpdesk semua hal itu bisa tertutupi dengan hanya mengakses helpdesk lewat PC yang telah di sediakan untuk tiap – tiap pegawai. Helpdesk hadir dengan bermacam nama atau sebutan, tapi satu yang pasti ketika sudah berhubungan dengan pelayanan konsumen, secara online, itulah yang dimaksud dengan helpdesk. Munculnya helpdesk dengan berbagai nama dan sebutan tersebut di karenakan adalah perbedaan tiap – tiap instansi juga kebutuhannya, maka dari itu, instansi yang berbeda, pasti memiliki bidang kerja yang berbeda, system yang berbeda pula, serta aplikasi pendukung yang berbeda.

Penelitian ini bertujuan membuat helpdesk system yang notabennya berfungsi sebagai aplikasi external perusahaan menjadi aplikasi yang berbasis internal perusahaan agar dapat digunakan oleh pegawai dalam perusahaan untuk berkomunikasi baik sesama pegawai ataupun antar divisi yang berbeda dalam hal menyampaikan keluhan atau pertanyaan serta mendokumentasikan perihal masalah yang di hadapi dalam menjalani pekerjaan selama di perusahaan.

Pada penelitian kali ini penulis menggunakan beberapa metode penelitian untuk membantu penulis menyelesaikan aplikasi ini, mulai dari pengumpulan data dan informasi serta perancangan system aplikasi yang dibutuhkan untuk membantu penulis dalam mendalami keadaan real yang terjadi di lapangan agar aplikasi yang di buat ini nantinya benar – benar bisa digunakan dan membantu pegawai perusahaan yang di jadikan tempat penelitian. Selanjutnya melakukan perancangan interface yang dibuat sederhana namun bisa berjalan dengan baik dan mudah digunakan. Dan terakhir melakukan implementasi dan testing sesuai dengan masalah real yang terjadi di lapangan, dan kemudian di terapkan dan di implementasikan kembali kepada helpdesk ini.

Hasil penelitian menunjukkan bahwa aplikasi helpdesk system internal pada LPP TVRI SULUT ini dapat membantu memfasilitasi pegawai dalam perusahaan dalam menyampaikan berbagai keluhan dan pertanyaan mereka dan juga dapat menutupi masalah jarak dan waktu dimana pegawai harus mendatangi tempat divisi lain untuk menyampaikan keluhan mereka, dan kini itu bisa dilakukan hanya dengan duduk di depan PC dan mengakses aplikasi helpdesk system ini yang telah di hosting ke server internalnya perusahaan. Dengan menggunakan aplikasi helpdesk system ini telah mempermudah pegawai dan mengatasi masalah jarak dalam hal menyampaikan pertanyaan dan keluhan mereka, dan juga terdokumentasi serta tepat sasaran tentunya kepada siapa masalah dan pertanyaan ini akan di arahkan.

Kata Kunci : Helpdesk, System, Tiket

KATA PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT atas berkat, rahmar, taufik dan hidayah-Nya sehingga penyusunan Tugas Akhir yang berjudul Helpdesk System Internal Pada LPP TVRI SULUT ini dapat diselesaikan dengan baik.

Dalam penulisan Tugas Akhir ini penulis menyadari banyak sekali kendala – kendala yang dihadapi. Dalam kesempatan ini penulis menyampaikan terima kasih yang sebesar – besarnya kepada :

1. Allah Subhanahu Wata'ala, yang tanpa karunia dan rahmatnya penulis tidak bisa menyelesaikan penulisan Tugas Akhir ini.
2. Ir. Jemmy Rangan, ST .MT, selaku Direktur Politeknik Negeri Manado.
3. Ir. Jusuf Luther Mappadang, MT, selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Manado.
4. Ir. Nikita A. E. Sejangbati, selaku Kepala Program Studi Teknik Informatika Politeknik Negeri Manado.
5. Ibu Sulastri Eksan, ST , selaku dosen pembimbing yang telah memberikan banyak sekali arahan dan bimbingannya demi kematangan penulisan ini.
6. Semua staf dosen Politeknik Negeri Manado yang telah memberikan bekal Ilmu kepada penulis.
7. Yang tercinta Ibu dan Bapak yang telah membesarkan, membimbing, serta mendidik penulis dengan penuh kasi sayang dan penuh kesabaran serta selalu memberikan dorongan moril dan do'a yang ikhlas sehingga terselesaikan penulisan ini.
8. Terima Kasih juga kepada semua pihak yang telah turut membantu dan tidak dapat penulis sebutkan satu persatu.

Dengan sangat rendah hati penulis menyadari bahwa penulisan ini masih sangat jauh dari sempurna, oleh karena itu segala bentuk kritik dan saran yang

menuntun kepada kebaikan dan penyempurnaan penulisan ini sangat diharapkan dan diterima penulis dengan tangan terbuka.

Akhirnya penulis mengharapkan penulisan ini berguna bagi penulis khususnya dan bagi para pembaca umumnya. Semoha Allah Subhanahu Wata'ala berkenan merestui segala yang penulis cita – citakan selama ini.

Manado, 27 Agustus 2015

Penulis,

Zulkifly U. Tueno

DAFTAR ISI

JUDUL.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
ABSTRAK.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Pembatasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penulisan.....	4
1.6 Sistematika Penulisan.....	4

BAB II TINJAUAN PUSTAKA

2.1 Helpdesk.....	6
2.1.1 Contoh Helpdesk.....	8
2.1.2 Fitur Helpdesk.....	9
2.1.3 Fitur Helpdesk TVRI SULUT.....	11
2.2 Hypertext Processor (PHP).....	11
2.2.1 Sejarah Singkat PHP.....	14
2.3 MySQL.....	15
2.3.1 Konektifitas PHP dan MySQL.....	16

2.4 HTML.....	17
2.5 CSS.....	19
2.5.1 Sejarah Singkat CSS.....	20
2.6 Bootstrap twitter.....	22
2.7 Struktur Organisasi.....	23
BAB III Perancangan Helpdesk System Internal TVRI SULUT	
3.1 Waktu dan Tempat.....	25
3.2 Kerangka Konseptual.....	25
3.3 Diagram Alur (Flowchart).....	26
3.4 Perancangan.....	30
3.4.1 Perancangan Database.....	31
3.4.2 Perancangan Script.....	34
3.4.3 Perancangan Interfsece.....	37
BAB IV PEN GUJIAN DAN ANALISA	
4.1 Tujuan Pengujian.....	41
4.2 Pengujian Log In User.....	41
4.3 Pengujian Halaman Home.....	43
4.4 Pengujian Contoh Kasus.....	49
4.1.1 Perbaikan PC.....	49
4.1.2 Pengadaan Inventaris.....	57
BAB V PENUTUP	
5.1 Kesimpulan.....	62
5.2 Saran.....	62
DAFTAR PUSTAKA	63
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Cara Kerja Helpdesk.....	7
Gambar 2.2 Contoh Skrip HTML.....	13
Gambar 2.3 Contoh Skrip PHP.....	13
Gambar 2.4 Hasil Skrip PHP.....	14
Gambar 2.5 Hasil Download.....	22
Gambar 2.6 Struktur Organisasi.....	23
Gambar 2.7 Diagram Alur Helpdesk.....	24
Gambar 3.1 Kerangka Konseptual.....	25
Gambar 3.2 Flowchart Tiket.....	27
Gambar 3.3 Alur Kerja Umum.....	28
Gambar 3.4 Script Customer.php.....	34
Gambar 3.5 Script General.php	35
Gambar 3.6 HDNews.php.....	35
Gambar 3.7 SLA.php.....	36
Gambar 3.8 Tiket.php.....	36
Gambar 3.9 user.php.....	37
Gambar 3.10 Halaman Log In.....	38
Gambar 3.11 Halaman Home.....	39

Gambar 4.1 Halaman Log In.....	41
Gambar 4.2 Invalid Username/Password.....	42
Gambar 4.3 Halaman Home.....	44
Gambar 4.4 Pembuatan Tiket Home.....	44
Gambar 4.5 Tampilan Tiek.....	45
Gambar 4.6 Tampilan Request.....	46
Gambar 4.7 Status Tiket.....	46
Gambar 4.8 Frame Navigasi.....	47
Gambar 4.9 Report Grafik.....	48
Gambar 4.10 Navigasi Admin.....	48
Gambar 4.11 Log Out.....	49
Gambar 4.12 Log In Yuiana.....	50
Gambar 4.13 Klik Buat Tiket.....	51
Gambar 4.14 Formulir Tiket Yuliana.....	52
Gambar 4.15 Report Tiket.....	53
Gambar 4.16 Beranda Zulkifly.....	53
Gambar 4.17 Pemberitahuan Tike.....	54
Gambar 4.18 Link Tiket.....	55
Gambar 4.19 Daftar Permintaan.....	55
Gambar 4.20 Form Tanggapan.....	56
Gambar 4.21 Assign to.....	56
Gambar 4.22 Status.....	57
Gambar 4.23 Follow Up Tiket.....	57

Gambar 4.24 Log In Jane.....	59
Gambar 4.25 Beranda Jane.....	59
Gambar 4.26 Form Tiket Jane.....	60
Gambar 4.27 Report Tiket Janr.....	60
Gambar 4.28 Log In Sofyan.....	61

DAFTAR TABEL

Tabel 3.1 Tabel Divisi.....	31
Tabel 3.2Tabel SLA.....	32
Tabel 3.3 Tabel Tiket.....	32
Tabel 3.4Tabel News.....	33
Tabel 3.5 Tabel Tiket.....	33
Tabel 3.6 Tabel User.....	36
Tabel 4.1 Tabel Pengujian Log In.....	43

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di bidang aplikasi berbasis web sudah menjadikan kualitas kehidupan manusia semakin tinggi. Aplikasi berbasis web dikembangkan untuk mempermudah pekerjaan manusia. Banyak aplikasi berbasis web yang dirancang dikarenakan aplikasi berbasis web banyak digunakan pada berbagai instansi perkantoran atau berbagai macam organisasi lainnya. Instansi yang berbeda, pasti mempunyai fungsi yang berbeda, struktur yang berbeda pula, dan otomatis mempunyai aplikasi pembantu yang berbeda, tergantung kebutuhan tiap – tiap instansi atau organisasi tersebut. Apalagi dengan instansi yang menggunakan aplikasi helpdesk system yang bergerak di bidang pelayanan konsumen. Untuk memudahkan perusahaan dalam menangani keluhan maupun pertanyaan dari pelanggan, biasanya perusahaan menyediakan helpdesk atau servis desk atau contact center.

Helpdesk sebagai Single Point Of Contact (SPOC) menjadi fasilitas komunikasi antara pelanggan atau pengguna dengan tim pendukung di perusahaan penyedia produk atau jasa. Pelanggan dan pengguna biasanya menyampaikan keluhan atau pertanyaan melalui telepon atau email, ada juga yang melalui website jika disediakan. Keluhan maupun pertanyaan itu akan pertama kali ditanggapi oleh tim helpdesk. Dengan bantuan aplikasi helpdesk system, tim helpdesk dapat memasukan masalahnya dan mencari solusinya. Jika tim helpdesk dapat menyelesaikannya, maka ticket helpdesk dapat langsung di close, jika tidak maka ticket di eskalasi (assigned) ke teknisi atau tim support.

Selanjutnya, tim support akan melakukan analisa dan mencari solusinya. Jika solusi sudah didapat, maka tim support akan memasukkan solusi tersebut ke dalam aplikasi helpdesk system, sehingga tim helpdesk dapat menindak lanjutinya kepada pelanggan. Jika teknisi atau tim support tidak dapat menyelesaikan

masalahnya, maka status ticket helpdesk akan dibuat menjadi pending. Artinya, masalah ini perlu di eskalasi ke vendor penyedia produk.

Helpdesk sudah berkembang sejak lama, ada beberapa contoh perusahaan atau instansi yang menggunakan aplikasi ini untuk berkomunikasi dan melayani para konsumennya, Biasanya helpdesk system digunakan sebagai alat komunikasi antara perusahaan dan pelanggan, namun pada tugas akhir ini aplikasi helpdesk system berbasis web akan digunakan sebagai alat komunikasi internal perusahaan, dimana tiap – tiap karyawan dapat berkomunikasi atau menyampaikan keluhan serta pertanyaan mereka kepada karyawan lain sesama divisi ataupun berbeda divisi ataupun bisa langsung dari atasan kepada bawahannya, layaknya media social internal perusahaan. Dan kali ini penyusun mengambil lokasi penelitian di Lembaga Penyiaran Publik Televisi Republik Indonesia Stasiun Sulut (LPP TVRI SULUT). Jika pada umumnya aplikasi helpdesk system membutuhkan orang atau tim untuk menggerakkan aplikasi ini agar keluhan konsumen tersampaikan kepada bagian atau divisi pada terdapat pada perusahaan tersebut, maka pada kali ini penulis berusaha membuat agar aplikasi ini digerakkan oleh penggunanya sendiri, dalam hal ini antar sesama pegawai dalam perusahaan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas tujuan penulisan tugas akhir yang hendak dicapai yaitu untuk mempermudah dan memfasilitasi karyawan LPP TVRI SULUT dalam menampung, mendokumentasikan, dan menanggapi berbagai keluhan serta pertanyaan karyawan yang muncul dalam melaksanakan tugas keseharian di kantor.

1.3 Batasan Masalah

Untuk lebih fokusnya penelitian ini maka permasalahan yang hendak diteliti hanya di batasi pada :

1. Merancang alur kerja helpdesk system internal LPP TVRI SULUT
2. Program yang dibuat berbasis web dengan menggunakan perintah – perintah yang di pemograman PHP dan HTML.
3. Melakukan pengkodean atau coding script PHP dan HTML yang utamanya tentang hak akses, grafik, dan skala prioritas.

1.4 Tujuan Penelitian

Dalam bentuk mengoptimalisasikan penggunaan teknologi yang umum digunakan saat ini penulis berinisiatif membuat helpdesk system internal pada LPP TVRI SULUT yang memungkinkan pengguna untuk dapat menyampaikan keluhan, memberikan solusi untuk keluhan pengguna lain, dan sekaligus mendokumentasikan keluhan dan pertanyaan tersebut untuk menjadi laporan atau report bagi atasan dalam mengevaluasi bergai kendala dalam pekerjaan di kantor sehari – hari dan menjadi catatan apabila keluhan atau pertanyaan yang sama muncul kembali.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah sendiri intinya adalah memudahkan para pegawai dalam berkomunikasi khususnya menyampaikan keluhan, pendapat, dan saran atas permasalahan di kantor sehari – harinya.

1.6 Metodologi Penulisan

Dalam tugas akhir ini penulis menggunakan beberapa metode penelitian, diantaranya sebagai berikut :

1. Studi Lapangan

Metode ini dilakukan penulis untuk mendesain, merancang, menguji, dan mengimplementasikan aplikasi yang penulis buat dengan melakukan observasi kepada perusahaan atau instansi terkait.

2. Studi Pustaka

Studi pustaka dilakukan untuk mencari referensi yang dapat dijadikan dasar kajian dan landasan teori yang mendukung data – data informasi sebagai acuan dalam melakukan perencanaan, pembuatan, dan percobaan pada aplikasi maupun laporan Tugas Akhir ini.

1.7 Sistematika Penulisan

Sistematika Penulisan terdiri dari 5 bab, yang terdiri dari :

Sistematika penulisan terdiri dari 5 bab yang terdiri dari :

1. BAB 1 : bagian pertama adalah pendahuluan, pada bagian ini diuraikan tentang latar belakang penelitian, permasalahan – permasalahan yang ada pada penelitian, tujuan penelitian, serta batasan dan ruang lingkup dari penelitian. Dan dibagian akhir bab ini diuraikan sistematika penyajian laporan penelitian.
2. BAB 2 : bagian kedua mengenai tentang landasan teori dalam pembuatan tugas akhir ini yang membahas tentang helpesk system, dasar pemogramannya, situasi lokasi yang dijadikan tempat penelitian, serta pengalihan helpdesk system yang umumnya bersifat external antara perusahaan dengan konsumen menjadi bersifat internal yaitu antara pegawai dengan pegawai lainnya maupun antar divisi dengan divisi lainnya.
3. BAB 3 : bagian ini memaparkan tentang perancangan system.

4. BAB 4: bagian ini berisi mengenai informasi dari hasil uji coba dan implementasi terhadap aplikasi yang dibuat.
5. BAB 5: menyajikan kesimpulan serta saran yang bisa digunakan sebagai dasar dalam pengembangan alat ini untuk kedepannya.

BAB II

TINJAUAN PUSTAKA

2.1 Helpdesk

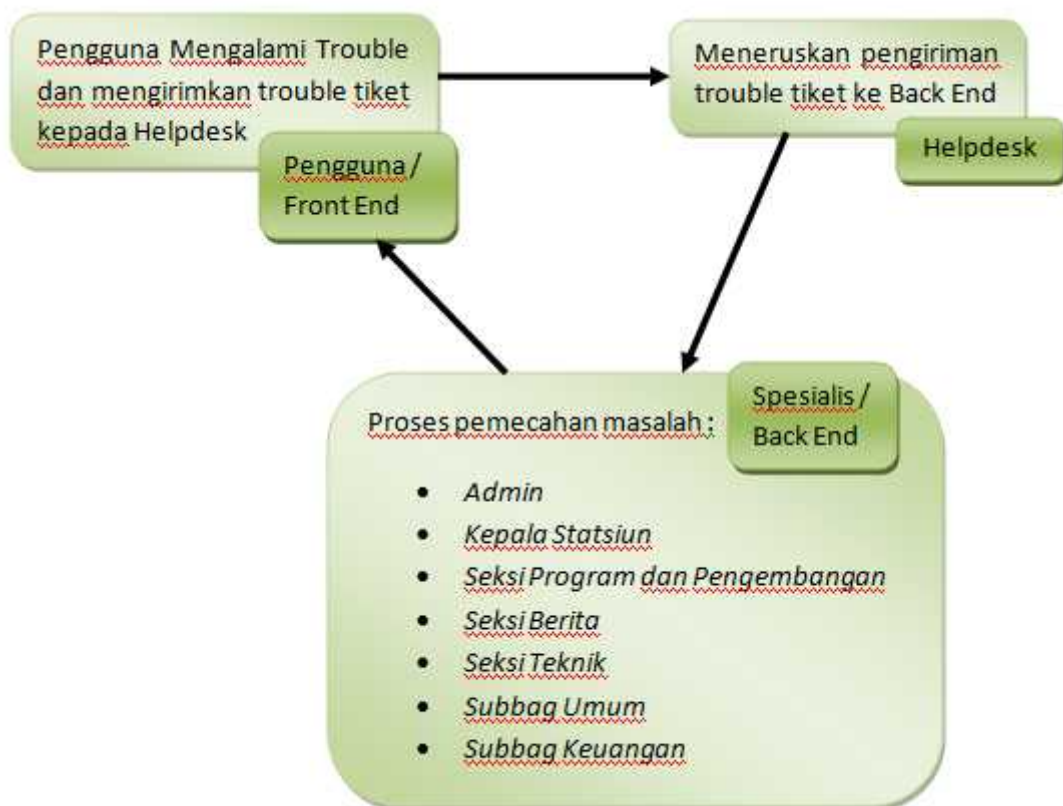
Secara definisi Helpdesk adalah salah satu media untuk membantu menyelesaikan masalah yang biasanya berhubungan dengan komputer. Helpdesk pada dasarnya adalah sebuah center point dimana masalah atau issue dilaporkan dan diatur secara terurut dan diorganisasikan. Dari perspektif umum, helpdesk merupakan bagian pelengkap dari sebuah fungsi pelayanan, dan bertanggung jawab sebagai sumber dari pemecahan masalah atau issue lainnya. Menurut Wooten (2001, p5), "*Helpdesk is a formal organization that provides support function to users of the companies product, services, or technology*". Jadi helpdesk merupakan bagian dari perusahaan yang menyediakan dokumen fungsi produk, servis atau teknologi dari perusahaan tersebut. Helpdesk adalah sebuah departemen atau bagian dalam perusahaan yang melayani atau menanggapi pertanyaan teknis pengguna. Helpdesk digunakan untuk menjawab pertanyaan client. Pertanyaan dan jawaban dapat disampaikan melalui telepon, email, web atau fax. Bahkan ada software helpdesk yang membuat orang lebih mudah untuk menjalankan helpdesk dengan cepat untuk menemukan jawaban yang bersifat umum. Berdasarkan definisi diatas helpdesk merupakan bagian struktur atau program yang menangani pertanyaan dan keluhan baik itu dari pihak internal atau pihak eksternal dengan menyediakan informasi atau solusi yang dibutuhkan pengguna.

Keuntungan memakai Helpdesk adalah seluruh request dan support yang telah dilakukan tersimpan dengan terstruktur di dalam database, sehingga jika terjadi masalah yang sama atau yang sering terjadi maka dapat ditemukan solusinya dengan lebih cepat. Kitapun dapat memilah masalah yang masuk, sehingga tahu siapa yang harus menangani dan memberikan skala prioritas untuk masalah yang masuk ke sistem. Helpdesk memberikan pelayanan terbaik pada

penggunanya serta dapat menurangi biaya. Hal ini dikarenakan berbagai keuntungan penggunaan helpdesk.

1. Helpdesk dapat memberikan solusi atas pertanyaan – pertanyaan dalam kurun waktu singkat.
2. Helpdesk dapat mengecek permasalahan yang ada dan mengatur pembagian staff.
3. Helpdesk dapat meningkatkan efisiensi perusahaan dalam menangani pertanyaan dan keluhan pelanggan.
4. Helpdesk dapat memberi laporan kerja perkembangan kinerja para staff dan pimpinan.
5. Helpdesk dapat menangani pertanyaan dan keluhan yang sejenis karena pertanyaan dan keluhan dicatat.

Berikut adalah gambaran cara kerja helpdesk :



Gambar 2.1 Cara Kerja Helpdesk

2.1.1 Contoh Helpdesk

Helpdesk sudah ada sekian lama dan telah banyak di gunakan oleh perusahaan – perusahaan besar untuk membantu kinerja dalam meanggapi masalah dalam perusahaan baik internal maupun external perusahaan. Dari sekian banyak helpdesk ada beberapa contoh helpdesk yang paling banyak digemari oleh penggunanya, yaitu Trelis Desk dan Zendesk.

a. Trelis Desk

Trelis Desk dari Accord5 hampir mengakomodasi semua sistem yang dibutuhkan pada aplikasi helpdesk, misalnya:

1. Kategori support berdasar department yang bisa pengguna sesuaikan,
2. Kategori user dan pemberian level akses untuk departemen tertentu
3. Multi Template dan Multi bahasa
4. Support ticket bisa disertai dengan attachment
5. Support SMTP dan banyak lagi

Aplikasi ini dibuat dengan menggunakan PHP dan sistem templatanya menggunakan Smarty, bagi yang sudah familiar dengan smarty mungkin akan mudah untuk melakukan pembuatan template.

Ada sedikit yang kurang pada aplikasi ini, meskipun bisa mengkategorikan departemen dan memberikan akses level kepada group member spesifik, tapi belum terdapat fitur yang sub topik untuk akses level pada group user tertentu meskipun bisa menggunakan Custom Departement Filed namun tetap saja email notifikasi di broadcast ke semua member yang berada pada 1 departemen tersebut.

Jika dibandingkan dengan appikasi sejenis Kayako dengan lisensi berkisar \$39/bulan aplikasi ini sudah sangat bagus dan sangat cocok digunakan untuk internal support system (offline) maupun eksternal support system (online).

Situr Resmi <http://www.accord5.com/trellis>

Download : <http://www.accord5.com/download>

b. Zendesk

Zendesk adalah sistem tiket berbasis web yang menawarkan solusi pintar untuk masalah manajemen tiket dan help desk. Integrasi TeamViewer memperluas produk ini dengan fitur antri layanan TeamViewer yang berguna dan memungkinkan pengguna terhubung dengan dan membantu pelanggan langsung dari tiket.

Dengan TeamViewer teintegrasi secara langsung dengan Zendesk, pengguna dapat mengelola dan memulai semua kasus dukungan dari satu tempat. Dengan mudah terhubung dengan pelanggan dari dalam tiket Zendesk. Pelanggan memulai sesi hanya dengan satu klik mudah dan tidak harus khawatir dengan pengunduhan, sesi, ID atau apa pun.

2.1.2 Fitur Helpdesk

Dari dua jenis helpdesk yang telah di bahas di atas penulis merangkum seluruh fitur – fitur yang terdapat pada helpdesk pada umumnya. antara lain sebagai berikut:

1. Account

Pada helpdesk terdapat beberapa macam tingkatan hak akses, seperti manager, user, dan admin. Bisa juga di tambahkan dengan yang lainnya tergantung kebutuhan.

2. Pembagian Tiket Berdasarkan Departemen

Dengan feature ini maka semua departement dapat menerima permintaan dukungan dan tidak tertutup untuk bagian IT saja. Misalnya departement Supply Chain Management yang menangani masalah inventaris barang dan sirkulasi barang dapat menggunakan aplikasi ini untuk menangani permintaan barang dari departement atau cabang kantor lain.

3. Prioritas Tiket

Tiket dapat dibuat dan diberikan prioritas apakah tiket tersebut bersifat penting atau biasa saja. Dengan demikian bagian yang menerima tiket dapat memberikan dukungan dengan tepat sesuai prioritas tingkat kepentingan.

4. Pengalihan Tiket

Tiket dapat dialihkan kepada staff tertentu sehingga tiket tersebut dapat ditangani oleh staff tersebut dan bertanggung jawab atas penyelesaian tiket tersebut.

5. Pengaturan Hak Akses Berdasarkan Grup

Tingkat akses berlevel yang dibagi berdasarkan group sehingga memiliki tingkat keamanan penggunaan yang baik.

6. RSS Tiket

Pengguna dengan mudah membuat RSS feed pada tiket yang menjadi tanggung jawab pengguna. Dengan demikian pengguna dengan mudah mengecek apakah ada tiket baru yang ditujukan pada pengguna tersebut dengan cepat.

7. Multi Language

Secara default aplikasi helpdesk memiliki tampilan dengan bahasa Inggris. Namun dengan mudah pengguna dapat menerjemahkan aplikasi ini ke bahasa lain yang pengguna inginkan. Cukup dengan mengisi terjemahan dari kalimat-kalimat yang digunakan dalam aplikasi.

8. Knowledge Base

Memiliki knowledge base atau basis pengetahuan yang terintegrasi. Hal ini memudahkan kita dalam mencari langkah penyelesaian sebelum memasukkan tiket. Dengan feature yang memberikan saran solusi yang tersedia sebelum pengguna memasukkan tiket akan membantu meningkatkan efisiensi kerja karena mengurangi frekuensi dalam menangani masalah yang sama.

9. Canned Replies

Semisal ada pertanyaan atau pengaduan yang sama dengan pertanyaan-pertanyaan sebelumnya yang bersifat umum. Maka feature ini bisa digunakan tanpa kita perlu mengetikkan ulang jawaban atas pertanyaan itu secara langsung (direct message).

Itulah beberapa contoh fitur – fitur dari aplikasi helpdesk system, dan masih banyak lagi seperti, Manage articles and categories, Article ratings and comments, Smart display of KB articles during ticket submission, Pipe support emails into your helpdesk, Email notifications, dan lain – lain.

2.1.3 Fitur Helpdesk TVRI SULUT

Dari sekian banyak fitur yang telah di bahas pada sub bab sebelumnya, ada beberapa fitur yang akan di terapkan pada Helpdesk LPP TVRI SULUT dikarenakan fitur – fitur tersebut adalah yang apling utama dan yang paling di butuhkan dalam internal perusahaan yang bergerak di bidang penyiaran public ini, antara lain :

- Account
- Pembagian Tiket Berdasarkan Departemen
- Skala Prioritas Tiket
- Pengalihan Tiket
- Dan Pengaturan Hak Akses

2.2 Hipertext Processor (PHP)

PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client). Pada awalnya PHP merupakan singkatan dari Personal Home Page. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun

perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

Saat ini PHP adalah singkatan dari PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri PHP: Hypertext Preprocessor. PHP dapat digunakan dengan gratis dan bersifat Open Source. PHP dirilis dalam lisensi PHP License, sedikit berbeda dengan lisensi GNU General Public License (GPL) yang biasa digunakan untuk proyek Open Source. Kemudahan dan kepopuleran PHP sudah menjadi standar bagi programmer web di seluruh dunia.

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML dan CSS ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap. Sebagai perbandingan, website dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Website dinamis juga bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari user, memproses form, dll.

Untuk pembuatan web, kode PHP biasanya di sisipkan kedalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai Scripting Language atau bahasa pemrograman script. Sebagai contoh penggunaan PHP, misalkan kita ingin membuat list dari nomor 1 sampai nomor 10. Dengan menggunakan HTML murni, kita bisa membuatnya secara manual seperti kode berikut ini:


```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Contoh list dengan HTML</title>
5   </head>
6   <body>
7     <h2>Daftar Absensi Mahasiswa</h2>
8     <ol>
9       <li>Nama Mahasiswa ke 1</li>
10      <li>Nama Mahasiswa ke-2</li>
11      <li>Nama Mahasiswa ke-3</li>
12      <li>Nama Mahasiswa ke-4</li>
13      <li>Nama Mahasiswa ke-5</li>
14      <li>Nama Mahasiswa ke-6</li>
15      <li>Nama Mahasiswa ke-7</li>
16      <li>Nama Mahasiswa ke-8</li>
17      <li>Nama Mahasiswa ke-9</li>
18      <li>Nama Mahasiswa ke-10</li>
19    </ol>
20  </body>
21 </html>

```

Gambar 2.2 Contoh Skrip HTML

Halaman HTML tersebut dapat dibuat dengan mudah dengan cara men-copy-paste tag `` sebanyak 10 kali dan mengubah sedikit angka-angka no urut di belakangnya. Namun jika yang kita inginkan adalah menambahkan list tersebut menjadi 100 atau 1000 list, cara *copy-paste* tersebut menjadi tidak efektif. Jika menggunakan PHP, kita tinggal membuat perulangan for sebanyak 1000 kali dengan perintah yang lebih singkat seperti berikut ini:

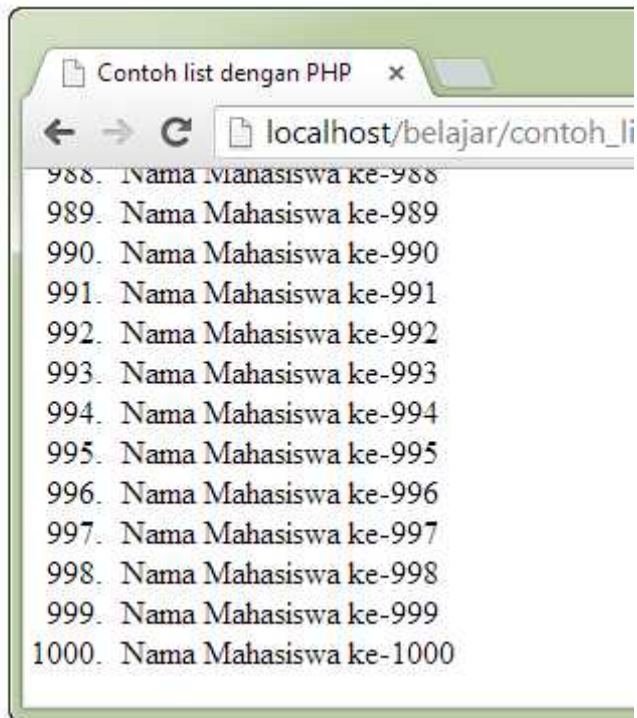
```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Contoh list dengan PHP</title>
5   </head>
6   <body>
7     <h2>Daftar Absensi Mahasiswa</h2>
8     <ol>
9       <?php
10        for ($i= 1; $i <= 1000; $i++)
11          {
12            echo "<li>Nama Mahasiswa ke-$i</li>";
13          }
14        ?>
15     </ol>
16 </body>
17 </html>

```

Gambar 2.3 Contoh Skrip PHP

Dengan menggunakan kode baris yang bahkan lebih sedikit, kita dapat membuat list tersebut menjadi 1000 kali, bahkan 100.000 kali dengan hanya mengubah sebuah variabel `$i`. berikut adalah hasil skrip pada gambar 2.3 di atas :



Gambar 2.4 Hasil Skrip PHP

PHP tidak hanya dapat melakukan pengulangan tersebut, masih banyak hal lain yang bisa kita lakukan dengan PHP, seperti menginput data ke database, menghasilkan gambar, menkonversi halaman text menjadi PDF, management cookie dan session, dan hal lainnya.

2.2.1 Sejarah Singkat PHP

PHP bermula pada tahun 1994 ketika programmer kelahiran Denmark yang sekarang berdomisili di Canada, **Rasmus Lerdorf** membuat sebuah script (kode program) dengan bahasa Perl untuk web pribadinya. Salah satu kegunaan script ini adalah untuk menampilkan resume pribadi dan mencatat jumlah pengunjung ke sebuah website. Dengan alasan untuk meningkatkan performa, Rasmus Lerdorf kemudian membuat ulang kode program tersebut dalam bahasa C. Ia juga mengembangkannya lebih lanjut sehingga memiliki script tersebut memiliki kemampuan untuk memproses form HTML dan berkomunikasi dengan database. Lerdorf menyebut kode program ini sebagai Personal Home Page/Forms Interpreter atau PHP/FI. Inilah asal mula penamaan PHP digunakan, PHP/FI dapat

digunakan untuk membuat aplikasi web dinamis sederhana. Lerdorf kemudian merilis kode tersebut ke publik dengan sebutan Personal Home Page Tools (PHP Tools) version 1.0. Perilisan ini diumumkan pada 8 Juni 1995 di alamat comp.infosystems.www.authoring.cgi, sebuah group diskusi Usenet.

2.3 MySQL

MySQL dapat digunakan untuk membuat dan mengola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database. MySQL merupakan sistem manajemen database yang bersifat at relational. Artinya data-data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat.

MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah Structured Query Language (SQL) untuk mengelola database-database yang ada di dalamnya. Hingga kini, MySQL sudah berkembang hingga versi 5. MySQL 5 sudah mendukung trigger untuk memudahkan pengelolaan tabel dalam database.

MySQL sebenarnya produk yang berjalan pada platform Linux. Karena sifatnya yang open source, dia dapat dijalankan pada semua platform baik Windows maupun Linux. MySQL juga merupakan program pengakses database yang bersifat jaringan sehingga dapat digunakan untuk aplikasi Multi User (banyak pengguna). MySQL menggunakan bahasa Query standar yang dimiliki SQL (Struktur Query Language). MySQL tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (interface). MySQL dapat didukung hampir semua oleh program aplikasi baik yang open source seperti PHP maupun yang tidak. Secara umum akses ke database harus melalui tiga tahap, yaitu :

- Koneksi ke database.
- Query ke database.
- Pemutusan koneksi dari database.

Sedangkan fungsi-fungsi PHP yang digunakan untuk koneksi dengan database adalah :

- `Mysql_connect();`
- `Mysql_pconnect();`

`Mysql_select_db();`

2.3.1 Konektifitas PHP dan MySQL

Bagian ini membahas pengaturan hubungan antara PHP dan MySQL secara general. Dimana untuk dapat berhubungan dengan mysql, PHP harus membuka hubungan ke server MySQL yaitu dengan:

`Mysql_connect (nama_host, nama_user, password);`

Keterangan :

- **Nama_host** adalah lokasi tempat MySQL dipublikasikan.
- **Nama_user** adalah string dengan isi data nama user yang terdaftar dalam MySQL yang digunakan untuk mengakses data pada MySQL.
- **Password** adalah string yang berisikan password yang digunakan untuk mengakses database.

Setelah tersambung, maka user memilih database yaitu dengan:

`Mysql_select_db (nama_database);`

Untuk mengeksekusi query-query yang ada pada MYSQL namun tidak dimiliki oleh PHP digunakan fungsi:

`Mysql_query (query);`

Untuk mengambil hasil query yang dilakukan oleh `mysql_query ()` digunakan fungsi:

mysql_fetch_row (result_id) atau

mysql_fetch_array (result_id)

Pada `mysql_fetch_array ()` hasil yang diperoleh dalam bentuk array asosiatif. Setelah penggunaan database mysql selesai, maka koneksi diputus dengan menggunakan:

mysql_close ();

2.4 Hypertext Markup Language (HTML)

HTML adalah singkatan dari HyperText Markup Language yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web Internet (Browser). HTML dapat juga digunakan sebagai link - link antara file-file dalam situs atau dalam komputer dengan menggunakan localhost, atau link yang menghubungkan antar situs dalam dunia internet. Supaya dapat menghasilkan tampilan wujud yang terintegrasi pemformatan hiperteks sederhana ditulis dalam berkas format ASCII sehingga menjadi halaman web dengan perintah-perintah HTML. HTML merupakan sebuah bahasa yang bermula bahasa yang sebelumnya banyak dipakai di dunia percetakan dan penerbitan yang disebut Standard Generalized Markup Language (SGML).

Sekarang ini HTML merupakan standar Internet yang dikendalikan dan didefinisikan pemakaiannya oleh World Wide Web Consortium (W3C). Pada tahun 1989, HTML dibuat oleh kolaborasi Berners-lee Robert dengan Caillau TIM pada saat mereka bekerja di CERN (CERN merupakan lembaga penelitian fisika energi tinggi di Jenewa) HTTP atau Hypertext Transfer Protokol merupakan protokol yang digunakan untuk mentransfer data atau document yang berformat HTML dari web server ke web browser. Dengan HTTP inilah yang memungkinkan kita menjelajah internet dan melihat halaman web.

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang

merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser. HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: **TAMPIL TEBAL**, maka penulisannya dilakukan dengan cara: `TAMPIL TEBAL`. Tanda `` digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda `` untuk menonaktifkan cetak tebal tersebut. Secara garis besar, terdapat 4 jenis elemen dari HTML:

1. Structural, tanda yang menentukan level atau tingkatan dari sebuah teks contoh, `<h1>Golf</h1>` akan memerintahkan browser untuk menampilkan “Golf” sebagai teks tebal besar yang menunjukkan sebagai Heading 1
2. Presentational, tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut contoh, `boldface` akan menampilkan bold. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks,
3. Hypertext, tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain contoh, `IlmuKita` akan menampilkan IlmuKita sebagai sebuah hyperlink ke URL tertentu),
4. Elemen widget yang membuat objek-objek lain seperti tombol (`<button>`), list (``), dan garis horizontal (`<hr>`).

Selain markup presentational, markup yang lain tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan tag HTML untuk

menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan Cascading Style Sheets.

2.5 Cascading Style Sheet (CSS)

CSS adalah singkatan dari Cascading Style Sheets. Berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil di halaman web. Perancangan desain text dapat dilakukan dengan mendefinisikan fonts (huruf) , colors (warna), margins (ukuran), latar belakang (background), ukuran font (font sizes) dan lain-lain. Elemen-elemen seperti colors (warna) , fonts (huruf), sizes (ukuran) dan spacing (jarak) disebut juga “styles”. Cascading Style Sheets juga bisa berarti meletakkan styles yang berbeda pada layers (lapisan) yang berbeda. CSS terdiri dari style sheet yang memberitahukan browser bagaimana suatu dokumen akan disajikan. Fitur-fitur baru pada halaman web lama dapat ditambahkan dengan bantuan style sheet. Saat menggunakan CSS, Anda tidak perlu menulis font, color atau size pada setiap paragraf, atau pada setiap dokumen. Setelah Anda membuat sebuah style sheet, Anda dapat menyimpan kode tersebut sekali saja dan dapat kembali menggunakannya bila diperlukan.

CSS (Cascading Style Sheet) adalah salah satu bahasa desain web (style sheet language) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda markup language. Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL bahkan ANDROID. CSS dibuat untuk memisahkan konten utamadengan tampilan dokumen yang meliputi layout, warna da font. Pemisahan ini dapat meningkatkann daya akses konten pada web, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi darisebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi halaman untuk sebuah formatting dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tableless pada desain web.

CSS juga memungkinkan sebuah halaman untuk ditampilkan dalam berbagai style dengan menggunakan metode pembawaan yang berbeda pula, seperti on-screen, in-print, by voice, dan lain-lain. Sementaraitu, pemilik konten web bisa menentukan link yang menghubungkan konten dengan file CSS. Tujuan utama CSS diciptakan untuk membedakan konten dari dokumen dan dari tampilan dokumen, dengan itu, pembuatan ataupun pemrograman ulang web akan lebih mudah dilakukan. Hal yang termasuk dalam desain web diantaranya adalah warna, ukura dan formatting. Dengan adanya CSS, konten dan desain web akan mudah dibedakan, jadi memungkinkan untuk melakukan pengulangan pada tampilan-tampilan tertentu dalam suatu web, sehingga akan memudahkan dalam membuat halaman web yang banyak, yang pada akhirnya dapat memangkas waktu pembuatan web.

fungsi utama css adalah merancang, merubah, mendisain, membentuk halaman wesite(blog juga website). dan isi dari halaman website adalah tag-tag html, logikanya css itu dapat merubah tag-tag html (yang sederhana) sehingga menjadi lebih fungsional dan menarik.

2.5.1 Sejarah Singkat CSS

Sampai saat ini, CSS sudah sampai pada versinya yang ke 3. Berikut kita akan membahasnya dari versi yang pertama :

1. CSS 1

Pada tanggal 17 Agustus 1996 World Wide Web Consortium (W3C) menetapkan CSS sebagai bahasa pemrograman standard dalam pembuatan web. Tujuannya adalah untuk mengurangi pembuatan tag-tag baru oleh Netscape dan Internet Explorer, karena kedua browser tersebut sedang bersaing mengembangkan tag sendiri untuk mengatur tampilan web.

CSS 1 mendukung pengaturan tampilan dalam hal :

- a. Font (Jenis ketebalan).
- b. Warna, teks, background dan elemen lainnya.
- c. Text attributes, misalnya spasi antar baris, kata dan huruf.
- d. Posisi teks, gambar, table dan elemen lainnya.
- e. Margin, border dan padding.

2. CSS 2

Pada tahun 1998, W3C menyempurnakan CSS tahap awal dengan menciptakan standard CSS 2 yang menjadi standard hingga saat ini. Pada level CSS 2 ini, dimasukkan semua atribut dari CSS 1 dan diperluas dengan penekanan pada International Accessibility and Capabilty khususnya media-specific CSS. CSS 2 dikembangkan untuk memenuhi kebutuhan terhadap format dokumen agar bisa ditampilkan di printer.

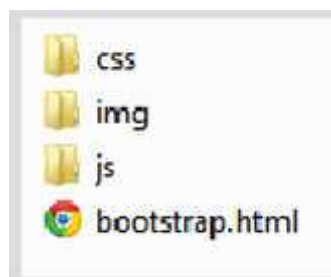
3. CSS 3

CSS 3 adalah versi terbaru dari CSS yang mampu melakukan banyak hal dalam mendesain website. CSS 3 dapat melakukan animasi pada halaman website, diantaranya animasi warna dan animasi 3D. Dengan CSS 3 desainer dimudahkan dalam hal kompatibilitas websitenya pada smartphone dengan dukungan fitur baru yakni media query. Selain itu, banyak fitur baru pada CSS 3 yaitu : Multiple background, border-radius, drop-shadow, border-image, CSS-Math dan CSS Object Model. Pada CSS 3 juga terdapat beberapa fitur baru, antara lain :

- a. Animasi, sehingga pembuatan animasi tidak memerlukan program sejenis Adobe Flash dan Microsoft Silverlight.
- b. Beberapa efek teks, seperti teks berbayang, kolom koran dan "Word-Wrap".
- c. Beberapa efek pada kotak, seperti kotak yang ukurannya dapat diubah-ubah, transformasi 2 dimensi dan 2 dimensi, sudut-sudut yang tumpul dan bayangan.

2.6 Bootstrap twitter

Bootstrap adalah sebuah framework CSS yang menyediakan kumpulan komponen-komponen antarmuka dasar pada web yang telah dirancang sedemikian rupa untuk digunakan bersama-sama. Selain komponen antarmuka, Bootstrap juga menyediakan sarana untuk membangun layout halaman dengan mudah dan rapi, serta modifikasi pada tampilan dasar HTML untuk membuat seluruh halaman web yang dikembangkan senada dengan komponen-komponen lainnya. Bootstrap dibuat untuk memberikan sekumpulan perangkat yang dapat digunakan untuk membangun website sederhana dengan mudah. Untuk menggunakan Bootstrap pada aplikasi kita, tentunya kita harus memiliki Bootstrap terlebih dahulu. Buka website resmi Bootstrap, <http://getbootstrap.com/> dan klik tombol "Download Bootstrap" untuk memulai download bootstrap. Selesai download, anda akan memiliki file bootstrap.zip yang berisi file-file yang dibutuhkan untuk menggunakan Bootstrap. Ekstrak file tersebut pada tempat yang sama dengan bootstrap.html, sehingga sekarang kita memiliki file seperti berikut:



Gambar 2.5 Hasil Download

2.7 STRUKTUR ORGANISASI

Berhubung aplikasi helpdesk ini akan melakukan pengujian atau testing di LPP TVRI SULUT maka akan sedikit dibahas mengenai struktur organisasinya, agar dapat dilihat langsung permasalahan yang terjadi.

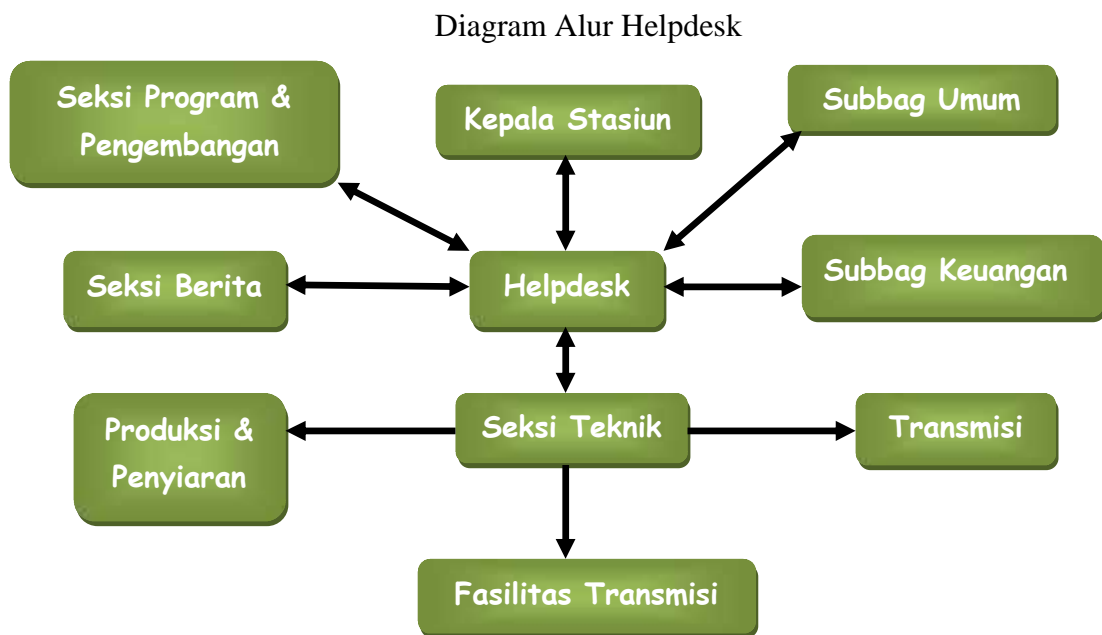


Gambar 2.6 Struktur Organisasi

Dari struktur organisasi diatas dapat dilihat, bahwa TVRI Stasiun Sulawesi Utara dipimpin oleh seorang KEPSTA (Kepala Stasiun) eselon IIIa, memiliki tugas pokok dan fungsi melaksanakan penyelenggaraan penyiaran televisi publik sesuai dengan kebijakan umum atau pun khusus yang ditetapkan dewan direksi LPP TVRI sesuai dengan peraturan dewan direksi LPP TVRI nomor : 115/PRTD/DIREKSI/TVRI/2006. Dalam menjalankan tugasnya Kepala stasiun dibantu 5 orang Kepala seksi dan Kasubbag eselon IVa yang masing – masing memiliki bidang tugas sesuai dengan fungsinya untuk menjalankan pelaksanaan kegiatan program dan pengembangan usaha, pelaksanaan produksi dan penyiaran berita, pelaksanaan kegiatan bidang teknik, pelaksanaan bidang umum dan pelaksanaan bidang keuangan. Kepala seksi program dan pengembangan usaha membawahi dua pejabat struktural eselon IVb yaitu Kepala subseksi produksi dan penyiaran dan subseksi pengembangan usaha. Sedangkan Kepala Seksi Teknik membawahi tiga pejabat struktural eselon IVb yaitu Kepala subseksi teknik studio

dan produksi siaran, Kepala subseksi transmisi, serta Kepala subseksi fasilitas transmisi.

Sesuai dengan struktur organisasi yang tergambar di atas, maka penulis menggambarkan diagram alur Helpdesk system pada LPP TVRI SULUT sebagai berikut :



Gambar 2.7 Diagram Alur Helpdesk

BAB III

PERANCANGAN HELPDESK INTERNAL SYSTEM TVRI SULUT

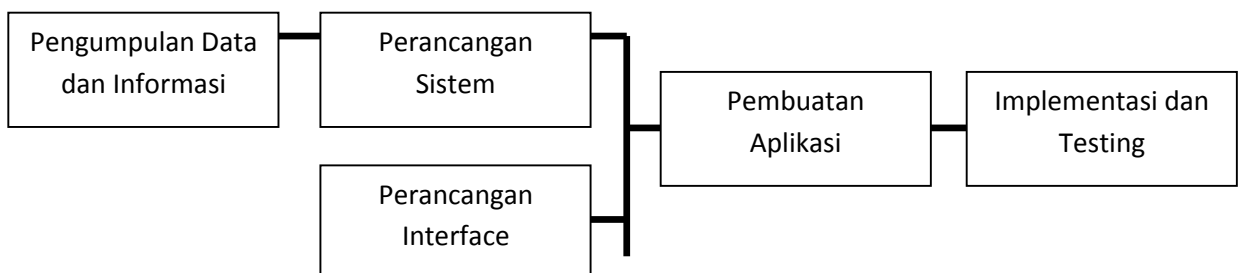
3.1 Waktu dan Tempat

- Lokasi : LPP TVRI SULUT
- Waktu : Mei – Juli 2015

3.2 Kerangka Konseptual

Kerangka konseptual merupakan suatu bentuk kerangka berpikir yang dapat digunakan sebagai pendekatan dalam memecahkan masalah. Biasanya kerangka penelitian ini menggunakan pendekatan ilmiah dan memperlihatkan hubungan antar variable dalam proses analisisnya.

Adapun gambar kerangka berpikir dalam penelitian ini dapat dilihat pada gambar berikut.



Gambar 3.1 Kerangka Konseptual

1. Pengumpulan Data dan Informasi

Penelitian ini diawali dengan pengumpulan informasi mengenai rumusan masalah penggunaan helpdesk pada TVRI SULUT, bisa atau tidaknya aplikasi ini digunakan, secara umum perusahaan yang bergerak di bidang penyiaran ini sangat kurang menghadapi permasalahan dengan konsumen, karna tidak lain jasa yang di tawarkan perusahaan ini yaitu jasa penyiaran, dan hanya beroperasi dari pukul 16.00 s/d 20.00 WITA. Maka dari itu penulis mengalihkan helpdesk dari yang umumnya bersifat external antara perusahaan dengan konsumen, menjadi internal yaitu antara sesama

pegawai perusahaan, serta pengambilan beberapa sampel atau contoh kasus yang terjadi di perusahaan dan coba di terapkan dalam helpdesk.

2. Perancangan System

Perancangan system merupakan rancang bangun system yang dibuat. Perancangan yang dibuat berupa helpdesk system yang bersifat internal perusahaan. Perancangan system ini dibuat berdasarkan informasi yang telah didapat melalui research aplikasi yang banyak digunakan oleh perusahaan pada umumnya.

3. Perancangan Interface

Perancangan antarmuka (*Interface*) merupakan rancangan dasar aplikasi, mengenai tentang seperti apa bentuk aplikasi tersebut sampai kepada pengguna.

4. Pembuatan Aplikasi Helpdesk System

Pembuatan aplikasi menggunakan pemograman PHP, dengan tampilan yang menggunakan HTML dan sedikit dipadukan dengan CSS dan Bootsrap Twitter.

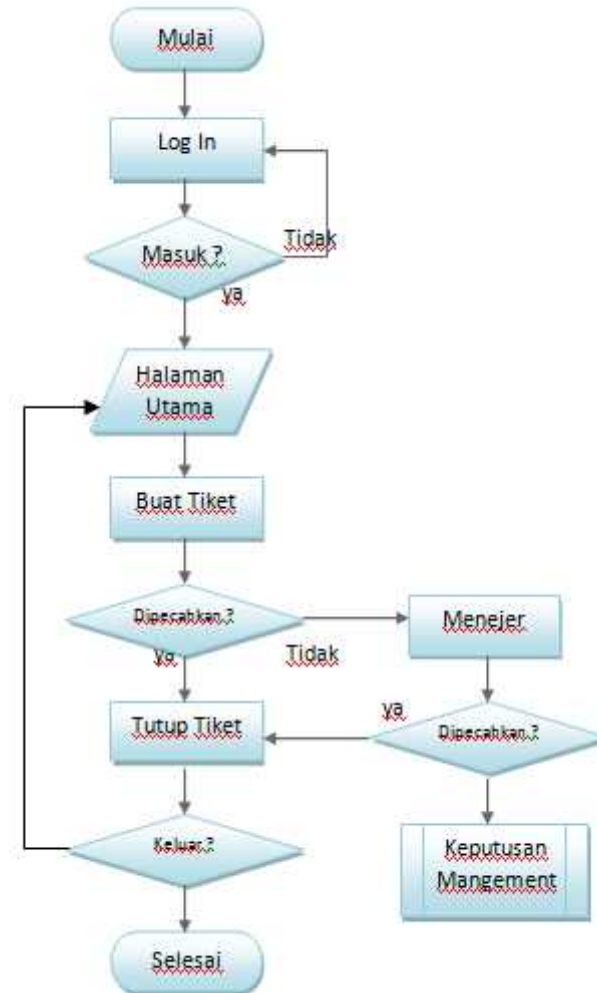
5. Implementasi dan Testing

Aplikasi helpdesk system yang telah dibuat kemudian akan diuji dan diimplementasikan dengan coba mengoperasikan aplikasi tersebut, apakah semua perintah – perintah berjalan dengan semestinya.

3.3 Diagram Alur (Flowchart)

Pada pembuatan aplikasi ini, dibutuhkan suatu teknik perancangan yang mempunyai struktur yang baik, biasanya diawali dengan pembuatan diagram alur (flowchart). Diagram alur digunakan untuk menggambarkan terlebih dahulu apa yang harus dikerjakan sebelum mulai merancang atau membuat suatu system seperti yang akan dijelaskan dibawah ini. Berikut adalah diagram alur (flowchart) dari aplikasi helpdesk system ini.

- Flowchart pembuatan Tiket Helpdesk

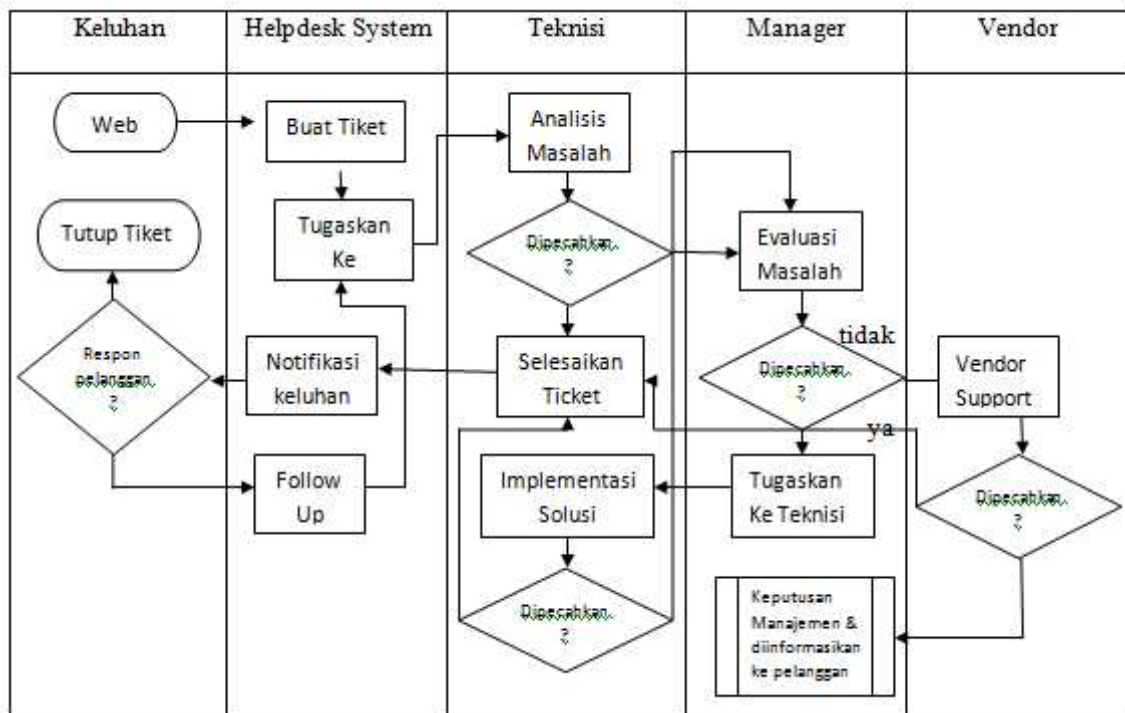


Gambar 3.2 Flowchart Tiket

Penjelasan Flowchart pembuatan tiket :

1. Dimulai dengan start dan langsung masuk ke halaman log in. user memasukkan username dan password, apabila username dan password valid, maka akan langsung masuk ke halaman beranda, namun apabila username dan password tidak valid akan kembali ke halaman log in sekaligus dengan pemberitahuan tentang kesalahan log in.

2. Di halaman utama atau beranda, pengguna bisa langsung membuat tiket, dan tiket akan langsung diproses, apabila tiket berhasil dipecahkan maka tiket akan langsung ditutup, namun apabila tidak maka tiket akan langsung diarahkan kepada atasan untuk dimintai solusi.
3. Setelah diarahkan ke atasan atau manajer, kemudian manajer menganalisa dan mengevaluasi kembali permasalahan yang terjadi ini, apabila berhasil terpecahkan maka tiket akan langsung ditutup, namun apabila tidak, kembali lagi kepada kebijakan pimpinan dan perusahaan, untuk diambil jalan tengah agar tidak merugikan satu pihak saja.
4. Setelah tiket ditutup pengguna bisa langsung ke luar atau ataupun bisa langsung kembali ke halaman awal, atau beranda untuk membuat tiket lagi, dan proses seperti di atas akan kembali berulang



Gambar 3.3 Alur Kerja Umum

Penjelasan Alur Kerja Umum TVRI SULUT

1. Kita mulai flowchart ini dari sisi kiri atas, tepatnya pada kolom keluhan dan pada start web. Pengguna mengakses dan login ke aplikasi yang saat ini masih di sediakan dalam bentuk website, sesuai dengan username dan password yang telah di buat oleh admin
2. Selanjutnya membuat tiket keluhan, tiket ini berupa formulir yang wajib di isi, dalam tiket ini terdapat berbagai informasi keluhan serta informasikan tentang siapa yang melaporkan keluhan dan kepada siapa keluhan itu disampaikan.
3. Setelah semua field dalam formulir tiket di isi dan di submit, maka tiket tersebut akan langsung sampai kepada pengguna lain yang di tujukan, selanjutnya pengguna atau divisi tempat tiket itu dituju, akan menganalisis masalah atau keluhan itu, dan mencari solusi.
4. Apabila tiket masalah berhasil dipecahkan, maka teknisi atau pegawai tersebut bisa langsung mengganti status tiket menjadi complete atau telah di terselesaikan, dan langsung di kirimkan kembali kepada pengguna yang melaporkan tadi.
5. Selanjutnya pengguna yang melaporkan melihat apakah permintaan sudah sesuai dengan apa yang di requestnya atau belum. Jika ya, maka pelapor bisa langsung meng close atau menutup tiket tersebut, namun apabila tidak, maka konsumen bisa men follow up atau re-open kembali tiket, dan akan kembali kepada teknisi atau pegawai yang bersangkutan.
6. Kemudian teknisi kembali menganalisis masalah tersebut, apabila terpecahkan, prosesnya akan kembali seperti pada penjelasan nomor 4, namun apabila tidak atau teknisi tetap tidak mampu memecahkan masalah ini, maka keluhan atau masalah ini akan teknisi atau pegawai arahkan kepada kepala divisi atau manajer, kemudian kepala divisi menganalisa permasalahan itu, apabila terpecahkan, maka kepala divisi akan menugaskan penyelesaian tersebut kepada teknisi atau pegawai, selanjutnya pegawai mengimplementasikan sesuai dengan instruksi yang

di dapat dari kepala divisi, dan kemudian proses kembali seperti yang telah dijelaskan di atas tadi.

7. Namun apabila masalah itu kembali lagi manajer atau kepala divisi yang artinya masalah tersebut belum bisa dipecahkan juga, maka jika permasalahan tersebut bersangkutan dengan produk barang atau jasa pihak ketiga yaitu eksternalnya perusahaan, maka kepala divisi bisa langsung menghubungi vendor penyedia layanan jasa atau produsen tersebut dengan sepengetahuan kepala stasiun.
8. Apabila vendor berhasil memecahkan, maka proses akan kembali di jalankan sebagaimana yang telah di jelaskan di atas. Namun apabila tidak ditemukan juga solusinya, maka akhirnya kembali kepada keputusan manajemen, dan di informasikan kepada pelanggan.

3.4 Perancangan

Pada tahap perancangan ini, penulis merancang aplikasi ini dengan menggunakan bantuan dari beberapa buku yang telah di baca, dan beberapa hasil searching internet, serta tidak lupa pula untuk berkonsultasi dengan dosen pembimbing dan beberapa orang yang mempunyai kapasitas dan kapabilitas di bidangnya. Pada perancangan ini penulis membutuhkan beberapa tools, atau software pembantu dalam merancang, mengeksekusi serta melakukan testing dan implementasi program aplikasi ini, diantaranya :

1. Xampp v3.0.12

Yaitu aplikasi localhost untuk menjalankan perintah – perintah PHP, Mysql, ajax dll. serta sebagai tempat untuk melakukan testing program dan implementasi program.

2. Dreamweaver 8

Dreamweaver adalah aplikasi atau software untuk mengedit berbagai macam script PHP, HTML, CSS, dan Bootstrap Twitter yang coba sedikit di implementasikan dalam aplikasi ini. Aplikasi ini sangat membantu karena dalam proses pengkodean atau scripting tidak sedikit terjadi error atau

kesalahan, mulai dari notice sampai dengan fatal error, dengan adanya aplikasi ini penulis bisa langsung mengedit atau memperbaiki segala macam kesalahan dalam pengkodean, dan membantu penulis dalam menemukan dengan cepat dimana letak kesalahan tersebut. Dreamwaver 8 ini juga sudah bisa mengedit tampilan HTML dan CSS secara visual meskipun tetap harus memperhatikan kode atau script suatu program.

3. Google Chrome

Sebenarnya kita bisa menggunakan berbagai macam browser yang kita inginkan, namun kali ini penulis menggunakan browser google chrome karna tampilannya yang simple, sederhana, dan mudah digunakan.

Selain itu juga dalam perancangan aplikasi ini penulis membaginya menjadi tiga bagian, yaitu perancangan database, interface (antarmuka), dan pengkodean atau scripting program, yang akan di bahas satu persatu.

3.4.1 Perancangan Database

Tahap perancangan database ini membahas tentang tabel – tabel apa saja yang diperlukan, dan query apa saja yang diperlukan dalam tabel tersebut.

1. Tabel Divisi

Tabel divisi ini berfungsi untuk menampung data – data sebuah divisi, berikut adalah gambaran divisi tersebut.

Field	Type	Length	Index	A_I (Auto Increment)
iddivisi	INT	11	Primary	Yes
namadivisi	VARCHAR	50		
Telp	VARCHAR	20		
PIC	VARCHAR	50		

Tabel 3.1 Tabel Divisi

2. Tabel SLA

Tabel SLA atau tabel skala prioritas, dimana user bisa menentukan seberapa pentingnya keluhan yang di ajukan dan dalam tabel ini terdapat waktu, yaitu kapan seharusnya keluhan ini telah selesai. Berikut ada tabel SLA dan apa – apa saja yang termasuk dalam tabel itu.

Field	Type	Length	Index	A_I (Auto Increment)
slaid	INT	11	Primary	Yes
namasla	VARCHAR	30		
responstime	INT	11		
resolutiontime	INT	11		

Tabel 3.2 Tabel SLA

3. Tabel News

Tabel ini berisikan berita, atau pengumuman yang disampaikan oleh admin, yang nantinya akan muncul di beranda tiap – tiap user, agar user bisa langsung mengetahui apabila ada gangguan umum yang terjadi, sehingga user tidak perlu repot membuat tiket.

Field	Type	Length	Index	A_I
Id	INT	11	Primary	Ya
Newsdate	INT	11		
Title	VARCHAR	80		
Detail	TEXT			
Createdby	VARCHAR	50		
Createdon	INT			
Expired	INT			

Tabel 3.3 Tabel News

4. Tabel Tiket

Tabel tiket ini merupakan formulir yang wajib di isi oleh pengguna apabila ingin menyampaikan keluhan atau pertanyaan.

Field	Type	Lenght	Index	A_I
id	INT	11	Primary	Yes
tiketnumber	VARCHAR	20		
SLA	VARCHAR	10		
iddivisi	VARCHAR	10		
reporteddate	INT	11		
reportedby	VARCHAR	50		
telp	VARCHAR	20		
problemsummary	VARCHAR	80		
Problemdetail	TEXT			
resolution	TEXT			
closedate	INT	11		

Tabel 3.4 Tabel Tiket

5. Tabel Users

Tabel ini berisikan informasi tentang user, serta level atau hak akses user

Field	Type	Lenght	Index	A_I
id	INT	11	Primary	Yes
username	VARCHAR	30		
password	VARCAHR	30		
level	VARCHAR	25		
fullname	VARCHAR	100		
Telp	VARCHAR	15		
confirmed	INT	11		

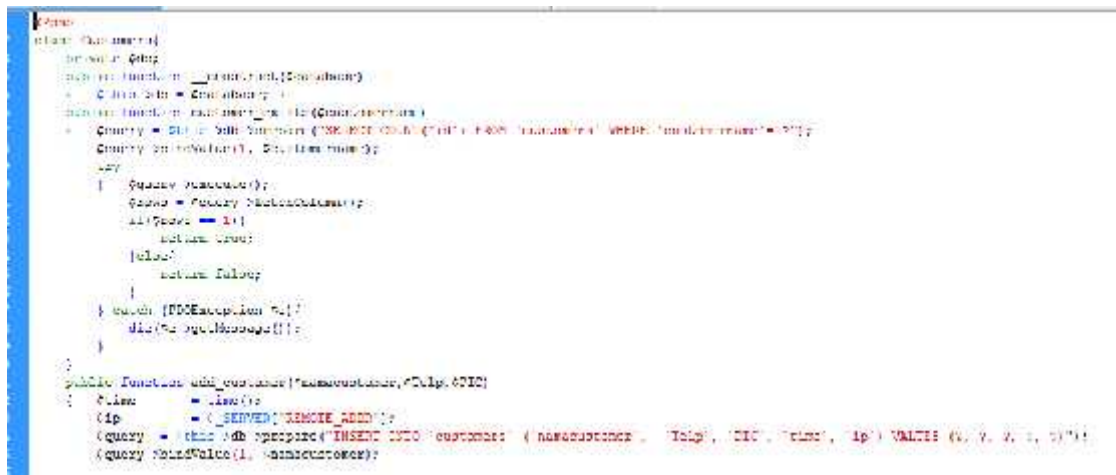
Tabel 3.5 Tabel Users

3.4.2 Perancangan Script

Dalam perancangan skrip ini, akan kita awali dengan menggunakan sedikit teknik pemrograman berbasis objek yang menggunakan class – class, dan kemudian akan di lanjutkan dengan scripting atau pengkodean perintah – perintah lainnya, dimana semua perintah dalam script – script ini akan saling terhubung. Script – script ini hanya akan ditampilkan dalam screenshot saja, karna akan sangat panjang, nanti lebih lengkapnya akan di lampirkan.

1. Customers.php

Skrip ini merupakan class dari divisi dimana skrip ini menampung semua perintah – perintah dari class divisi yang nantinya akan di panggil oleh skrip – lainnya. Berikut adalah script dari Customers.php



```
class Customers {
    private $db;
    public function __construct($koneksi) {
        $this->db = $koneksi;
    }
    public function add_customer($nama_customer, $no_telp, $alamat, $ip) {
        $query = $this->db->prepare("INSERT INTO customers (nama_customer, no_telp, alamat, ip) VALUES (?, ?, ?, ?)");
        $query->execute();
    }
    public function add_customer($nama_customer, $no_telp, $alamat, $ip) {
        $query = $this->db->prepare("INSERT INTO customers (nama_customer, no_telp, alamat, ip) VALUES (?, ?, ?, ?)");
        $query->execute();
    }
}
```

Gambar 3.4 Script Customers.php

2. General.php

File ini berisikan pengaturan perlindungan user berupa log out dan log in menggunakan perintah session

```
<?php
class General
{
    public function logged_in () {
        return(isset($_SESSION['loginid'])) ? true : false;
    }
    public function logged_in_protect() {
        if ($this->logged_in() == true) {
            header('Location: home.php');
            exit();
        }
    }
    public function logged_out_protect() {
        if ($this->logged_in() == false) {
            header('Location: index.php');
            exit();
        }
    }
}
?>
```

Gambar 3.5 Script General.php

3. HDNews.php

Skip ini berisikan perintah yang menyangkut dengan News atau berita pemberitahuan yang akan muncul di beranda users yang sebagaimana telah dibahas sebelumnya.

```
<?php
class HDNews
{
    public function __construct($database)
    {
        $this->db = $database;
    }
    public function add_news($title,$url,$url_thumb,$category,$category_id,$password)
    {
        $query = $this->db->prepare("INSERT INTO 'news' ('newsid','title','url','urlthumb','category','password') VALUES (?, ?, ?, ?, ?, ?)");
        $query->bindValue(1, $newsid);
        $query->bindValue(2, $title);
        $query->bindValue(3, $url);
        $query->bindValue(4, $urlthumb);
        $query->bindValue(5, $category);
        $query->bindValue(6, $password);
        $query->execute();
        $query->close();
        return($this->db->lastInsertId());
    }
    public function update_news($id,$title,$url,$urlthumb,$category,$password)
    {
        $query = $this->db->prepare("UPDATE 'news' SET 'title' = ?, 'url' = ?, 'urlthumb' = ?, 'category' = ?, 'password' = ? WHERE 'newsid' = ?");
        $query->bindValue(1, $title);
        $query->bindValue(2, $url);
        $query->bindValue(3, $urlthumb);
        $query->bindValue(4, $category);
        $query->bindValue(5, $password);
        $query->bindValue(6, $id);
        $query->execute();
        $query->close();
    }
}
```

Gambar 3.6 HDNews.php

4. SLA.php

SLA atau lebih kita kenal dengan skala prioritas ini, juga telah di bahas sebelumnya yaitu dimana mengatur tentang skala prioritas. Berikut adalah bebeapa tampilan skripnya.

```

<?php
class SLA:
private $db;
public function __construct($database)
| $this->db = $database;
public function __destruct(){}
public function __wakeup(){}
public function __sleep(){}
public function __serialize(){}
public function __unserialize(){}
public function __clone(){}
public function __toString(){}
public function __get($name)
| $query = $this->db->prepare("SELECT COUNT(*) FROM sla WHERE sla_id = ?");
| $query->bindValue(1, $slaid);
| try
| {
| $query->execute();
| $row = $query->fetch(PDO::FETCH_ASSOC);
| if($row == 1)
| return true;
| else
| return false;
| }
| catch (PDOException $e)
| die($e->getMessage());
}

public function add_sla($slaid,$nama,$responsibls,$resolusitime,$slawarning)
| $querystring = "INSERT INTO sla (slaid, nama, responsibls, resolusitime, slawarning) VALUES (:slaid, :nama, :responsibls, :resolusitime, :slawarning)";
| $query = $this->db->prepare($querystring);
| $query->bindValue(1, $slaid);
| $query->bindValue(2, $nama);
| $query->bindValue(3, $responsibls);
| $query->bindValue(4, $resolusitime);
| $query->bindValue(5, $slawarning);
| $query->execute();
| return $this->db->lastInsertId();
}
}

```

Gambar 3.7 SLA.php

5. Tiket.php

File ini berisikan tentang perintah – perintah yang bersangkutan dengan tiket yang nantinya akan di panggil oleh skrip lainnya.

```

<?php
date_default_timezone_set('Asia/Jakarta');
class Tiket:
private $db;
public function __construct($database)
| $this->db = $database;
public function __destruct(){}
public function __wakeup(){}
public function __sleep(){}
public function __serialize(){}
public function __unserialize(){}
public function __clone(){}
public function __toString(){}
public function __get($name)
| $query = $this->db->prepare("SELECT COUNT(*) FROM tiket WHERE tiket_id = ?");
| $query->bindValue(1, $ticketid);
| try
| {
| $query->execute();
| $row = $query->fetch(PDO::FETCH_ASSOC);
| if($row == 1)
| return true;
| else
| return false;
| }
| catch (PDOException $e)
| die($e->getMessage());
}

public function add_tiket($tiketnumber,$slaid,$iduser,$partid,$partiddate,$partidby,$partid,$partidsummary,$partidstatus,$tiketstatus,$partid,$partiddate)
| $querystring = "INSERT INTO tiket (tiketnumber, slaid, iduser, partid, partiddate, partidby, partid, partidsummary, partidstatus, tiketstatus, partid, partiddate) VALUES (:tiketnumber, :slaid, :iduser, :partid, :partiddate, :partidby, :partid, :partidsummary, :partidstatus, :tiketstatus, :partid, :partiddate)";
| $query = $this->db->prepare($querystring);
| $query->bindValue(1, $tiketnumber);
| $query->bindValue(2, $slaid);
| $query->bindValue(3, $iduser);
| $query->bindValue(4, $partid);
| $query->bindValue(5, $partiddate);
| $query->bindValue(6, $partidby);
| $query->bindValue(7, $partid);
| $query->bindValue(8, $partidsummary);
| $query->bindValue(9, $partidstatus);
| $query->bindValue(10, $tiketstatus);
| $query->bindValue(11, $partid);
| $query->bindValue(12, $partiddate);
| $query->execute();
| return $this->db->lastInsertId();
}
}

```

Gambar 3.8 Tiket.php

6. Users.php

Sebagaimana nama file ini, file ini berisikan tentang segala perintah – perintah yang bisa dilakukan oleh seorang user.

```
<?php
class Users{
    private $db;
    public function __construct(&$database)
    {
        $this->db = $database;
    }
    public function user_exists(&$username)
    {
        $query = $this->db->prepare("SELECT COUNT(*) FROM `users` WHERE `username` = ?");
        $query->bindValue(1, $username);
        try
        {
            $query->execute();
            $rows = $query->fetchColumn();
            if($rows == 1){
                return true;
            }else{
                return false;
            }
        } catch (PDOException $e){
            die($e->getMessage());
        }
    }

    public function email_exists($email)
    {
        $query = $this->db->prepare("SELECT COUNT(*) FROM `users` WHERE `email` = ?");
        $query->bindValue(1, $email);
        try{
            $query->execute();
            $rows = $query->fetchColumn();
        }
    }
}
```

Gambar 3.9 Users.php

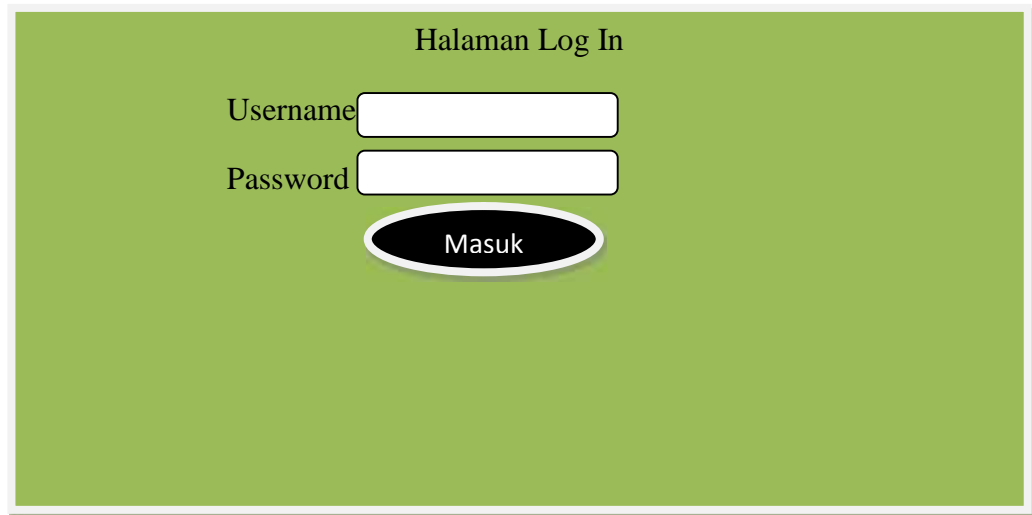
Selanjutnya masih banyak juga script yang dibutuhkan agar aplikasi ini berjalan sebagaimana mestinya, namun tidak semua akan di bahas satu persatu karna sebenarnya hanya dengan mengoperasikan aplikasi helpdesk ini kita akan langsung mengetahui script mana yang sedang berjalan, dan dimana script itu di gunakan tapi akan tetap penulis lampirkan dalam lampiran tugas akhir ini.

3.4.3 Perancangan Interface

Untuk perancangan interface ini penulis hanya akan membuat dua interface, dimana terdiri dari interface log in, dan home atau beranda. Log in berfungsi sebagai awal atau permulaan untuk masuk ke dalam system aplikasi ini, setelah itu akan langsung masuk kedalam halaman home, di mana halaman home tersebut

terdapat script PHP untuk menampilkan berbagai macam perintah yang kita namakan pada ini sebagai content, atau content.php. untuk lebih jelasnya akan di gambarkan pada gambar di bawah ini.

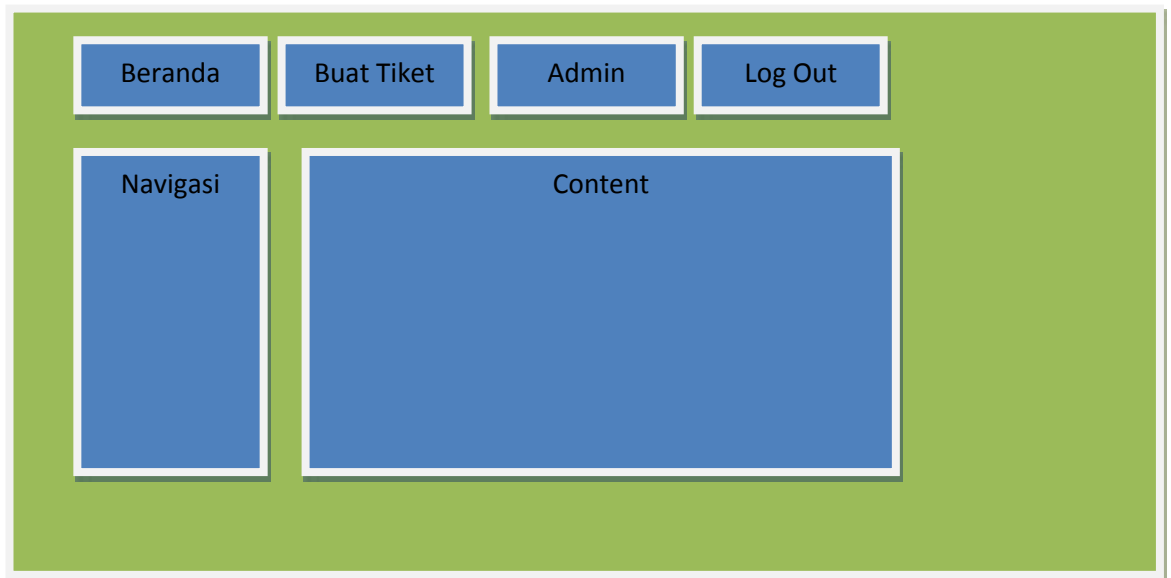
1. Halaman Log In

The image shows a login page with a green background. At the top center, the text "Halaman Log In" is displayed. Below this, there are two input fields: "Username" and "Password". Each input field is a white rectangle with a black border. Below the "Password" field is a black oval button with the word "Masuk" written in white text.

Gambar 3.10 Halaman Log In

Halaman log in ini sebagai awal masuk ke system helpdesk, user atau pengguna harus memasukkan username dan password yang valid agar dapat masuk ke dalamnya, apabila tidak maka akan muncul pemberitahuan di bawah button masuk, bahwa username dan password anda tidak valid, atau id anda sedang terkunci oleh admin, atau id anda tidak termasuk ke dalam database aplikasi ini.

2. Halaman Home



Gambar 3.11 Halaman Home

a. Beranda

Halaman ini berisikan segala aktifitas dan hal – hal atau perintah – perintah yang bisa kita lakukan dalam aplikasi helpdesk system ini, di bagian header terdapat button beranda yang link langsung ke tampilan awal home, yaitu tampilan yang pertama kali muncul setelah kita log in.

b. Buat Tiket

Ketika kita mengklik button ini maka akan langsung menampilkan formulir pengisian tiket, yang akan muncul pada bagian content.

c. Admin

Button admin ini akan langsung mengarahkan anda menuju navigasi admin atau segala sesuatu yang dapat dilakukan admin, yang akan di tampilkan pada bagian navigasi. Admin bisa melakukan semua yang bisa dilakukan oleh user, namun user tidak bisa melakukan apa – apa yang bisa dilakukan oleh admin.

d. Log Out

Log out akan langsung mengarahkan anda kembali kepada halaman log in, anda harus log in kembali, untuk bisa masuk ke halaman home

e. Navigasi

Navigasi berisikan berbagai macam perintah yang bisa dilakukan oleh user, navigasi ini terbagi menjadi dua, navigasi user, dan navigasi admin.

f. Content

Content akan menampilkan semua perintah – perintah yang dilakukan oleh user, sehingga tidak perlu proses yang lebih untuk mengarah ke link yang lain untuk melakukan suatu proses, kecuali log out, ketika user ingin log out maka akan langsung kembali ke halaman log in.

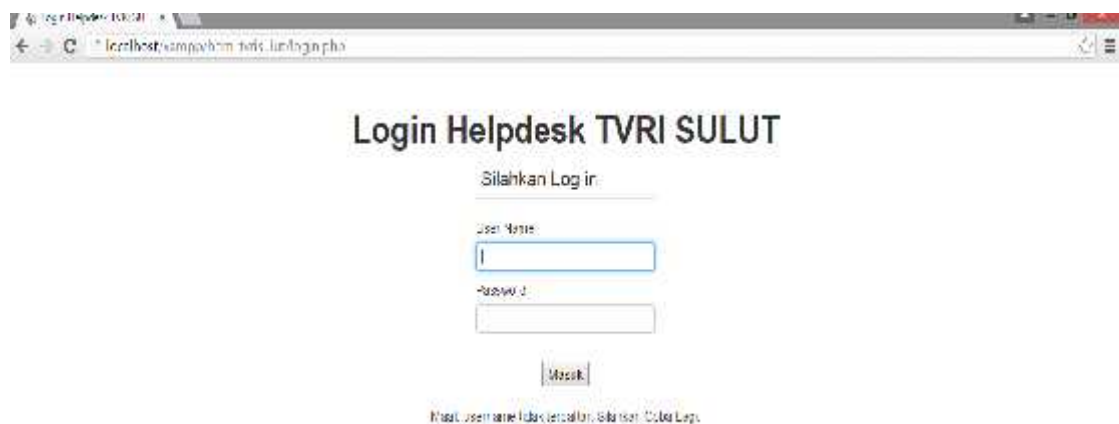
BAB IV PENGUJIAN DAN ANALISA

4.1 Tujuan Pengujian

Pengujian system ini memiliki tujuan untuk menguji kinerja serta helpdesk system. Dengan pengujian ini dapat diketahui apakah aplikasi yang telah dirancang dapat bekerja sesuai dengan yang diinginkan atau tidak. Cara pengujian ini dilakukan dalam pengujian hak akses user, tampilan grafik, skala prioritas, serta beberapa fitur lain yang terdapat pada aplikasi ini.

4.2 Pengujian Log in User

Tentunya sebelum menggunakan aplikasi ini pengguna harus log in terlebih dahulu, penulis menambahkan beberapa perintah PHP pada halaman log in, apabila username dan password yang dimasukkan valid, maka akan langsung menuju ke halaman home, namun apa bila tidak maka akan menampilkan beberapa pemberitahuan sebagaimana yang terjadi pada gambar berikut :



Gambar 4.1 Gagal Log In

Perhatikan pesan yang di tampilkan di bawah button Masuk, ini di karenakan username yang dimasukkan tidak terdaftar dalam database tabel user. Ada juga peringatan lain seeperti yang di bawah ini



Gambar 4.2 Invalid Username/Password

Ini menandakan bahwa username yang di masukkan ada dalam database, namun password yang di masukkan tidak valid dengan username yang di masukkan. Untuk menanggulangi masalah – masalah yang terjadi di atas user atau pengguna bisa langsung menghubungi admin untuk menanyakan apakah id pengguna tersebut sudah di buat atau belum, atau apabila sudah di buat namun lupa password nya juga bisa langsung menghubungi admin.

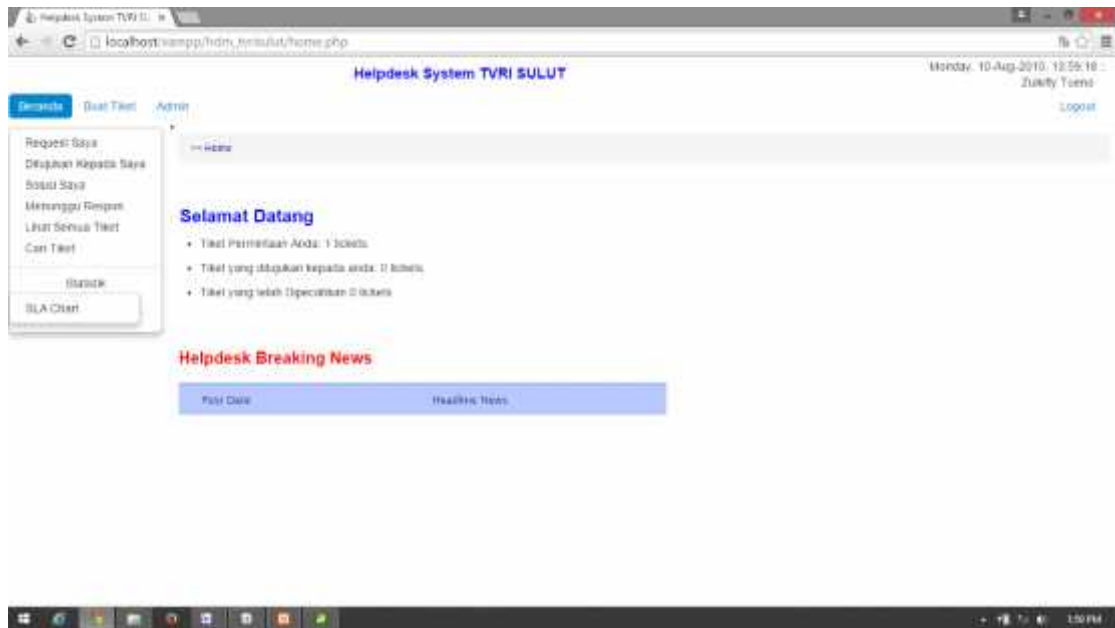
Berikut adalah tabel pengujian log in user

Username	Password	Masuk	Keterangan
Ya	Ya	ya	Berhasil Log In
Ya	Tidak	tidak	Username / Password Tidak Valid
Tidak	Tidak	Tidak	Username yang anda masukkan tidak terdaftar
Ya	Ya	tidak	Id anda sedang terkunci, silahkan hubungi administrator

Tabel 4.1 Pengujian Log In

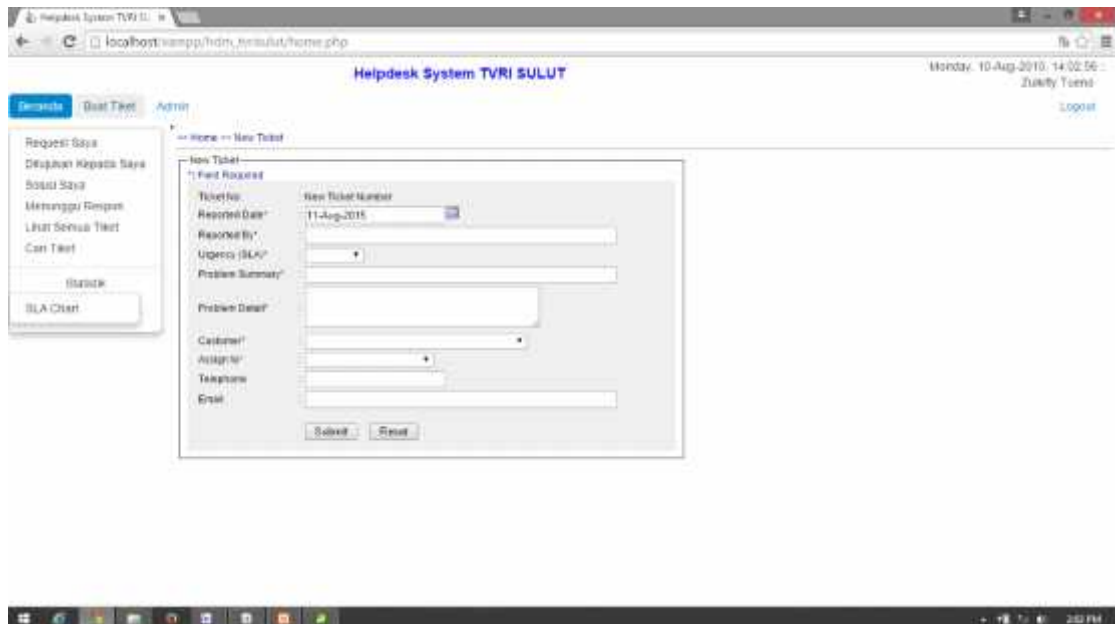
4.3 Pengujian Halaman Home

Setelah pengguna berhasil log in, maka akan langsung masuk ke halaman home, atau halaman utama. Berikut adalah tampilan halaman home setelah berhasil melakukan log in :



Gambar 4.3 Halaman Home

Selanjutnya kita akan mencoba membuat tiket baru dengan cara mengklik button yang ada pada bagian header, tepatnya berada di sebelah button Beranda. Berikut tampilan setelah button buat tiket di klik :



Gambar 4.4 Pembuatan tiket Home

Sebagaimana telah di jelaskan sebelumnya pengguna harus mengisi semua field yang terdapat pada halaman ini agar tiket bisa di buat, jika tidak maka akan muncul beberapa pemberitahuan tentang kesalahan pengisian yang terjadi. Selanjutnya kita akan melihat tiket yang berhasil di buat :

Ticket No: 46/SR/Aug/2015

Customer	: Seksi Teknik (Produksi dan Penyiaran)
Ticket No:	: 46/SR/Aug/2015
Reported Date	: 11-Aug-2015
Reported By	: Zulkifly
Urgency (SLA)	: Medium
Problem Summary	: Kena Virus
Problem Detail	: mati tota;
Telp	: 0808080
Email	: maa@gmail.com

Assign to	: Muhammad Aldy Amiri
Status	: Assigned
Resolution	:

Ticket Audit Trail

Updated On	Updated By	Description
11-Aug-2015 04:06:27	Zulkifly Tueno	Create New Ticket

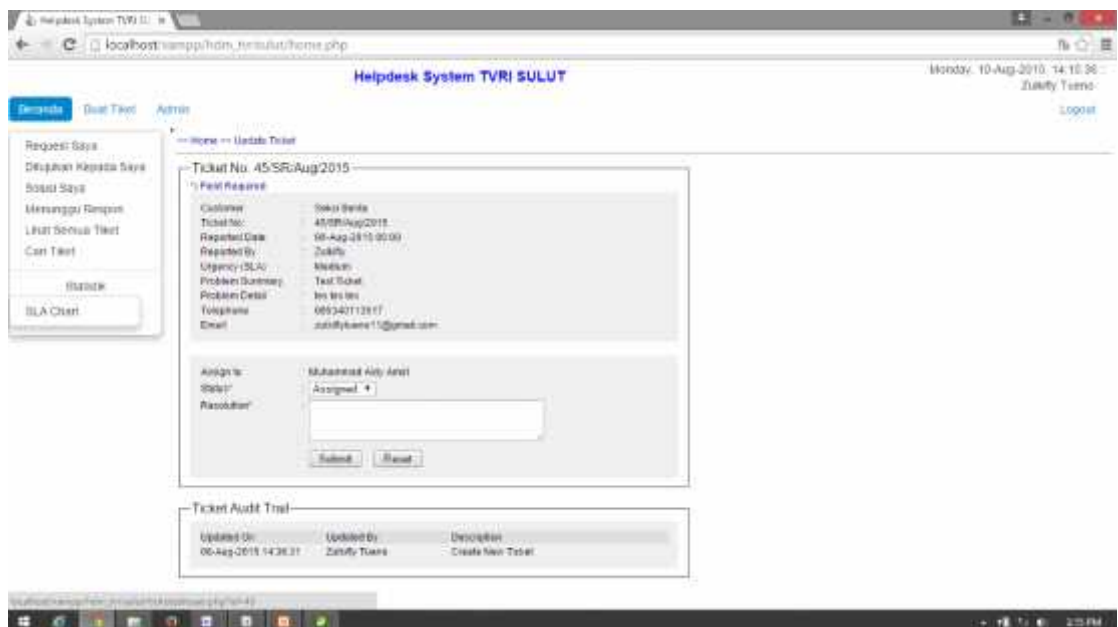
Gambar 4.5 Tampilan Tiket

Selanjutnya kitan akan menampilkan daftar request tiket yang telah kita buat, agar dapat mengetahui proses, dan sudah sampai mana permintaan kita di follow up, ini bisa dilakukan dengan menekan link yang ada pada navigasi yaitu request saya, berikut adalah tampilan setelah tombol itu di klik



Gambar 4.6 Tampilan Request

Apabila link pada nomor tiket di klik maka akan menampilkan informasi tiket sebagai berikut:



Gambar 4.7 Status Tiket

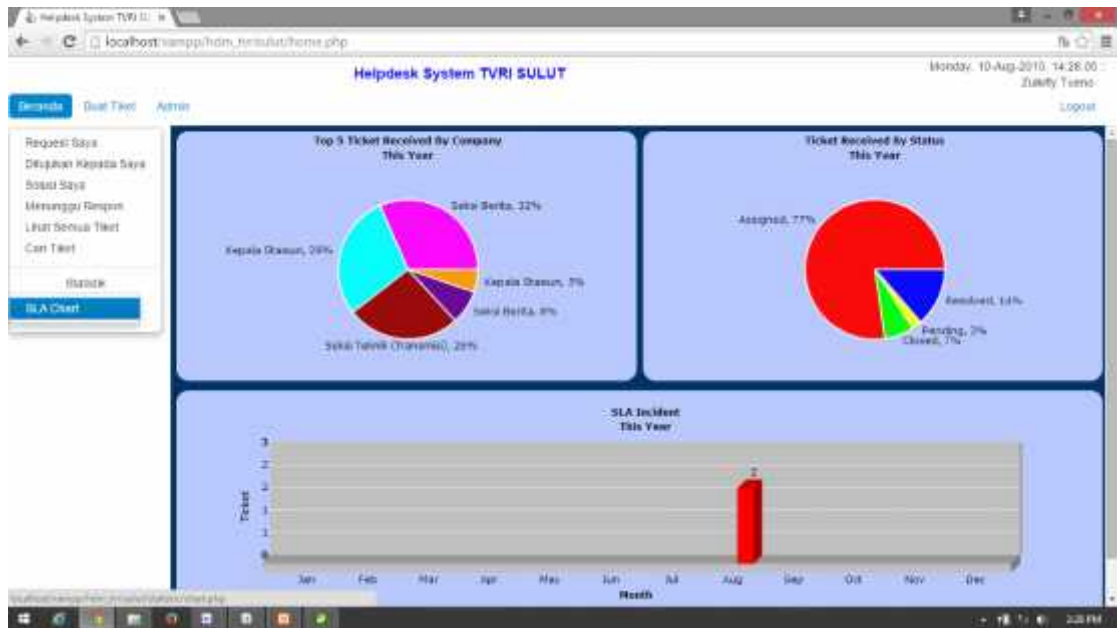
Halaman ini menampilkan status tiket, karna tiket ini kita yang buat maka kita bisa mengganti status tiket tersebut, apakah mau kita arahkan ke orang lain.

Selanjutnya juga berbagai fitur seperti melihat tiket yang di alamatkan kepada pengguna, solusi saya, menunggu respon, cari tiket, lihat semua tiket, fitur tersebut terdapat dalam frame navigasi pengguna tinggal mengklik sesuatu yang ingin di lihatnya, dan tentunya akan di tampilkan pada halaman content



Gambar 4.8 Frame Navigasi

Selanjutnya akan dilakukan pengujian untuk menampilkan grafik berdasarkan SLA atau skala prioritas yang telah di buat.



Gambar 4.9 Report Grafik

Selanjutnya akan di tampilkan halaman admin, atau navigasi apa saja yang bisa dilakukan oleh seorang admin, dengan mengklik tombol admin yang ada pada header.



Gambar 4.10 Navigasi Admin

Sebagaimana telah di jelaskan sebelumnya, admin mempunyai navigasi tersendiri, yang tidak bisa di akses oleh user biasa, namun segala perintah yang bisa dilakukan oleh user, pasti bisa dilakukan oleh admin.

yang terakhir adalah tombol logout, ketika tombol ini di tekan, maka akan langsung menghantarkan pengguna kembali ke halaman Log In, tombol Log Out terdapat pada bagian kanan atas.



Gambar 4.11 Log Out

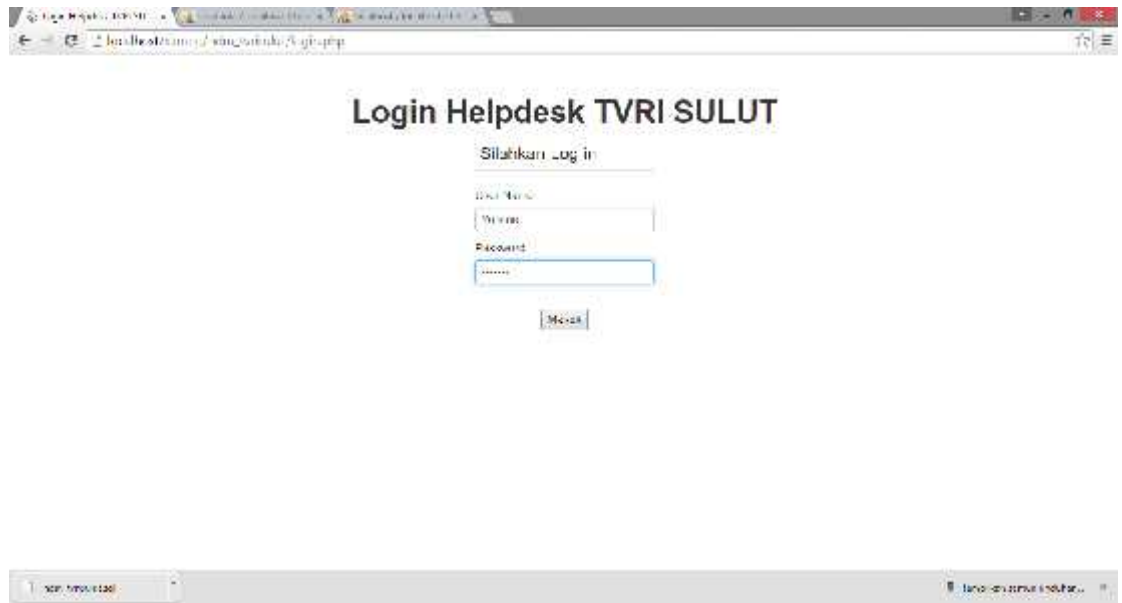
4.4 Pengujian Contoh Kasus

Penulis akan melakukan pengujian terhadap aplikasi dengan mengambil beberapa contoh kasus atau masalah yang terjadi dalam aktifitas keseharian di kantor TVRI SULUT. Beberapa diantaranya yaitu perbaikan beberapa PC, pengadaan barang, spesifikasi barang yang di perlukan, serta *budget* yang dibutuhkan untuk pengadaan barang tersebut. Dalam hal ini ada beberapa seksi dan sub bagian yang terlibat, diantaranya, Seksi Berita, Seksi Teknik (Produksi dan Penyiaran), Seksi Program dan Pengembangan, Sub Bagian Umum, dan Sub Bagian Keuangan. Untuk lebih jelasnya akan di uraikan pada penjelasan di bawah ini.

4.4.1 Perbaikan PC

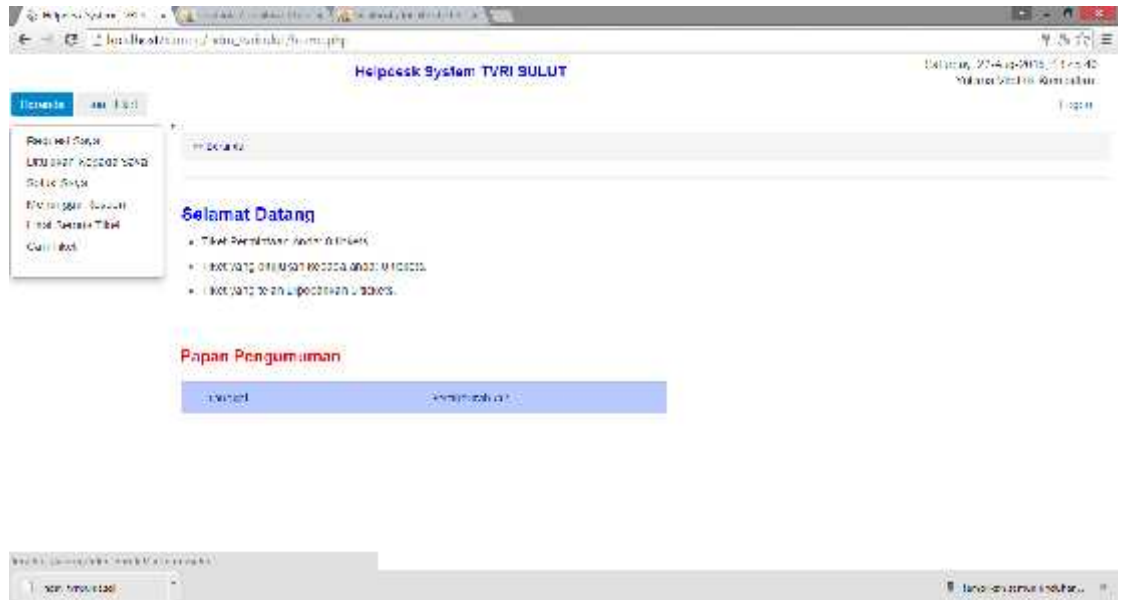
Dalam scenario kali ini Seksi Berita mengalami beberapa masalah dengan PC yang biasa digunakan, dan Seksi Berita melaporkan masalah tersebut kepada Seksi Teknik (Produksi dan Penyiaran) selaku divisi yang bertanggung jawab atas segala hal yang berhubungan dengan computer, termasuk perbaikan dan pemeliharaan, penulis akan menampilkan tutorial pengujian *step by step* untuk lebih jelasnya.

1. Salah seorang pegawai dalam seksi berita mengakses dan melakukan log in ke helpdesk system, dalam kasus ini yang melaporkan adalah yuliana maka log in menggunakan id user milikk yuliana yang telah dibuatkan oleh admin sebagaimana gambar berikut.



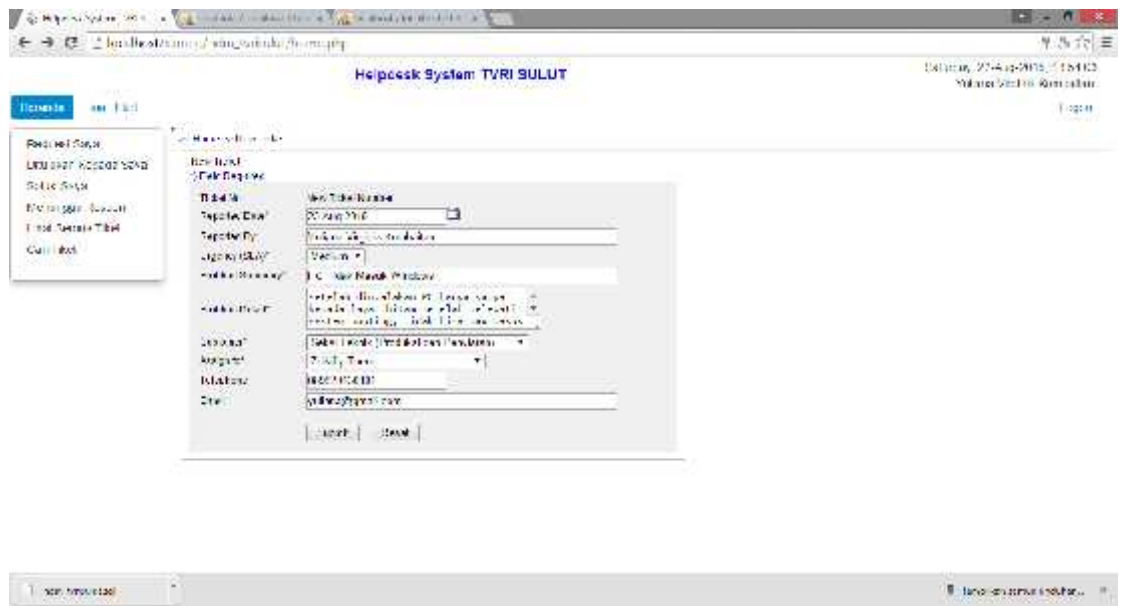
Gambar 4.12 Log in Yuliana

2. Selanjutnya yuliana langsung masuk ke beranda helpdesk dan meng-klik button Buat Tiket yang telah di sediakan, tepatnya berada di kiri atas.



Gambar 4.13 Klik Buat Tiket

3. Selanjutnya yuliana akan langsung dialihkan ke dalam form tiket, atau formulir pembuatan tiket dimana yuliana harus mengisi semua data – data yang di butuhkan, setelah itu menekan button submit yang telah disediakan dibagian paling bawah formulir.



Gambar 4.14 Formulir Tiket Yuliana

4. Setelah menekan button submit, akan langsung dipeerlihatkan semacam report bahwa data tentang tiket tersebut, selanjutnya yuliana tinggal menunggu respon balik dari orang yang telah di tunjuk untuk menanggapi tiket ini, seperti yang terlihat pada gambar 4.12, tiket ini diarahkan kepada zulkifylly tueno.

Pada halaman beranda zulkifly selaku admin dan anggota divisi teknik (produksi dan penyiaran) tampak tidak berbeda dengan halaman beranda yuliana, seperti yang terlihat pada gambar 4.13, tidak terlalu banyak perbedaan, namun jika diperhatikan dengan seksama maka ada dua hal yang berbeda yaitu pada halaman admin memiliki button admin, di bagian kiri atas, dan pada bagian kanan atas terdapat nama lengkap user yang sedang log in, atau yang sedang aktif pada saat itu.

Pada halaman ini juga ditunjukkan pada bagian tengah bahwa ada tiket yang di tunjukan kepada id tersebut.

Selamat Datang

- Tiket Permintaan Anda: 2 tickets.
- Tiket yang dilujukan kepada anda: 1 tickets.
- Tiket yang telah Dipcahkan 0 tickets.

Gambar 4.17 Pemberitahuan Tiket

Dengan melihat pemberitahuan tersebut, akan langsung di ketahui bahwa ada seseorang yang sedang melakukan permintaan kepada kita. Untuk melihat permintaan tersebut, yang bersangkutan telah disediakan link navigasi untuk melihat permintaan tersebut, tepat terletak di seblah kiri pemberitahuan yang di tunjukan oleh gambar 4.14.



Gambar 4.18 Link Tiket

6. Dengan meng-klik link “Ditunjukkan kepada saya” maka akan langsung ditampilkan daftar tentang tiket yang telah ditunjukkan kepada pengguna yang bersangkutan. Disini akan terlihat data tentang permintaan tersebut.

List of tickets that assigned to me
Number of tickets: 1

Ticket No.	Urgency	Customer	Reported Date	Reported By	Problem Summary	Status	Assigned To
4200200000000	Medium	teknik teknik (Pusat layanan konsultan)	2 Jul 2018	Juliana Virginia Kombaran	PC Tidak Masuk Pekerjaan	Assigned	Juliana Lucia

Showing 1 to 1 of 1 entries

Gambar 4.19 Daftar Permintaan

Pada daftar permintaan yang terlihat di atas, tepatnya pada gambar 4.15, terlihat nomor tiket berwarna biru dan digaris bawahi, itu merupakan link untuk menampilkan report tiket dengan lebih detail, maka untuk melihatnya, tinggal mengklik link tersebut.

7. Setelah di klik, akan langsung muncul report seperti yang yang di tampilkan report yuliana pada gambar 4.12, namun ada perbedaan yaitu pada bagian bawah form, yang disediakan agar yang bersangkutan bisa menanggapi permintaan tersebut.

Assign to* : Zulkify Tueno
Status* : Assigned
Resolution* :
Submit Reset

Gambar 4.20 Form Tanggapan

Pada bagian atas terdapat *assign to*, di maksudkan bahwa apabila orang yang di berikan permintaan tak mampu untuk menyelesaikan permintaan atau pertanyaan yang di ajukan, maka yang bersangkutan bisa mengarahkannya kepada orang lain yang di anggap mampu untuk menyelesaikan permintaan tersebut.

Assign to* : Zulkify Tueno
Status* :
Resolution* :
Zulkify Tueno
Yuliana Virginia Kombaitan
Subiantoro J.P. Masloman, S.Sos
Hanif
Bambang Suryo, SE, S.IP
Rahanra Onisimus, S.Sos
Muhammad Aldy Amiri
Zulkify Tueno

Gambar 4.21 Assign to

Selanjutnya ada field status, field ini menandakan status tiket apakah tiket bisa diselesaikan, atau pending, atau bahkan ingin di tujukan kepada user lain yang lebih berkompeten.

Assign to* : Zulkifly Tueno
 Status* : Assigned
 Resolution* :
 Assigned
 Resolved
 Pending
 Closed
 Reset

Gambar 4.22 Status

Dan terakhir ada field resolution, field ini digunakan untuk mengisi solusi, atau berbagai macam keterangan yang berkaitan dengan tiket permintaan yang dimkasudkan, dan di submit untuk di kirinkan kembali ke user yang melakukan permintaan.

8. Selanjutnya Yuliana bisa melihat atau mem-follow up status tiket yang dibuatnya tersebut, dengan meng-klik link navigasi “request saya”, lalu meng-klik link nomor tiket dan akan langsung di tampilkan report status tiket tersebut.

Assign to : Zulkifly Tueno
 Status* : Resolved
 Resolution* : saya akan segera memriksanya, kemungkinan permasalahan terjadi pada hardisk PC tersebut. saya telah
 Submit Reset

Gambar 4.23 Follow up Tiket

Setelah melihat tiket resolusi yang di berikan oleh Zulkifly maka Yuliana bisa memutuskan apakah permintaan atau keluhannya sudah terpenuhi atau belum, selanjutnya terdapat pilihan pada field status untuk menutup atau closing tiket.

4.4.2 Pengadaan Inventaris

Selanjutnya contoh kasus yang didapati setelah melakukan observasi langsung ke lapangan adalah tentang perihal pengadaan barang, seperti PC. Dalam hal ini yaitu tentang pengadaan barang harus melalui beberapa

prosedur agar bisa di adakannya barang inventaris tersebut. Pada kasus kali ini Seksi Program dan Pengembangan akan melakukan pengadaan beberapa PC terkait dengan adanya penerimaan pegawai baru, maka dibutuhkan pengadaan untuk menunjang segala aktifitas keseharian mereka.

Pertama – tama Seksi Program dan Pengembangan melakukan request tiket kepada Subbag Umum selaku bidang divisi yang bertanggung jawab mengenai pengadaan barang ini. Selanjutnya karna pengadaan barang kali ini berhubungan dengan computer maka, Subbag Umum melanjutkan tiket tersebut kepada Seksi Teknik (Penyiaran dan Produksi) selaku yang berkompeten pada bidang tersebut, untuk menanyakan bahwa spesifikasi apa saja yang dibutuhkan oleh Seksi Program dan Pengembangan dan berapa kisaran harga barang tersebut untuk dibuatkan estimasi dana. Konfirmasi tentang harga ini sangat penting, karna beda divisi beda pula ruang lingkup kerjanya, dan beda pula barang inventaris penunjang, contoh Subbag Umum yang notabennya hanya bergelut di bidang pengolahan data dokumen dan menggunakan software seperti Microsoft Office hanya memerlukan PC dengan spesifikasi menengah kebawah dan tentunya dengan harga yang lebih murah pula, berda dengan seksi teknik yang bergelut dengan bidang editing video misalnya, tentu akan memerlukan spesifikasi PC yang tinggi dengan harga yang tinggi pula. Penulis akan coba menerapkannya dalam helpdesk.

Setalah melakukan konfirmasi perihal harga ataaau budget yang di butuhkan, maka selanjutnya akan langsung di arahkan kepada Subbag Keuangan untuk pencairan dana, dan di berikan kepada Subbag Umum untuk pengadaan, dan kembali lagi keapda Seksi Teknik untuk penginstalan, apabila barang sudah tersedia.

Untuk penerapannya kepada helpdesk, akan di pandu dengan tutorial step by step sebagai berikut:

1. Pada kasus kali ini yang akan menyampaikan permintaan berasal dari salah satu pegawai yang berada pada Seksi Program dan Pengembangan, Jane Christine Aror dengan username Jane, dan akan Log in ke Helpdesk system.

Login Helpdesk TVRI SULUT

Silahkan Log in

User Name :

Jane

Password :

....

Masuk

Gambar 4.24 Log In Jane

2. Kemudian Jane masuk ke halaman buat tiket dengan cara klik Buat Tiket yang ada pada header halaman untuk menyampaikan permintaan.



Gambar 4.25 Beranda Jane

3. Jane mengisi seluruh field pada form buat tiket.

Gambar 4.26 Form Tiket Jane

4. Setelah di submit akan langsung terlihat report tiket tersebut.

Gambar 4.27 Report Tiket Jane

5. Selanjutnya sesuai dengan report tiket pada gambar 2.22 bahwa jane mengarahkan atau assign tiket tersebut kepada Sofyan Napu selaku salah satu pegawai yang ada pada Sub. Bag Umum. Tutorial berikut akan log ini sebagai Sofyan untuk melihat tiket yang di kirimkan oleh Jane.

Login Helpdesk TVRI SULUT

Silahkan Log in

User Name :

Password :

Gambar 4.28 Log In Sofyan

Dan selanjutnya proses akan seperti yang sudah di jelaskan pada sub bab sebelumnya.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan perencanaan dan membuat suatu sistem pengujian beserta analisisnya, maka dapat diambil beberapa kesimpulan dari kinerja keseluruhan helpdesk System :

1. Pengguna dapat menyampaikan keluhan, memberikan solusi untuk keluhan pengguna atau divisi lain, dan sekaligus mendokumentasikan keluhan dan pertanyaan untuk menjadi laporan atau report bagi atasan dalam mengevaluasi berbagai kendala dalam pekerjaan kesharian di kantor serta menjadi catatan apabila keluhan atau pertanyaan yang sama muncul kembali.
2. Aplikasi dapat berfungsi dengan baik untuk log in multiuser, pembuatan tiket, pembuatan skala prioritas tiket, dan report grafik menampilkan grafik
3. Aplikasi dapat berfungsi dengan baik dalam menampung dan mendokumentasikan berbagai keluhan dari pengguna.
4. Script dan database yang dibuat dapat bersinergi dan berjalan dengan lancar tanpa menampilkan error dan notice yang lainnya.

5.2 Saran

Aplikasi helpdesk system seperti sangat bergantung pada keadaan lokasi tiap – tiap instansi atau perusahaan, dan pasti suatu hari perusahaan itu akan berkembang, maka akan berkembang pula system yang digunakan dan situasi kondisi di perusahaan tersebut, oleh karena itu, aplikasi ini tidaklah bersifat tetap, jika perusahaan atau instansi tempat aplikasi ini telah berkembang, maka aplikasi pun harus berkembang agar tetap sesuai dengan kinerja para pegawainya dan dapat terus digunakan. Besar harapan dari penulis agar aplikasi helpdesk system ini dapat dikembangkan menjadi lebih sempurna lagi dan bisa menjadi sebuah produk yang memiliki nilai jual dan dapat bersaing di era perkembangan teknologi sekarang ini khususnya daerah Manado dan sekitarnya.

DAFTAR PUSTAKA

BUKU

Alatas Husein. (2013). **Responsive Web Design dengan PHP dan BOOTSTRAP**. Lokomedia. . ISBN 978-602-14306-1-3

Jubilee Enterprise. (2014). **HTML 5 Manual Book**. PT Elex Media Komputindo. . ISBN: 978-602-02-5357-2

Santoso Hendra. (2014). **Membuat Helpdesk System Berbasis OOP dan PDO**. Lokomedia. . ISBN: 987-602-14306-7-5

URL

Helpdesk, <http://www.help-desk-world.com/help-desk.htm> Diakses pada tanggal 3 Mei 2015

Web PC, <http://www.webpcwebipedia.com> Diakses pada tanggal 10 July 2015

Apa itu Helpdesk, <http://tito.pandubrahmanto.com/computer/apa-itu-help-desk/> Diakses pada tanggal 3 Mei 2015

Online Web Tutorial, <http://www.w3schools.com/> Diakses pada tanggal 22 Maret 2015

Fungsi PHP, <http://www.duniaikom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/> Diakses pada tanggal 14 Juli 2015

JQuery Official Website, <http://jquery.com/> Diakses pada tanggal 28 Juli 2015

Get Bootsrap, <http://getbootstrap.com/2.3.2/> Diakses pada tanggal 1 Agustus 2015

Lampiran

1. General.php

```
<?php
class General
{
    public function logged_in () {
        return(isset($_SESSION['loginid'])) ? true : false;
    }

    public function logged_in_protect() {
        if ($this->logged_in() === true) {
            header('Location: home.php');
            exit();
        }
    }

    public function logged_out_protect() {
        if ($this->logged_in() === false) {
            header('Location: index.php');
            exit();
        }
    }
}
?>
```

2. HDNews.php

```
<?php
class HDNews{
    private $db;

    public function __construct($database)
    {
        $this->db = $database;
    }

    public function
add_news($newsdate,$title,$detail,$createdby,$createdon,$expired)
    {
        $query = $this->db->prepare("INSERT INTO `news`
(`newsdate`,`title`,`detail`,`createdby`,`createdon`,`expired`)
VALUES (?, ?, ?, ?, ?, ?)");

        $query->bindValue(1, $newsdate);
        $query->bindValue(2, $title);
        $query->bindValue(3, $detail);
        $query->bindValue(4, $createdby);
        $query->bindValue(5, $createdon);
        $query->bindValue(6, $expired);

        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
    }

    public function
update_news($id,$newsdate,$title,$detail,$createdby,$createdon,$expir
ed)
    {
        $query = $this->db->prepare("UPDATE `news` SET
`newsdate` = ? , `title` = ? , `detail` = ? , `createdby` = ? ,
`createdon` = ? , `expired` = ? WHERE `id` = ?");
```

```
$query->bindValue(1, $newsdate);
$query->bindValue(2, $title);
$query->bindValue(3, $detail);
$query->bindValue(4, $createdby);
$query->bindValue(5, $createdon);
$query->bindValue(6, $expired);
$query->bindValue(7, $id);
try{
    $query->execute();
}
catch(PDOException $e){
    die($e->getMessage());
}
}

public function delete($id){
    $sql="DELETE FROM `news` WHERE `id` = ?";
    $query = $this->db->prepare($sql);
    $query->bindValue(1, $id);
    try{
        $query->execute();
    }
    catch(PDOException $e){
        die($e->getMessage());
    }
}
```

```

        public function news_data($id)
        {
            $query = $this->db->prepare("SELECT * FROM `news` WHERE
`id` = ?");
            $query->bindValue(1, $id);
            try{
                $query->execute();
                return $query->fetch();
            } catch(PDOException $e){
                die($e->getMessage());
            }
        }

        public function get_news()
        {
            $query = $this->db->prepare("SELECT * FROM `news` ORDER
BY `newsdate` ASC");
            try{
                $query->execute();
            }catch(PDOException $e){
                die($e->getMessage());
            }
            return $query->fetchAll();
        }

        public function get_headline_news()
        {
            $query = $this->db->prepare("SELECT * FROM `news` WHERE
UNIX_TIMESTAMP( curdate( ) ) < `expired` ORDER BY `newsdate` ASC");
            try{
                $query->execute();
            }catch(PDOException $e){

```

```
        die($e->getMessage());
    }
    return $query->fetchAll();
}
}
```


3. SLA.php

```
<?php
class SLA{
    private $db;

    public function __construct($database)
    {    $this->db = $database; }

    public function sla_exists($slaid)
    {    $query = $this->db->prepare("SELECT COUNT(`slaid`) FROM
`sla` WHERE `slaid`= ?");

        $query->bindValue(1, $slaid);

        try
        {    $query->execute();

            $rows = $query->fetchColumn();

            if($rows == 1){

                return true;

            }else{

                return false;

            }

        } catch (PDOException $e){

            die($e->getMessage());

        }

    }

    public function
add_sla($slaid,$namasla,$responsetime,$resolutiontime,$slawarning)

    {    $querystring = "INSERT INTO `sla`
(`slaid`,`namasla`,`responsetime`,`resolutiontime`,`slawarning`)
VALUES (?, ?, ?, ?, ?)";

        $query    = $this->db->prepare($querystring);
```

```

$query->bindValue(1, $slaid);
$query->bindValue(2, $namasla);
$query->bindValue(3, $responsetime);
$query->bindValue(4, $resolutiontime);
$query->bindValue(5, $slawarning);
try{
    $query->execute();
}catch(PDOException $e){
    die($e->getMessage());
}
}

public function
update_sla($slaid,$namasla,$responsetime,$resolutiontime,$slawarning
)
{
    $querystring = "UPDATE `sla` SET `namasla` = ? ,
`responsetime` = ? , `resolutiontime` = ? , `slawarning` = ? WHERE
`slaid` = ?";

    $query      = $this->db->prepare($querystring);
    $query->bindValue(1, $namasla);
    $query->bindValue(2, $responsetime);
    $query->bindValue(3, $resolutiontime);
    $query->bindValue(4, $slawarning);
    $query->bindValue(5, $slaid);
    try{
        $query->execute();
    }
    catch(PDOException $e){
        die($e->getMessage());
    }
}

```

```

    }

}

public function delete($id){
    $sql="DELETE FROM `sla` WHERE `slaid` = ?";
    $query = $this->db->prepare($sql);
    $query->bindValue(1, $id);
    try{
        $query->execute();
    }
    catch(PDOException $e){
        die($e->getMessage());
    }
}

public function sla_data($slaid)
{
    $query = $this->db->prepare("SELECT * FROM `sla` WHERE
`sleid` = ?");
    $query->bindValue(1, $slaid);
    try{
        $query->execute();
        return $query->fetch();
    } catch(PDOException $e){
        die($e->getMessage());
    }
}

public function get_sla()

```

```
        {      $query = $this->db->prepare("SELECT * FROM `sla` ORDER
BY `slaid` ASC");
            try{
                $query->execute();
            }catch(PDOException $e){
                die($e->getMessage());
            }
            return $query->fetchAll();
        }
    }
```

4. Tiket.php

```
<?php
date_default_timezone_set('Asia/Jakarta');
class Tickets{
    private $db;

    public function __construct($database)
    {    $this->db = $database; }

    public function ticket_exists($projectid)
    {    $query = $this->db->prepare("SELECT COUNT(`id`) FROM
`tickets` WHERE `ticketnumber`= ?");

        $query->bindValue(1, $projectid);

        try
        {    $query->execute();

            $rows = $query->fetchColumn();

            if($rows == 1){

                return true;

            }else{

                return false;

            }

        } catch (PDOException $e){

            die($e->getMessage());

        }

    }

    public function
add_ticket($ticketnumber,$sla,$idcustomer,$reporteddate,$reportedby,
$telp,$email,$problemsummary,$problemdetail,$ticketstatus,$assignee,
$documentedby)
```

```

        {
            $current = time();

            $querystring = "INSERT INTO `tickets`
(`ticketnumber`,`sla`,`idcustomer`,`reporteddate`,`reportedby`,`
telp`,`email`,`
problemsummary`,`problemdetail`,`ticketstatus`,`assignee`,`document
edby`,`documenteddate`) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?)";

            $query = $this->db->prepare($querystring);
            $query->bindValue(1, $ticketnumber);
            $query->bindValue(2, $sla);
            $query->bindValue(3, $idcustomer);
            $query->bindValue(4, $reporteddate);
            $query->bindValue(5, $reportedby);
            $query->bindValue(6, $telp);
            $query->bindValue(7, $email);
            $query->bindValue(8, $problemsummary);
            $query->bindValue(9, $problemdetail);
            $query->bindValue(10, $ticketstatus);
            $query->bindValue(11, $assignee);
            $query->bindValue(12, $documentedby);
            $query->bindValue(13, $current);
            try{
                $query->execute();
            }catch(PDOException $e){
                die($e->getMessage());
            }
        }

        public function
update_ticket($id,$sla,$reporteddate,$reportedby,$telp,$email,$probl

```

```
emsummary,$problemdetail,$ticketstatus,$assignee,$assigneddate,$pendingby, $pendingdate,
$resolution,$resolvedby,$resolveddate,$closedby,$closeddate)
```

```
{    $querystring = "UPDATE `tickets` SET `sla` = ? ,
`reporteddate` = ? , `reportedby` = ? , `telp` = ? , `email` = ? ,
`problemsummary` = ? , `problemdetail` = ? , `ticketstatus` = ? ,
`assignee` = ? , `assigneddate` = ? , `pendingby` = ? , `pendingdate` =
? , `resolution` = ? , `resolvedby` = ? , `resolveddate` = ? , `closedby`
= ? , `closeddate` = ? WHERE `id` = ?";
```

```
    $query      = $this->db->prepare($querystring);
```

```
    $query->bindValue(1, $sla);
```

```
    $query->bindValue(2, $reporteddate);
```

```
    $query->bindValue(3, $reportedby);
```

```
    $query->bindValue(4, $telp);
```

```
    $query->bindValue(5, $email);
```

```
    $query->bindValue(6, $problemsummary);
```

```
    $query->bindValue(7, $problemdetail);
```

```
    $query->bindValue(8, $ticketstatus);
```

```
    $query->bindValue(9, $assignee);
```

```
    $query->bindValue(10, $assigneddate);
```

```
    $query->bindValue(11, $pendingby);
```

```
    $query->bindValue(12, $pendingdate);
```

```
    $query->bindValue(13, $resolution);
```

```
    $query->bindValue(14, $resolvedby);
```

```
    $query->bindValue(15, $resolveddate);
```

```
    $query->bindValue(16, $closedby);
```

```
    $query->bindValue(17, $closeddate);
```

```
    $query->bindValue(18, $id);
```

```
    try{
```

```

        $query->execute();
    }
    catch(PDOException $e){
        die($e->getMessage());
    }
}

public function delete($id){
    $sql="DELETE FROM `tickets` WHERE `id` = ?";
    $query = $this->db->prepare($sql);
    $query->bindValue(1, $id);
    try{
        $query->execute();
    }
    catch(PDOException $e){
        die($e->getMessage());
    }
}

public function ticket_data($id)
{
    $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `id`= ?");
    $query->bindValue(1, $id);
    try{
        $query->execute();
        return $query->fetch();
    } catch(PDOException $e){
        die($e->getMessage());
    }
}

```



```

        }
    }
    public function get_tickets()
    {
        $query = $this->db->prepare("SELECT * FROM `tickets`
ORDER BY `ticketnumber` DESC");
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }
    public function get_opened_tickets()
    {
        $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `ticketstatus` <> 'Closed' ORDER BY `ticketnumber` DESC");
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }
    public function get_tickets_by_requester($userid)
    {
        $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `documentedby` = ? ORDER BY `ticketnumber` DESC");
        $query->bindValue(1, $userid);
        try{

```

```

        $query->execute();
    }catch(PDOException $e){
        die($e->getMessage());
    }
    return $query->fetchAll();
}

public function get_tickets_by_assignee($userid)
{
    $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `assignee`= ? ORDER BY `ticketnumber` DESC");
    $query->bindValue(1, $userid);
    try{
        $query->execute();
    }catch(PDOException $e){
        die($e->getMessage());
    }
    return $query->fetchAll();
}

public function get_tickets_by_resolver($username)
{
    $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `resolvedby`= ? ORDER BY `ticketnumber` DESC");
    $query->bindValue(1, $username);
    try{
        $query->execute();
    }catch(PDOException $e){
        die($e->getMessage());
    }
    return $query->fetchAll();
}

```

```

    }

    public function get_tickets_by_resolver_not_closed($username)

    {
        $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `resolvedby`=? and `ticketstatus` <> ? ORDER BY `ticketnumber`
DESC");

        $query->bindValue(1, $username);
        $query->bindValue(2, 'Closed');
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }

    public function get_tickets_by_status($ticketstatus)

    {
        $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `ticketstatus`=? ORDER BY `ticketnumber` DESC");
        $query->bindValue(1, $ticketstatus);
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }

    public function search_closed_ticket($fromperiod, $toperiod)

```

```

        {
            $query = $this->db->prepare("SELECT * FROM `tickets`
WHERE `documenteddate` >= ? AND `documenteddate` <= ? AND
`ticketstatus` = 'Closed' ORDER BY `documenteddate` DESC");

            $query->bindValue(1, $fromperiod);
            $query->bindValue(2, $toperiod);

            try{

                $query->execute();

            }catch(PDOException $e){

                die($e->getMessage());

            }

            return $query->fetchAll();

        }

        public function count_tickets_by_customer()

        {
            $query = $this->db->prepare("SELECT `idcustomer`,
count(*) as `total` FROM `tickets` GROUP BY `idcustomer` ORDER BY
total DESC LIMIT 5");

            try{

                $query->execute();

            }catch(PDOException $e){

                die($e->getMessage());

            }

            return $query->fetchAll();

        }

        public function count_tickets_by_status()

        {
            $query = $this->db->prepare("SELECT ticketstatus,
count(*) as total FROM `tickets` GROUP BY ticketstatus");

            try{

                $query->execute();

```

```

        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }

    public function count_resolved_tickets_by_month()
    {
        $sql="SELECT Month(FROM_UNIXTIME(`documenteddate`)) as
        Bulan, Count(*) as Total FROM `tickets` WHERE
        (`ticketstatus`='Resolved' OR `ticketstatus`='Closed') AND
        FROM_UNIXTIME(`documenteddate`) >= CURDATE() - INTERVAL 1 YEAR GROUP
        BY Month(FROM_UNIXTIME(`documenteddate`))";

        $query = $this->db->prepare($sql);
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }

    public function count_inprogress_tickets_by_month()
    {
        $sql="SELECT Month(FROM_UNIXTIME(`documenteddate`)) as
        Bulan, Count(*) as Total FROM `tickets` WHERE
        (`ticketstatus`='Assigned' OR `ticketstatus`='Pending') AND
        FROM_UNIXTIME(`documenteddate`) >= CURDATE() - INTERVAL 1 YEAR GROUP
        BY Month(FROM_UNIXTIME(`documenteddate`))";

        $query = $this->db->prepare($sql);
        try{
            $query->execute();
        }catch(PDOException $e){

```

```

        die($e->getMessage());
    }
    return $query->fetchAll();
}

public function get_last_ticket()
{
    $query = $this->db->prepare("SELECT * FROM `tickets`
ORDER BY id DESC LIMIT 1");
    try{
        $query->execute();
        return $query->fetch();
    } catch(PDOException $e){
        die($e->getMessage());
    }
}/*

public function
notify_assignee($id,$ticketnumber,$email_assignee)
{
    if (substr(PHP_OS, 0, 7) == "Windows"){
        $cmd =
"D:\mowes_portable\www\helpdesk\batch\sendemail.bat";
        $WshShell = new COM("WScript.Shell");
        $oExec = $WshShell->Run("cmd /C $cmd", 0, false);
        return $oExec == 0 ? true : false;
    }
    else {
        $cmd = "php /batch/sendemail.bat";
        exec($cmd . " > /dev/null &");
    }
}

```

```

    }*/

    public function
log_tickets($id,$sla,$reporteddate,$reportedby,$telp,$email,$problem
summary,$problemdetail,$ticketstatus,$assignee,$assigneddate,$pendin
gby, $pendingdate,
$resolution,$resolvedby,$resolveddate,$closedby,$closeddate,$changes
,$changeby)

    {      $changedate = time();

            $querystring = "INSERT INTO `log_tickets`
(`id`,`sla`,`reporteddate`,`reportedby`,`telp`,`email`,`
`problemsummary`,`problemdetail`,`ticketstatus`,`assignee`,`assigned
date`,`pendingby`,`pendingdate`,`resolution`,`resolvedby`,`resolvedd
ate`,`closedby`,`closeddate`,`changes`,`changeby`,`changedate`)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?)" ;

            $query      = $this->db->prepare($querystring);

            $query->bindValue(1, $id);

            $query->bindValue(2, $sla);

            $query->bindValue(3, $reporteddate);

            $query->bindValue(4, $reportedby);

            $query->bindValue(5, $telp);

            $query->bindValue(6, $email);

            $query->bindValue(7, $problemsummary);

            $query->bindValue(8, $problemdetail);

            $query->bindValue(9, $ticketstatus);

            $query->bindValue(10, $assignee);

            $query->bindValue(11, $assigneddate);

            $query->bindValue(12, $pendingby);

            $query->bindValue(13, $pendingdate);

            $query->bindValue(14, $resolution);

            $query->bindValue(15, $resolvedby);

```

```

        $query->bindValue(16, $resolveddate);
        $query->bindValue(17, $closedby);
        $query->bindValue(18, $closeddate);
        $query->bindValue(19, $changes);
        $query->bindValue(20, $changeby);
        $query->bindValue(21, $changedate);
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
    }

    public function get_audit_trail($id)
    {
        $query = $this->db->prepare("SELECT * FROM `log_tickets`
WHERE `id`= ? ORDER BY `changedate` DESC");
        $query->bindValue(1, $id);
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }
}

```


5. User.php

```
<?php
class Users{
    private $db;

    public function __construct($database)
    {    $this->db = $database; }

    public function user_exists($username)
    {    $query = $this->db->prepare("SELECT COUNT(`id`) FROM
`users` WHERE `username`= ?");

        $query->bindValue(1, $username);

        try
        {    $query->execute();

            $rows = $query->fetchColumn();

            if($rows == 1){

                return true;

            }else{

                return false;

            }

        } catch (PDOException $e){

            die($e->getMessage());

        }

    }

    public function email_exists($email)

    {    $query = $this->db->prepare("SELECT COUNT(`id`) FROM
`users` WHERE `email`= ?");

        $query->bindValue(1, $email);
```

```

try{
    $query->execute();
    $rows = $query->fetchColumn();
    if($rows == 1){
        return true;
    }else{
        return false;
    }
} catch (PDOException $e){
    die($e->getMessage());
}
}

```

```

public function
register($idcustomer,$username,$password,$email,$fullname,$Telp,$level,$locked)

```

```

{
    $time          = time();
    $ip            = $_SERVER['REMOTE_ADDR'];
    $email_code    = sha1($username + microtime());
    $password      = sha1($password);
    $query         = $this->db->prepare("INSERT INTO `users`
(`idcustomer`,`username`,`password`,`level`,`fullname`,`email`,`Telp`,`ip`,`time`,`confirmed`) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
    $query->bindValue(1, $idcustomer);
    $query->bindValue(2, $username);
    $query->bindValue(3, $password);
    $query->bindValue(4, $level);

```

```

        $query->bindValue(5, $fullname);
        $query->bindValue(6, $email);
        $query->bindValue(7, $Telp);
        $query->bindValue(8, $ip);
        $query->bindValue(9, $time);

        $query->bindValue(10, 1);
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
    }

    public function
update($id,$idcustomer,$username,$password,$email,$fullname,$Telp,$l
evel,$locked)
    {
        $time            = time();
        $ip               = $_SERVER['REMOTE_ADDR'];
        $email_code      = sha1($username + microtime());
        $password         = sha1($password);

        $query           = $this->db->prepare("UPDATE `users` SET
`idcustomer` = ? , `password` = ? , `level` = ? , `fullname` = ? ,
`email` = ? , `Telp` = ? , `ip` = ? , `time` = ? , `confirmed` = ?
WHERE `id` = ?");

        $query->bindValue(1, $idcustomer);
        $query->bindValue(2, $password);
        $query->bindValue(3, $level);
        $query->bindValue(4, $fullname);

```

```

$query->bindValue(5, $email);
$query->bindValue(6, $Telp);
$query->bindValue(7, $ip);
$query->bindValue(8, $time);

$query->bindValue(9, $locked);
$query->bindValue(10, $id);
try{
    $query->execute();
}
catch(PDOException $e){
    die($e->getMessage());
}
}

public function changepwd($id,$password)
{
    $password = sha1($password);
    $query = $this->db->prepare("UPDATE `users` SET
`password` = ? WHERE `id` = ?");
    $query->bindValue(1, $password);
    $query->bindValue(2, $id);
    try{
        $query->execute();
    }
    catch(PDOException $e){
        die($e->getMessage());
    }
}

```

```

}

public function delete($id){
    $sql="DELETE FROM `users` WHERE `id` = ?";
    $query = $this->db->prepare($sql);
    $query->bindValue(1, $id);
    try{
        $query->execute();
    }
    catch(PDOException $e){
        die($e->getMessage());
    }
}

public function email_confirmed($username)
{
    $query = $this->db->prepare("SELECT COUNT(`id`) FROM
`users` WHERE `username`= ? AND `confirmed` = ?");
    $query->bindValue(1, $username);
    $query->bindValue(2, 1);
    try{
        $query->execute();
        $rows = $query->fetchColumn();
        if($rows == 1){
            return true;
        }else{
            return false;
        }
    }
}

```

```

        } catch(PDOException $e){
            die($e->getMessage());
        }
    }

    public function login($username, $password)
    {
        $query = $this->db->prepare("SELECT `password`, `id`
FROM `users` WHERE `username` = ?");
        $query->bindValue(1, $username);
        try{
            $query->execute();
            $data = $query->fetch();
            $stored_password = $data['password'];
            $id = $data['id'];
            if($stored_password === sha1($password)){
                return $id;
            }else{
                return false;
            }
        }catch(PDOException $e){
            die($e->getMessage());
        }
    }

    public function userdata($id)
    {
        $query = $this->db->prepare("SELECT * FROM `users` WHERE
`id` = ?");
        $query->bindValue(1, $id);
    }

```

```

        try{
            $query->execute();
            return $query->fetch();
        } catch(PDOException $e){
            die($e->getMessage());
        }
    }

    public function get_user_by_id($id)
    {
        $query = $this->db->prepare("SELECT * FROM `users` WHERE
`id` = ?");
        $query->bindValue(1, $id);
        try{
            $query->execute();
            return $query->fetch();
        } catch(PDOException $e){
            die($e->getMessage());
        }
    }

    public function get_user_by_level($level)
    {
        $query = $this->db->prepare("SELECT * FROM `users` WHERE
`level` = ?");
        $query->bindValue(1, $level);
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
    }

```

```

        return $query->fetchAll();
    }
    public function get_users()
    {
        $query = $this->db->prepare("SELECT * FROM `users` ORDER
BY `time` DESC");
        try{
            $query->execute();
        }catch(PDOException $e){
            die($e->getMessage());
        }
        return $query->fetchAll();
    }
    public function log_users($iduser,$log)
    {
        $time          = time();
        $ip            = $_SERVER['REMOTE_ADDR'];
        $browser       = $_SERVER['HTTP_USER_AGENT'];
        $query         = $this->db->prepare("INSERT INTO
`log_users` (`iduser`,`time`,`ip`,`browser`,`log`) VALUES (?, ?, ?,
?, ?)");
        $query->bindValue(1, $iduser);
        $query->bindValue(2, $time);
        $query->bindValue(3, $ip);
        $query->bindValue(4, $browser);
        $query->bindValue(5, $log);
        try{
            $query->execute();
        }catch(PDOException $e){

```



```
        die($e->getMessage());
    }
}
public function get_users_log()
{
    $query = $this->db->prepare("SELECT * FROM `log_users`
ORDER BY `time` DESC");
    try{
        $query->execute();
    }catch(PDOException $e){
        die($e->getMessage());
    }
    return $query->fetchAll();
}
}
```

6. Barcode Dropbox yang berisi file lengkap helpdesk Internal

