



ITS
Institut
Teknologi
Sepuluh Nopember

330 84/14/08



RSSI
cos. 37

Ans

1-1

2008

TUGAS AKHIR - CF 1380

**IMPLEMENTASI TEKNOLOGI
JAVASCRIPT LIBRARY EXT DAN JAVA
SERVER PAGES PADA APLIKASI WEB
CUSTOMER SUPPORT DENGAN SISTEM
MULTIPLE TOUCHPOINT**

RONNY ANSIS
NRP 5204 100 053

Pembimbing
Mudjahidin, S.T, M.T
Faizal Johan Atletiko, S.Kom

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2008

PERPUSTAKAAN ITS	
Tgl. Terima	8 - 8 - 2008
Terima Dari	H
No. Agenda Prp.	731539

FA2



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - CF 1380

**IMPLEMENTATION OF EXT JAVASCRIPT
LIBRARY TECHNOLOGY AND JAVA
SERVER PAGES ON WEB CUSTOMER
SUPPORT APPLICATION WITH
MULTIPLE TOUCHPOINT SYSTEM**

RONNY ANSIS
NRP 5204 100 053

Supervisor
Mudjahidin, S.T, M.T
Faizal Johan Atletiko, S.Kom

INFORMATION SYSTEM DEPARTMENT
Information Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2008

FA 2

**IMPLEMENTASI TEKNOLOGI JAVASCRIPT LIBRARY
EXT DAN JAVA SERVER PAGES PADA APLIKASI
WEB CUSTOMER SUPPORT DENGAN SISTEM
MULTIPLE TOUCHPOINT**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

RONNY ANSIS
NRP. 5204 100 053

Surabaya, Agustus 2008

KETUA

JURUSAN SISTEM INFORMASI



H.S.A. HOLLIL NOOR ALI M.KOM
NIP 131 996 150

**IMPLEMENTASI TEKNOLOGI JAVASCRIPT LIBRARY
EXT DAN JAVA SERVER PAGES PADA APLIKASI
WEB CUSTOMER SUPPORT DENGAN SISTEM
MULTIPLE TOUCHPOINT**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

RONNY ANSIS
NRP. 5204 100 053

Disetujui Tim Penguji :

Tanggal Ujian : 28 Juli 2008
Periode Wisuda : Oktober 2008


Mudjahidin, M.T

(Pembimbing I)


Faizal Johan Atletiko, S.Kom

(Pembimbing II)


Mahendrawathi Er, Ph.D

(Penguji I)


Arif Wibisono, S.Kom

(Penguji II)

IMPLEMENTASI TEKNOLOGI JAVASCRIPT LIBRARY EXT DAN JAVA SERVER PAGES PADA APLIKASI WEB CUSTOMER SUPPORT DENGAN SISTEM MULTIPLE TOUCHPOINT

Nama Mahasiswa : RONNY ANSIS
NRP : 5204 100 053
Jurusan : Sistem Informasi FTif ITS
Dosen Pembimbing : Mudjahidin, ST, MT
Faizal Johan Atletiko, S.Kom

Abstrak

Para ahli memiliki definisi yang berbeda-beda mengenai Customer relationship management (CRM). Secara umum, dapat disimpulkan bahwa CRM merupakan strategi dan usaha untuk menjalin hubungan dengan pelanggan dan memberikan pelayanan yang memuaskan bagi pelanggan. CRM mendukung suatu perusahaan untuk menyediakan pelayanan kepada pelanggan secara real time dengan menjalin hubungan dengan tiap pelanggan yang berharga melalui penggunaan informasi tertentu. Aplikasi yang menerapkan CRM diklasifikasikan menjadi dua (Dyche 2002), yaitu CRM Operasional dan CRM Analitik. Salah satu penerapan CRM yang termasuk dalam kategori CRM operasional adalah dalam bentuk aplikasi berbasis web. Melalui web, suatu perusahaan dapat mengimplementasikan berbagai aktifitas dan proses bisnisnya tanpa harus melakukan penyesuaian yang rumit terhadap lingkungan platform yang dimiliki.

Untuk dapat memahami hal tersebut akan dibangun sebuah aplikasi web Customer Support dengan sistem multiple touchpoint. Dimana aplikasi web yang dibangun menerapkan konsep Rich Internet Applications (RIA) dengan menggunakan teknologi dari JavaScript Library Ext sebagai basis pemrograman client side-nya. Aplikasi web ini akan dipergunakan dengan sistem yang akan mengkomunikasikan permasalahan yang dikirim pelanggan melalui berbagai media aplikasi pendukung lain (multiple touchpoint).

Media dari multiple touchpoint tersebut antara lain aplikasi web sisi pelanggan, aplikasi dengan protokol nirkabel atau Wireless Application Protocol (WAP) dan aplikasi layanan pesan singkat atau SMS (Short Message Service). Selain itu aplikasi web yang dibangun menyediakan berbagai informasi yang dapat mendukung kebutuhan pelanggan dalam memahami berbagai permasalahan yang dimilikinya.

Dengan adanya penelitian ini diharapkan dapat memberikan gambaran dan pengetahuan mengenai aplikasi berbasis web dengan menggunakan konsep RIA dan bagaimana JavaScript Library Ext dapat dimanfaatkan dalam implementasinya serta penerapan sistem dengan multiple touchpoint pada aplikasi Customer Support tersebut di dunia bisnis.

Kata Kunci: Rich Internet Application, Customer Support, Java, E-Business

IMPLEMENTATION OF EXT JAVASCRIPT LIBRARY TECHNOLOGY AND JAVA SERVER PAGES ON WEB CUSTOMER SUPPORT APPLICATION WITH MULTIPLE TOUCHPOINT SYSTEM

Name : RONNY ANSIS
Registration Number : 5204 100 053
Department : Sistem Informasi FTIf ITS
Supervisors : Mudjahidin, ST, MT
Faizal Johan Atletiko, S.Kom

Abstract

Some experts have different definition about Customer relationship management (CRM). But in common, CRM is a strategy and effort in order to make a relation and satisfying serving to all customer. CRM supports a company to provide a real time service and relation to each valuable customer through certain information. An application which is using CRM was classified into two kind of category, operational CRM and Analytical CRM. One of operational CRM implementation is a web-based application. A company can implement sort of activities and business process without complex adjustment to their environment platform just by using web.

As a mean to understand those things, this final project will develop a web-based customer support application with multiple touchpoint system. This application will be applied by using Rich Internet Applications (RIA) concept with JavaScript Library Ext technology as a client-side program based and will be used by a system to communicate problems which occasionally sent by customer through wide various support application media (multiple touchpoint). one of these multiple touchpoint are client-side web application, wireless application protocol (WAP) application, and short message service (SMS) application.

What is expected of this research is to provide an outlook and knowledge about web-based application with RIA concept and how to utilizing JavaScript Library Ext in its implementation. And of

course implementation this multiple touchpoint system on customer support application into business world.

Keywords: Rich Internet Application, Customer Support, Java, E-Business

KATA PENGANTAR

Tak ada kata yang terucap selain Syukur Alhamdulillah atas segala petunjuk, pertolongan, kasih sayang, dan ridho Allah SWT, sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul "IMPLEMENTASI TEKNOLOGI JAVASCRIPT LIBRARY EXT DAN JAVA SERVER PAGES PADA APLIKASI WEB CUSTOMER SUPPORT DENGAN MULTIPLE TOUCHPOINT", yang merupakan salah satu syarat kelulusan pada Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Buku ini merupakan dokumentasi dari pengerjaan tugas akhir yang dilakukan oleh penulis.

Tugas akhir ini tidak akan pernah terwujud tanpa dukungan, saran dan bantuan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu penulis baik dalam pelaksanaan Tugas akhir hingga selesainya penyusunan laporan tugas akhir ini. Ucapan terima kasih penulis sampaikan kepada :

- Bapak Mudjahidin, S.T, M.T dan Bapak Faizal Johan Atletiko, S.Kom selaku dosen pembimbing penulis yang telah banyak memberikan arahan, nasihat yang berharga, bimbingan, waktu, dan bantuan selama penyusunan tugas akhir.
- Ibu Mahendrawathi Er, S.T., M.Sc., Ph.D dan Bapak Arif Wibisono S.Kom selaku penguji tugas akhir.
- Kedua Orang Tuaku, Mama dan Papa yang selalu mendo'akan keberhasilan pendidikan seluruh anak-anaknya.
- Kakakku Ferdiansis dan Harry Ansis serta adikku Syukri Ansis yang selalu memotivasi semangatku untuk menyelesaikan tugas akhir ini.
- Pihak Suryasoft Konsultama, khususnya mas Aries Tony "Cassava" yang telah membangun kemampuanku dalam hal web programming. Mas Teguh Hardiansyah dan Bacus SI-2003 yang telah mengenalkanku pada Ext Js.

- Teman-teman yang kuanggap “guru” dalam pengerjaan tugas akhir ini, Faizal, Noval, Sukma, Kiki, Sonif, mas Soni, Artatik dan Anisah.
- Teman-teman seperjuangan yang mengerjakan tugas akhir dan selalu memotivasi, Nanda, Denny, Ivan, mas Agus, Andini, Finila, Icha, Krisna, Liya, Ndun, Rizka dan Sanny.
- Narsiis 2004 atas persahabatan selama 4 tahun yang indah ini. Terima kasih sahabat.
- Teman-teman SI Angkatan 2001, 2002, 2003, 2005, 2006 dan 2007 atas kerjasamanya selama ini.
- Pihak-pihak ”luar” yang tidak secara langsung telah membantu mendo’akan dan memotivasi, Wita, RaNi, Rieska dan Yuni.
- Serta semua pihak yang tidak dapat disebutkan satu persatu yang telah ikut membantu baik secara langsung maupun tidak langsung selama penulisan tugas akhir ini.

Surabaya, Agustus 2008

Penulis

DAFTAR ISI

Abstrak	v
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Manfaat	3
1.4 Permasalahan.....	3
1.5 Batasan Permasalahan	4
1.6 Metodologi	4
1.7 Sistematika Penulisan.....	6
BAB II DASAR TEORI	9
2.1 Customer Support.....	9
2.1.1 Definisi Customer Support	9
2.1.2 Aplikasi Customer Support.....	10
2.1.3 Fungsi-Fungsi Utama yang Perlu Disediakan.....	12
2.1.4 Arsitektur Sistem	13
2.2 Multiple Touchpoint.....	13
2.2.1 Deskripsi Multiple Touchpoint	14
2.2.2 Jenis Multiple Touch-Point.....	15
2.3 Ext (JavaScript Library)	17
2.3.1 Deskripsi Ext	17
2.3.2 Fungsionalitas yang Didukung Ext.....	17
2.3.3 Kelebihan dan Kelemahan Ext.....	18
2.3.4 JavaScript Object Notation (JSON).....	19
2.4 MySQL.....	20
2.4.1 Deskripsi MySQL.....	20
2.4.2 Fitur-fitur MySQL	21
BAB III ANALISIS KEBUTUHAN DAN PERANCANGAN	
PERANGKAT LUNAK	23
3.1 Analisis Kebutuhan	23
3.1.1 Pengguna	23
3.1.2 Teknologi.....	24

3.1.3	Fungsi	26
3.2	Deskripsi Umum	28
3.3	Arsitektur Sistem	29
3.4	Use Case Diagram	30
3.4.1	Deskripsi Aktor	30
3.4.2	Deskripsi Use Case	32
3.5	Sequence Diagram	32
3.6	Class Diagram	32
3.7	Perancangan Basis Data	32
3.8	Desain Antarmuka Aplikasi	35
3.8.1	Desain antarmuka aplikasi web Customer Support	36
3.8.2	Desain antarmuka aplikasi WAP Customer Support	36
BAB IV IMPLEMENTASI PERANGKAT LUNAK		39
4.1	Lingkungan Implementasi	39
4.2	Implementasi JavaScript Library Ext	40
4.2.1	Grid Panel	45
4.2.2	Tree	49
4.2.3	Toolbar	50
4.2.4	Accordion	51
4.2.5	Window	51
4.2.6	Viewport	53
4.3	Implementasi Teknologi Java	54
4.4	Implementasi Teknologi WAP	55
4.5	Implementasi Aplikasi SMS dengan GAMMU	57
4.6	Implementasi Proses	60
4.6.1	Proses Koneksi dengan Basis Data	60
4.6.2	Proses Login	62
4.6.3	Proses Menampilkan Data	63
4.6.4	Proses Menambahkan Data	67
4.6.5	Proses Mengubah Data	69
4.6.6	Proses Mencari Data	70
4.6.7	Proses Menampilkan Halaman Dinamis	72
4.7	Implementasi Basis Data	73
BAB V UJI COBA DAN EVALUASI		75
5.1	Hasil Penerapan JavaScript Library Ext	75
5.2	Lingkungan Uji Coba	76
5.3	Skenario Uji Coba	77
5.4	Pelaksanaan Skenario Uji Coba	77
5.4.1	Login dan Logout	78

5.4.2	Mengelola Data Pengguna	81
5.4.3	Mengelola Data Pesan	89
5.4.4	Mengelola Data Knowledgebase	106
5.4.5	Menampilkan Informasi yang Disediakan oleh Perusahaan 109	
5.5	Perbandingan Proses	115
BAB VI KESIMPULAN DAN SARAN		117
6.1	Kesimpulan	117
6.2	Saran.....	118
DAFTAR PUSTAKA.....		119
LAMPIRAN A		A-1
LAMPIRAN B.....		B-1
LAMPIRAN C.....		C-1



DAFTAR GAMBAR

Gambar 2.1 Menunjukkan sebuah arsitektur sistem aplikasi Customer Support	14
Gambar 2.2 Menunjukkan sebuah pendeklarasian tipe dokumen XHTML MP pada halaman WAP	16
Gambar 2.3 Contoh format data JSON	20
Gambar 3.1 Use Case Diagram sistem Aplikasi	31
Gambar 3.2 Model Data Konseptual aplikasi Customer Support	33
Gambar 3.3 Model Data Fisik aplikasi Customer Support	34
Gambar 3.4 Desain antarmuka aplikasi Web Customer Support	37
Gambar 3.5 Desain antarmuka aplikasi WAP Customer Support	38
Gambar 4.1 Kode program halaman contoh.jsp	41
Gambar 4.2 Kode program membuat data statis pada file contoh.js	42
Gambar 4.3 Kode program membuat store untuk menampung data pada file contoh.js	43
Gambar 4.4 Kode program membuat Grid Panel pada file contoh.js	44
Gambar 4.5 Halaman contoh.jsp	45
Gambar 4.6 Kode program membuat Data Store	46
Gambar 4.7 Kode program halaman Java Server Pages untuk meletakkan hasil Query pada halaman Back-End	47
Gambar 4.8 Kode program file Java pada Back-End untuk membentuk String dengan format JSON	47
Gambar 4.9 Contoh hasil pengembalian data String JSON	48
Gambar 4.10 Kode program membuat Grid Panel dengan paging	48
Gambar 4.11 Kode program untuk membuat Tree	49
Gambar 4.12 Kode program JSON untuk membuat data menu untuk Tree	50
Gambar 4.13 Kode program membuat Toolbar	51
Gambar 4.14 Kode program membuat menu Accordion	52
Gambar 4.15 Kode program membuat Window	52
Gambar 4.16 Kode program membuat layout web menggunakan Viewport	53
Gambar 4.17 Gambaran proses yang terjadi	54
Gambar 4.18 Kelas Java	55
Gambar 4.19 Kode program halaman Java Server Pages menggunakan Method pada Kelas Java	56
Gambar 4.20 Deklarasi tipe dokumen XHTML MP pada halaman Java Server Pages	56
Gambar 4.21 Window Device manager	58

Gambar 4.22 Port modem yang digunakan	58
Gambar 4.23 Konfigurasi port pada file gammurc	59
Gambar 4.24 Identifikasi lingkungan sistem GAMMU berjalan.....	60
Gambar 4.25 Proses pembuatan koneksi Basis Data	61
Gambar 4.26 Proses penutupan koneksi Basis Data	61
Gambar 4.27 Kode Program untuk melakukan proses login	62
Gambar 4.28 Form Login pada halaman logon.jsp.....	63
Gambar 4.29 Kode program mengambil data dari Basis Data dengan Ext.	64
Gambar 4.30 Kode program mengambil data yang dilakukan pada file Kelas setting	65
Gambar 4.31 Kode program tampilan kolom pada Grid Panel.....	65
Gambar 4.32 Kode program fungsi Render tampilan kolom Grid Panel....	66
Gambar 4.33 Kode program untuk menambahkan fungsi Paging	66
Gambar 4.34 Tampilan grid dengan paging	67
Gambar 4.35 Kode program untuk validasi data masukan	68
Gambar 4.36 Tampilan pesan kesalahan validasi inputan	68
Gambar 4.37 Kode program untuk menambahkan data	68
Gambar 4.38 Kode program untuk menampilkan Window Form Ubah data pengguna.....	69
Gambar 4.39 Kode program untuk mengubah data	70
Gambar 4.40 Form filter pencarian.....	71
Gambar 4.41 Kode Program untuk fungsi pencarian.....	71
Gambar 4.42 Kode program untuk Queri pada basis data	72
Gambar 4.43 Kode program untuk update isi dari Panel	72
Gambar 4.44 Kode program panel dengan isi halaman Java Server Pages	73
Gambar 4.45 Gambaran implementasi basis data pada sistem aplikasi.....	74
Gambar 5.1 Halaman Form Login.....	78
Gambar 5.2 Pesan kesalahan jika user ID atau password salah.....	79
Gambar 5.3 Halaman utama admin	79
Gambar 5.4 Personalisasi pengguna	80
Gambar 5.5 Logout berhasil	80
Gambar 5.6 Form menampilkan pesan peringatan bahwa nomor identitas tidak sesuai dengan ketentuan	82
Gambar 5.7 Form menampilkan pesan peringatan input alamat email tidak sesuai dengan format alamat email yang valid	82
Gambar 5.8 Pesan peringatan inputan belum valid	83
Gambar 5.9 Pesan peringatan gagal menyimpan data	83
Gambar 5.10 Pesan status sukses menyimpan data	84
Gambar 5.11 Daftar pengguna.....	85

Gambar 5.12 Memilih pengguna untuk diubah	85
Gambar 5.13 Pesan peringatan ketika tidak ada data yang dipilih	86
Gambar 5.14 Form edit data pengguna.....	86
Gambar 5.15 Pesan status data berhasil diubah	87
Gambar 5.16 Memilih pelanggan untuk dinon-aktifkan.....	87
Gambar 5.17 Pesan konfirmasi aksi non-aktifkan pelanggan.....	88
Gambar 5.18 Pesan status aksi non-aktifkan pengguna berhasil	88
Gambar 5.19 Form untuk melakukan pencarian data pelanggan	88
Gambar 5.20 Daftar kasus web aktif yang dimiliki pelanggan.....	89
Gambar 5.21 Form buat kasus dan pesan baru	90
Gambar 5.22 Pesan status pengiriman atau buat kasus dan pesan baru berhasil	91
Gambar 5.23 Kasus pesan baru tampil pada daftar pesan.....	92
Gambar 5.24 Window baca pesan	92
Gambar 5.25 Form tulis pesan baru untuk membalas.....	93
Gambar 5.26 Pesan balasan tampil pada daftar pesan kasus	94
Gambar 5.27 Memilih data kasus untuk diubah statusnya.....	94
Gambar 5.28 Pesan status Ubah Status data kasus berhasil.....	95
Gambar 5.29 Data kasus yang telah diubah statusnya	95
Gambar 5.30 Form pencarian kasus	96
Gambar 5.31 Pesan status menutup / non-aktifkan kasus berhasil.....	96
Gambar 5.32 Daftar data kasus tidak aktif.....	97
Gambar 5.33 Openwave Simulator.....	97
Gambar 5.34 Form Login aplikasi WAP Customer Support	98
Gambar 5.35 Halaman utama aplikasi WAP Customer Support	99
Gambar 5.36 Form buat data kasus baru WAP Customer Support.....	99
Gambar 5.37 Daftar data kasus aktif WAP Customer Support.....	100
Gambar 5.38 Daftar data pesan pada kasus yang dipilih	100
Gambar 5.39 Halaman isi pesan WAP Customer Support	101
Gambar 5.40 Daftar data pesan WAP pada aplikasi web	101
Gambar 5.41 Daftar data pesan WAP pada aplikasi WAP	102
Gambar 5.43 Daftar pesan SMS baru	103
Gambar 5.44 Baca Pesan SMS	104
Gambar 5.45 Form balas pesan SMS.....	105
Gambar 5.46 Daftar pesan SMS terkirim	105
Gambar 5.47 Baca pesan SMS pada handphone	106
Gambar 5.48 Daftar kategori knowledgebase.....	107
Gambar 5.49 Menekan tombol Tambah untuk membuat kategori artikel baru	107

Gambar 5.50 Form tambah kategori baru.....	108
Gambar 5.51 Form edit dan update data kategori.....	108
Gambar 5.52 Menekan tombol Ubah status untuk mengubah status kategori artikel.....	109
Gambar 5.53 Window detail profil pengguna.....	110
Gambar 5.54 Panel daftar dan ringkasan data produk.....	111
Gambar 5.55 Window Detail data produk.....	112
Gambar 5.56 Dokumen detail data produk.....	112
Gambar 5.57 Chart produk.....	113
Gambar 5.58 Chart pengguna.....	114
Gambar 5.59 Window baca artikel pada menu knowledgebase.....	114
Gambar 5.60 Window baca istilah pada menu knowledgebase.....	115

DAFTAR TABEL

Tabel 2.1 Kelemahan Ext	18
Tabel 2.2 Keunggulan Ext	19
Tabel 4.1 Spesifikasi Perangkat keras untuk implementasi sistem.....	39
Tabel 4.2 Teknologi-teknologi yang Digunakan untuk Implementasi Sistem	40
Tabel 5.1 Lingkungan Pengujian Aplikasi	76

BAB I

PENDAHULUAN

1.1 Latar Belakang

Customer Relationship Management (CRM) menjadi istilah yang beberapa tahun terakhir semakin populer. Ditambah dengan perkembangan teknologi informasi yang semakin merambah berbagai aplikasi bisnis. CRM menjadi salah satu proses bisnis yang menarik untuk diimplementasikan. Kegiatan pemasaran pada CRM mengelola seluruh aspek dari daur hidup pelanggan, mulai dari *acquisition*, *fullfilment*, hingga *retention*. CRM merupakan strategi komprehensif dari perusahaan agar setiap proses dari daur hidup pelanggan itu dapat dimanfaatkan dengan optimal.

Berhubungan dengan hal di atas, perusahaan dapat mendesain program loyalitas (*loyalty program*) yang sesuai untuk pelangganya. Program loyalitas ini akan sangat membantu perusahaan di dalam mempertahankan pelanggan, meningkatkan kepuasan, dan menjaga agar pelanggan tidak tergiur oleh berbagai tawaran yang diberikan oleh kompetitor lain. Program *customer retention* tersebut diatas itulah yang menjadi salah satu inti utama dari aktifitas CRM.

Aplikasi *Customer Support* adalah salah satu jenis sistem aplikasi yang dapat dimanfaatkan untuk meningkatkan loyalitas dari pelanggan terhadap perusahaan. Sistem dirancang dengan menyediakan layanan yang dapat membantu pelanggan dalam hal dukungan informasi terhadap permasalahannya pada produk tertentu secara spesifik. Sistem aplikasi juga didesain sedemikian hingga mudah bagi pelanggan untuk menggunakannya serta mengaksesnya. Tidak terkecuali perusahaan itu sendiri dalam mengelola permasalahan yang disampaikan pelanggan dalam bentuk pesan-pesan tertentu. Penyampaian pesan-pesan tersebut dilakukan dalam bentuk suatu komunikasi antara pelanggan dan perusahaan melalui berbagai media. Media yang dimaksud di sini dapat dikatakan

sebagai suatu wujud *touchpoint* pelanggan terhadap sistem utama dari *Customer Support*. *Touchpoint* yang dapat dibangun antara lain seperti *call center*, pemanfaatan media layanan pesan singkat atau *Short Message Service* (SMS), aplikasi dengan protokol nirkabel atau *Wireless Application Protocol* (WAP) maupun aplikasi berbasis web seperti surat elektronik atau e-mail, forum, dan berbagai aplikasi dengan platform lainnya.

Aplikasi berbasis web dianggap sebagai landasan untuk menjalankan aktifitas bisnis dengan efektif dan efisien karena kemudahan terhadap aksesnya. Di lain hal, dengan meningkatnya kecepatan akses dan perkembangan penetrasi akses *broadband*, teknologi web semakin berevolusi dengan menawarkan pengalaman baru bagi pengguna dan meningkatkan utilitas aplikasi dengan sebuah konsep yang disebut *Rich Internet Application* (RIA). RIA memberikan kemampuan sebuah aplikasi web dengan antarmuka dan kemampuan yang mendekati aplikasi *desktop*.

Java sebagai salah satu bahasa pemrograman memiliki salah satu jenis teknologinya yang menerapkan konsep RIA. Teknologi terbaru tersebut adalah *JavaScript library Ext*.

Dalam penelitian ini penulis menggunakan pengembangan teknologi baru dari *Java* yaitu *JavaScript library Ext*. Dimana aplikasi ini memakai kode pemrograman *JavaScript* pada *client-side* scripting-nya dan *Java Server Pages* (JSP) pada *server-side* scripting-nya.

Penelitian ini difokuskan pada pengenalan dan eksplorasi terhadap kemampuan *JavaScript library Ext* dalam memberikan kemampuan serta kenyamanan pengelolaan aplikasi web *Customer Support* dengan sistem *multiple touchpoint*.

1.2 Tujuan

1. Membangun aplikasi *Customer Support* dengan sistem *multiple touchpoint*, dimana *touchpoint* yang digunakan antara lain memanfaatkan media layanan pesan singkat (SMS), aplikasi

dengan protokol nirkabel (WAP) dan aplikasi *Customer Support* berbasis web sisi pelanggan.

2. Membangun aplikasi *Customer Support* berbasis web yang menerapkan teknologi *JavaScript library Ext* pada sisi manajemen perusahaan maupun pada sisi pelanggan sesuai dengan desain yang ada.

1.3 Manfaat

Manfaat yang diberikan oleh tugas akhir ini adalah :

1. Membantu pihak manajemen perusahaan dalam mengelola permasalahan yang disampaikan pelanggan dalam bentuk pesan-pesan, serta mengelola berbagai informasi yang disediakan dimana dapat bermanfaat bagi pelanggan.
2. Membantu serta memudahkan pihak pelanggan dalam mengkomunikasikan permasalahannya pada produk tertentu melalui berbagai *touchpoint* yang disediakan, serta akses terhadap informasi yang disediakan perusahaan.
3. Memberikan nilai yang lebih terhadap strategi CRM perusahaan sehingga meningkatkan rasa kepuasan dan loyalitas pelanggan.
4. Mengetahui kemampuan teknologi *JavaScript library Ext* yang diterapkan pada suatu aplikasi web *Customer Support*.

1.4 Permasalahan

Permasalahan yang sekiranya muncul pada pengerjaan tugas akhir ini :

1. Bagaimana mendesain sebuah sistem aplikasi *Customer Support* yang berfungsi mengkomunikasikan antara pihak pelanggan dalam menyampaikan permasalahannya serta pihak manajemen perusahaan dalam memberikan berbagai solusi bagi permasalahan pelanggan tersebut ?

2. Bagaimana pengelolaan sistem terhadap data pesan permasalahan yang dikomunikasikan oleh pihak manajemen maupun pelanggan melalui *multiple touchpoint* serta pengelolaan informasi yang disediakan perusahaan ?
3. Bagaimana implementasi dari desain yang ada dengan menggunakan teknologi *JavaScript library Ext* ?

1.5 Batasan Permasalahan

Dari permasalahan yang telah disebutkan di atas, maka batasan-batasan dalam tugas akhir ini adalah sebagai berikut:

1. Aplikasi *Customer Support* ini dibangun hanya sebagai bentuk dari CRM operasional bukan CRM analitik.
2. Aplikasi yang dibangun bertujuan untuk mengelola pesan yang dikirimkan oleh pihak pelanggan dan pihak manajemen perusahaan, selain itu aplikasi web sisi manajemen dapat mengelola berbagai informasi tertentu.
3. *Multiple touchpoint* yang dibangun adalah dengan menggunakan atau memanfaatkan media aplikasi layanan pesan singkat (SMS), aplikasi dengan protokol nirkabel (WAP) dan aplikasi berbasis web.
4. Aplikasi ini diimplementasikan pada sistem jaringan lokal, bukan *host* secara *online* pada jaringan *world wide web* (www).
5. Sistem basis data yang digunakan adalah MySQL.

1.6 Metodologi

Pembuatan tugas akhir ini terbagi menjadi beberapa tahap pengerjaan yang tertera sebagai berikut :

1. Studi Literatur

Studi literatur dilakukan untuk mempelajari konsep pengembangan aplikasi *Customer Support*. Informasi ini didapat dari buku-buku referensi serta informasi di internet yang terkait dengan tugas akhir ini.

2. Analisa Data

Analisa Data dilakukan untuk mengetahui sejauh mana data yang dimiliki perusahaan tersebut bisa mendukung sistem yang akan dibuat, serta jenis data yang dikelola oleh sistem, sehingga bisa ditentukan kebutuhan akan data apa saja yang diperlukan untuk melengkapi data yang sudah ada tersebut.

3. Perancangan dan Desain aplikasi

Tahap ini adalah proses analisa dan desain sistem yang bisa mengelola data-data pesan serta data lain yang digunakan perusahaan. Desain tersebut diharapkan bisa menghasilkan sistem yang dapat memudahkan pihak pelanggan dan pihak perusahaan sehingga dapat membantu pengguna-pengguna tersebut dalam kegiatannya masing-masing. Desain yang dirancang akan disesuaikan dengan desain standar dari suatu aplikasi *Customer Support*.

4. Pengembangan Aplikasi

Tahap ini merupakan tahap pembuatan dan pengembangan aplikasi sesuai dengan desain sistem yang ditetapkan pada tahap sebelumnya. Sistem ini dikembangkan dengan teknologi *Java* dan *JavaScript library Ext* dengan menggunakan basis data *MySQL*.

5. Implementasi dan Uji Coba

Dalam tahap ini akan dilakukan implementasi dari sistem yang telah dibuat sekaligus akan dilakukan uji coba dengan menjalankan semua fungsi dan fitur yang telah didefinisikan sebelumnya. Dari uji coba tersebut akan dievaluasi sejauh mana sistem bisa membantu kinerja dari pihak pelanggan dan manajemen perusahaan dalam melakukan kegiatannya masing-

masing. Selain itu juga dievaluasi bagaimana hasil dari implementasi teknologi yang telah dipilih.

6. Penyusunan Buku Tugas Akhir

Pada tahap terakhir ini akan disusun buku sebagai dokumentasi dari pelaksanaan tugas akhir. Dokumen ini diharapkan dapat berguna bagi pembaca yang memiliki keinginan untuk mengkaji lebih lanjut atau untuk keperluan pengembangan sistem menuju ke arah kesempurnaan.

1.7 Sistematika Penulisan

Sistematika penulisan Laporan Tugas Akhir dibagi menjadi enam bab sebagai berikut:

BAB I PENDAHULUAN

Bagian ini berisi latar belakang, permasalahan, tujuan, batasan masalah, metodologi yang digunakan, serta sistematika pembahasan yang diterapkan dalam memaparkan Tugas Akhir ini.

BAB II DASAR TEORI

Pada bab ini dijelaskan mengenai teori-teori yang digunakan dalam pengerjaan Tugas Akhir ini, meliputi teori-teori tentang pemasaran dan teknologi-teknologi yang terlibat.

BAB III ANALISIS KEBUTUHAN DAN PERANCANGAN PERANGKAT LUNAK

Pada bab ini dijelaskan mengenai analisis kebutuhan dan perancangan perancangan perangkat lunak berdasarkan analisis yang telah dibuat.

BAB IV IMPLEMENTASI SISTEM

Pada bab ini dijelaskan mengenai pembuatan aplikasi berdasarkan rancangan sistem yang telah dibuat sebelumnya.

BAB V UJI COBA DAN EVALUASI

Pada bab ini dijelaskan mengenai proses uji coba aplikasi dengan menggunakan beberapa skenario uji coba.

BAB VI KESIMPULAN DAN SARAN

Bagian ini berisi kesimpulan dari seluruh proses pengerjaan Tugas Akhir beserta saran untuk proses pengembangan selanjutnya.

Halaman ini sengaja dikosongkan

BAB II DASAR TEORI

Pada bab ini dijelaskan mengenai teori-teori yang digunakan dalam pengerjaan Tugas Akhir ini, meliputi teori-teori tentang aplikasi *Customer Support*, *multiple touchpoint* dan teknologi-teknologi yang terlibat didalamnya.

2.1 Customer Support

Pembahasan mengenai *Customer Support* akan dibagi ke dalam beberapa sub-bab sebagaimana berikut.

2.1.1 Definisi Customer Support

Menurut Dyche, Jill (2002) pada buku *The CRM Handbook: A Business Guide to Customer Relationship Management*, Sebuah *call center* adalah suatu kantor terpusat yang digunakan untuk menerima dan mentransmisikan sejumlah permintaan melalui kontak telepon. Biasanya *call center* dioperasikan oleh perusahaan untuk mengurangi dukungan terhadap produk atau informasi. Teknologi *call center* memasuki pasar untuk mengurangi pekerjaan yang berulang dan meningkatkan efisiensi, membuat perusahaan dapat mengatasi volume kontak telepon yang masuk pada perusahaan. Produk dari *call center* menawarkan cara pengumpulan komplain pelanggan dan menawarkan solusi-solusi yang mungkin diberikan. Pada sistem *call center*, *Customer Service Representatives* (CSRs) bisa mencari permasalahan yang serupa pada sistem sementara seorang pelanggan menunggu diujung lain pada suatu hubungan kontak telepon.

Customer Support merupakan suatu layanan yang diberikan oleh perusahaan dimana layanan ini adalah bentuk lain dari *call center* yang umumnya dimiliki oleh perusahaan. *Customer Support* yang diimplementasikan ke dalam aplikasi dapat memberikan cara alternatif kepada perusahaan maupun pelanggan selain menerapkan



sistem *call center*. Melalui aplikasi *Customer Support* perusahaan dapat memberikan layanan *self-service* kepada pelanggan berupa dukungan terhadap produk atau informasi tertentu yang berhubungan dengan permasalahan dan komplain yang dimiliki pelanggan.

Selain itu, sebagaimana dijelaskan pada *Wikipedia*, *Customer Support* yang dapat ditinjau sebagai suatu bentuk dari *technical support* adalah kegiatan atau layanan (bisa bersifat teknis) yang disediakan oleh perusahaan dalam bentuk bantuan berupa informasi terhadap produk-produk tertentu kepada pelanggan. Umumnya layanan *Customer Support* berusaha untuk membantu pelanggan memecahkan permasalahan tertentu yang spesifik terhadap produk tertentu, lebih dari pada sekedar menyediakan pelatihan atau *training*, kustomisasi dari produk atau bentuk layanan bantuan lainnya.

Kebanyakan perusahaan menawarkan layanan bantuan terhadap bantuan mereka secara gratis maupun berbayar. Layanan yang disediakan oleh *Customer Support* bisa disampaikan melalui telepon atau melalui berbagai media *online* seperti e-mail atau situs web. Internet yang merupakan platform dari situs web juga merupakan sebuah sumber yang baik untuk sebuah bentuk bantuan layanan secara gratis dimana pelanggan bisa mendapatkan saran maupun pertolongan terhadap masalah spesifik mereka pada produk tertentu. Namun layanan *Customer Support* kebanyakan tidak disediakan secara gratis kepada siapapun melainkan disediakan kepada pelanggan yang merupakan anggota atau member dari sistem CRM yang dibangun oleh perusahaan itu sendiri.

2.1.2 Aplikasi Customer Support

Umumnya, *Customer Support* diimplementasikan ke dalam suatu bentuk sistem aplikasi berbasis *desktop* ataupun web guna memudahkan dalam pengelolaan data dan informasi pada basis data perusahaan yang disampaikan antara pelanggan dengan perusahaan. Banyak perusahaan menerapkan sistem *call center* dimana aplikasi *desktop* sebagai sarana mengelola basis data permasalahan

pelanggan dan telepon sebagai alat bantu sistem komunikasi permasalahan yang ditransmisikan antara pihak pelanggan dengan pihak manajemen perusahaan (dalam hal ini dapat sebagai operator). Namun tidak selamanya sistem ini dapat diterapkan pada semua kondisi. Hal tersebut dikarenakan tidak seluruhnya permasalahan yang dikemukakan pelanggan bisa secara langsung didapatkan solusinya melainkan ditampung terlebih dahulu kemudian diproses untuk selanjutnya permasalahan tersebut ditindaklanjuti. Proses yang dimaksud dapat berupa analisa permasalahan pelanggan sesuai dengan konteks subjek atau produk yang diangkat.

Berdasarkan beberapa perusahaan yang mengimplementasi dari sistem ini, aplikasi *Customer Support* dibangun dengan bentuk yang bermacam-macam sesuai dengan kebutuhannya.

Perusahaan produsen perangkat lunak atau *software* Windriver (www.windriver.com) memiliki suatu sistem aplikasi *Customer Support*, dimana spesifikasi fitur dari sistem aplikasi tersebut antara lain seperti berikut :

- Informasi spesifikasi dari suatu produk, termasuk FAQ, saran, dan catatan-catatan lainnya yang dapat membantu pelanggan.
- HTML atau dokumen PDF yang dapat diunduh dimana merupakan manual dari produk yang dimiliki perusahaan.
- Download Patch, perangkat lunak maupun driver.
- Diskusi permasalahan (teknis maupun non-teknis).
- Berita-berita terbaru yang dimiliki perusahaan.

Selain itu, Perusahaan LiveWorld (www.liveworld.com) merupakan salah satu jenis perusahaan yang bergerak dibidang CRM dimana menawarkan berbagai solusi strategi CRM berbentuk *tools* maupun sistem aplikasi. Salah satu solusi CRM yang dimilikinya adalah aplikasi *online Customer Support*. Pada situs yang dimilikinya dijelaskan bahwa aplikasi *online Customer Support* tersebut fokus terhadap bentuk sistem dimana permasalahan pelanggan dapat

disampaikan dalam suatu diskusi tanya jawab dengan pihak manajemen perusahaan. Selain itu aplikasi menawarkan pengalaman *self-service* kepada pelanggan karena ketersediaan berbagai informasi yang dapat membantunya.

Berdasarkan sedikit contoh gambaran diatas dan definisi sebelumnya tentang *Customer Support* dapat diambil bentuk umum bahwa aplikasi dirancang dengan fitur atau fasilitas utama yaitu adanya fungsi bagi para pengguna khususnya pelanggan untuk menyampaikan permasalahannya terhadap produk tertentu, serta bagi pihak manajemen perusahaan dalam mengelola permasalahan-permasalahan tersebut dalam rangka menyediakan berbagai solusi yang dituju. Selain itu terdapat fasilitas-fasilitas tambahan berupa informasi-informasi tentang produk, pengetahuan, berita dan sebagainya.

2.1.3 Fungsi-Fungsi Utama yang Perlu Disediakan

Dalam suatu aplikasi *Customer Support* terdapat fungsi utama yang diperlukan untuk mendukung segala proses pengelolaan permasalahan dalam bentuk pesan yang dikirim maupun diterima oleh para pengguna sistem. Fungsi tersebut merupakan fungsi yang digunakan sebagai fungsi dasar yang digunakan dalam mendesain sistem maupun antarmuka aplikasi ini.

Fungsi dasar pada aplikasi *Customer Support* yang akan dibangun pada tugas akhir ini adalah sebagai berikut.

1. Manajemen Pengguna

Fungsi ini menyediakan fasilitas dasar pada kebanyakan aplikasi manajemen, yaitu untuk mengelola pengguna seperti tambah, ubah, hapus pengguna.

2. Manajemen Pesan

Fungsi ini menyediakan fasilitas bagi pengguna terutama pihak manajemen perusahaan / administrator dalam hal mengelola

permasalahan dan solusi yang disampaikan oleh pengguna aplikasi dalam bentuk pesan-pesan.

3. Manajemen dan Menyediakan Informasi

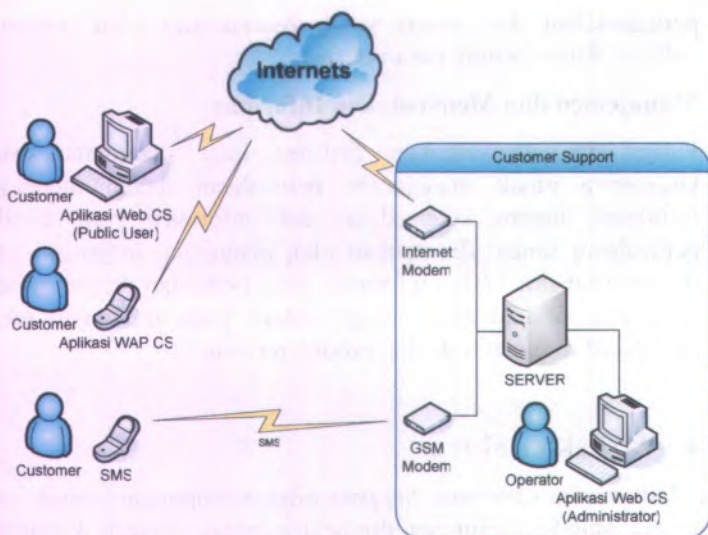
Fungsi ini menyediakan fasilitas bagi pengguna sistem khususnya pihak manajemen perusahaan dalam mengelola informasi dimana ketersediaan dari informasi yang dimiliki perusahaan sangat dibutuhkan oleh pengguna. Informasi yang dimaksud disini adalah informasi yang berkaitan dengan produk karena kebutuhan akan fungsi sistem yaitu dukungan teknis (*technical support*) terhadap produk tertentu.

2.1.4 Arsitektur Sistem

Arsitektur sistem *Customer Support* tidak mempunyai bentuk yang baku atau standar, sehingga dirancang sesuai dengan kebutuhan perusahaan masing-masing. Pada dasarnya terdapat aplikasi utama pada perusahaan yang berfungsi sebagai *server* dan aplikasi pendukung lainnya sebagai *touchpoint* pengguna atau pelanggan terhadap aplikasi utama tersebut. Berikut arsitektur dari sistem *Customer Support* dengan aplikasi berbasis web sebagai sistem utamanya dan *multiple touchpoint* yang dibangun.

2.2 Multiple Touchpoint

Bagian berikut ini akan membahas tentang *multiple touchpoint* serta media yang digunakan pada tugas akhir ini



Gambar 2.1 Menunjukkan sebuah arsitektur sistem aplikasi Customer Support

2.2.1 Deskripsi Multiple Touchpoint

Pada buku *The CRM Handbook* oleh Jill Dyche (2002), dalam suatu CRM operasional atau yang dapat dikenal sebagai "front-office" CRM, aktivitas yang terjadi melibatkan area dimana hubungan perusahaan dengan pelanggan dapat muncul secara langsung. Pada buku tersebut mengatakan interaksi-interaksi tersebut sebagai suatu *touchpoint* pelanggan. Sebuah *touchpoint* bisa menjadi batasan dalam kontak. Contohnya sebuah kontak telepon menuju sebuah *hotline Customer Support* perusahaan. Atau sebuah batasan luar kontak, contohnya sebuah telepon penjualan secara pribadi atau promosi melalui e-mail. Bentuk-bentuk implementasi dari *touchpoint* dapat berupa *call center*, *web access*, e-mail, *direct sales*, fax dan lain-lainnya.

2.2.2 Jenis Multiple Touch-Point

Berikut adalah beberapa jenis *multiple touchpoint* yang digunakan pada tugas akhir ini.

2.2.2.1 Aplikasi Web Sisi Pelanggan

Implementasi dari salah satu *touchpoint* ini merupakan aplikasi web yang menerapkan teknologi *JavaScript library Ext* sebagaimana aplikasi web sisi manajemen perusahaan. Fitur yang dimiliki aplikasi ini sama sebagaimana yang telah dijelaskan sebelumnya pada fungsi dasar aplikasi.

2.2.2.2 Aplikasi Protokol Nirkabel (WAP)

Wireless Application Protocol (WAP) adalah standar internasional untuk aplikasi yang menggunakan komunikasi nirkabel / *mobile*. Tujuan utama kegunaannya adalah akses terhadap internet melalui sebuah perangkat telepon selular (ponsel) atau *Personal Data Assistant* (PDA). Sebuah browser WAP pada ponsel menyediakan semua layanan dasar dari sebuah internet browser pada komputer tetapi lebih disederhanakan untuk dioperasikan dengan batasan sebuah perangkat telepon seluler, salah satunya seperti layar yang kecil.

Situs wap adalah situs web yang ditulis atau secara dinamis dikonversikan menjadi *Wireless Markup Language* (WML) dan diakses melalui browser WAP pada ponsel.

Sesuai dengan berkembangnya teknologi perangkat telepon seluler atau perangkat nirkabel maka WAP pun berkembang menjadi WAP 2.0 dimana teknologi bahasa markup yang digunakan tidak lagi menggunakan WML namun dengan *Extensible Hyper Text Markup Language Mobile Profile* (XHTML MP).

Implementasi WAP sebagai salah satu *touchpoint* pada sistem aplikasi *Customer Support* ini merupakan sebuah simulasi implementasi dari aplikasi yang dapat diakses melalui perangkat mobile seperti telepon seluler. Halaman yang digunakan adalah *Java Server Pages* dengan tipe dokumen bahasa *markup* XHTML.

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">

<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.1//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile11.dtd">

<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">
```

Gambar 2.2 Menunjukkan sebuah pendeklarasian tipe dokumen XHTML MP pada halaman WAP

2.2.2.3 Aplikasi Layanan Pesan Singkat (SMS)

Short Message Service (SMS) merupakan sebuah protokol komunikasi yang memfasilitasi suatu proses pertukaran pesan singkat antara perangkat telepon seluler. Layanan pesan SMS merupakan suatu aplikasi komunikasi data yang sangat luas dipakai pada seluruh penjuru dunia, dengan 2.4 milyar pengguna aktif atau 74% dari seluruh penggunaan perangkat telepon seluler yang ada didunia (sumber:wikipedia.org).

Teknologi SMS telah memfasilitasi pengembangan dari berbagai aplikasi pemasaran yang dimiliki perusahaan dalam rangka memperoleh keuntungan. Layanan seperti polling, quiz, iklan bahkan akses terhadap informasi tertentu sering diimplementasikan pada teknologi ini.

Implementasi layanan aplikasi SMS pada sistem aplikasi *Customer Support* ini merupakan sebuah implementasi dari salah satu *touchpoint* akan yang disediakan.

2.3 Ext (JavaScript Library)

Ext (disebut "extent") adalah *JavaScript library* berbasis open *source* yang membantu pengembang untuk membangun aplikasi web dengan lebih dinamis.

2.3.1 Deskripsi Ext

Ext adalah sebuah *JavaScript library* yang dapat digunakan untuk membangun sebuah aplikasi web yang interaktif dengan menggunakan teknik seperti AJAX, DHTML, dan DOM scripting. Ext dapat berkerja secara stand alone, namun dapat juga berjalan library yang berbeda seperti YUI, jQuery, atau Prototype. Saat ini versi terbaru Ext adalah versi 2.1, versi ini sedang dikembangkan dan masih dalam versi beta. Pada tugas akhir ini penulis menggunakan Ext versi 2.0.

2.3.2 Fungsionalitas yang Didukung Ext

Bermacam-macam bentuk kontrol atau "widget" yang tersedia untuk digunakan dalam aplikasi web antara lain:

text fields and textareas

- date fields dengan sebuah pop-up date-picker
- numeric fields
- comboboxes
- radio and checkbox
- html editor
- grid control (baik dengan model read-only ataupun model edit, data yang bisa diurutkan, kolom yang bisa di drag, dan bermacam-macam fitur yang lain)
- tree control

- tab panels
- toolbars
- Windows-style menus
- region panels untuk membuat form dapat dibagi kedalam beberapa sub bagian.
- Viewport merupakan layout dasar yang didesain untuk sebuah tampilan dari aplikasi web

Beberapa kontrol tersebut dapat berkomunikasi dengan *server* dengan menggunakan AJAX. Untuk penyimpanan data bisa dilakukan dengan format JSON atau bisa juga dengan format XML.

2.3.3 Kelebihan dan Kelemahan Ext

Sebagai sebuah *JavaScript library*, Ext memiliki kelebihan dan kelemahan.

Tabel 2.1 Kelemahan Ext

Titik kelemahan	Penjelasan
Keterbatasan versi <i>browser</i>	Ext tidak bisa berjalan pada semua versi browser. Ada versi browser yang tidak dapat menjalankan <i>JavaScript library</i> Ext. Kasus ini terjadi pada browser Internet Explorer versi 5, namun di Internet Explorer versi 6 Ext bisa berjalan. Walaupun tetap bisa jalan di Internet Explorer 6, terkadang masih ada fungsi yang tidak berjalan. Ext tidak menemui masalah bila dijalankan di browser Mozilla Firefox.
	Pada browser Internet Explorer kesalahan sedikitpun pada penulisan script Ext Js seperti adanya ekstra koma pada akhir suatu atribut dalam suatu fungsi maka, aplikasi tidak akan

	berjalan sama sekali.
--	-----------------------

Tabel 2.2 Keunggulan Ext

Kelebihan	Penjelasan
Aplikasi web lebih interaktif	Ext dapat digunakan untuk mengembangkan aplikasi web dengan lebih interaktif bila dibandingkan dengan pengembangan aplikasi web konvensional.

2.3.4 JavaScript Object Notation (JSON)

Pada aplikasi web ini Ext yang dikembangkan menggunakan format *JavaScript Object Notation* (JSON) dalam hal pertukaran data antara *client-side scripting* dengan *server-side programming*.

JSON adalah sebuah format pertukaran data yang ringan. JSON mudah bagi manusia untuk dibaca dan ditulis. Selain itu mudah bagi mesin untuk di-parsing dan di-generate. Dasar dari JSON adalah bahasa pemrograman *JavaScript*, Standar ECMA-262 edisi ke-3 – Desember 1999.

Selain itu, JSON sering digunakan untuk mentransmisikan data terstruktur melalui sebuah koneksi internet pada sebuah proses pemanggilan (*called*).

JSON merupakan sebuah format text yang secara penuh bersifat independen tapi dapat digunakan secara familiar digabungkan dengan bahasa C termasuk C++, C#, *Java*, *JavaScript*, Perl, Python dan lainnya. Kegunaan utamanya adalah pada aplikasi Web berbasis pemrograman AJAX, dimana JSON sebagai suatu alternatif untuk format data konvensional yang sering digunakan yaitu XML.

Contoh format data JSON pada gambar 2.3.

```

1  {
2      "namaDepan": "Sandra",
3      "namaBelakang": "Dewi",
4      "alamat": {
5          "jalan": "mawar 20",
6          "kota": "Surabaya",
7          "provinsi": "Jawa Timur",
8          "kodepos": 60235
9      },
10     "NomorTelepon": [
11         "031 - 8283690",
12         "085648659460"
13     ]
14 }
15

```

Gambar 2.3 Contoh format data JSON

2.4 MySQL

Berikut ini akan dibahas secara singkat mengenai MySQL sebagai basis data yang digunakan dalam pengerjaan tugas akhir ini.

2.4.1 Deskripsi MySQL

MySQL adalah sebuah *Relational Database Management System* (RDBMS) yang dimiliki oleh perusahaan *sun microsystem*. Program ini berjalan sebagai sebuah *server* yang menyediakan akses multi-user pada beberapa basis data. MySQL terkenal penggunaannya untuk aplikasi berbasis web dan berperan sebagai komponen basis data dari platform LAMP, BAMP, MAMP dan WAMP (Linux / BSD / Mac / Windows – Apache – MySQL – PHP / Perl / Python).

2.4.2 Fitur-fitur MySQL

Berikut ini akan dibahas fitur-fitur utama MySQL yang dapat digunakan dalam pengembangan tugas akhir ini:

- **Referential Integrity Constraint.**

Integrity Constraint adalah suatu cara untuk mendefinisikan sebuah aturan bisnis pada sebuah kolom di dalam tabel. Integrity constraint yang didukung MySQL adalah: NOT NULL, UNIQUE KEY, PRIMARY KEY, INDEX.

- **Encrypted connection via SSL**

Koneksi telah dienkrpsi via SSL. Yang membuat keamanan basis data lebih baik.

- **phpMyAdmin**

phpMyAdmin adalah sebuah tools berbasis web untuk mengelola basis data MySql. Dengan tool ini pengguna bisa memasukkan dan menjalankan statement-statement SQL serta melakukan banyak tugas tambahan lainnya dengan baik.

Halaman ini sengaja dikosongkan

BAB III

ANALISIS KEBUTUHAN DAN PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan diuraikan hal-hal yang berkaitan dengan perancangan sistem aplikasi *Customer Support*. Perancangan yang dibahas meliputi perancangan proses, perancangan basis data dan aturan-aturan yang diperlukan dalam pengembangan sebuah aplikasi.

3.1 Analisis Kebutuhan

Kebutuhan pengguna terhadap sistem dapat dijabarkan sebagaimana pada bagian berikut ini.

3.1.1 Pengguna

Klasifikasi pengguna dalam sistem aplikasi ini dibedakan menjadi:

- **Administrasi / Manajemen Perusahaan**

Bagian Administrasi yang merangkap sebagai pihak manajemen perusahaan mengurus masalah-masalah yang berhubungan dengan dasar pengelolaan pengguna seperti tambah, ubah profil serta hapus pengguna. Pengguna ini juga mengurus masalah-masalah yang berhubungan dengan pengelolaan permasalahan yang disampaikan pelanggan dalam bentuk pesan melalui *multiple touchpoint* yang disediakan. Selain itu pihak manajemen perusahaan dapat mengelola berbagai informasi yang akan disediakan bagi pengguna sistem.

- **Pelanggan**

Bagian pelanggan mengurus hal yang berhubungan dengan ubah profil serta pengelolaan permasalahan yang dimilikinya. Namun pada aplikasi *touchpoint* yang lain yaitu aplikasi SMS

dan WAP, pengguna hanya bisa menyampaikan permasalahan tanpa bisa mengubah profil akun yang dimilikinya.

3.1.2 Teknologi

Beberapa teknologi dipilih berdasarkan kebutuhan perusahaan, sedangkan sebagian lainnya ditentukan sendiri oleh penulis karena tidak ada kebutuhan khusus atas teknologi tersebut.

- **Web**

Alasan penggunaan aplikasi utama dengan teknologi berbasis web ialah kemudahannya dalam hal akses dari mana saja menggunakan perangkat komputer yang terhubung dengan jaringan internet dan dapat berjalan pada berbagai platform sistem operasi (multi-platform), serta tidak adanya konfigurasi tertentu yang diperlukan untuk menjalankan aplikasi.

- **Java Server Pages**

Java Server Pages (JSP) merupakan sebuah teknologi servlet-based yang digunakan pada web tier untuk menghadirkan dynamic dan static content. *Java Server Pages* merupakan text-based dan kebanyakan berisi template text HTML yang digabungkan dengan spesifik tags dynamic content. Tidak ada kebutuhan khusus dari perusahaan atas penggunaan teknologi *Java Server Pages*.

- **MySQL**

Sebagai media penyimpanan yang telah sering digunakan dan dapat memberikan hasil optimal dari sisi kecepatan, kemudahan serta reliabilitas manajemen data pada penggunaan jumlah data skala menengah kebawah. Selain itu MySQL dapat berjalan pada berbagai platform sistem operasi. Tidak ada kebutuhan khusus dari perusahaan atas penggunaan teknologi MySQL.

- **Ext (*JavaScript Library*)**

Ext digunakan sebagai teknologi yang mengurus bentuk dari antarmuka aplikasi web yang dikembangkan pada tugas akhir ini. Tidak ada kebutuhan khusus dari perusahaan atas penggunaan teknologi ini.

- **Wireless Application Protocol (WAP) 2.0 dengan XHTML MP**

Extensible Hypertext Markup Language Mobile Profile (XHTML MP) merupakan sebuah bahasa markup yang didefinisikan dalam *Wireless Application Protocol* (WAP) 2.0, yaitu sebuah protokol komunikasi untuk aplikasi-aplikasi nirkabel yang dibuat oleh WAP forum. Aplikasi dengan bahasa XHTML ini dikhususkan untuk membangun aplikasi-aplikasi yang dapat dibaca melalui perangkat-perangkat mobile, seperti : telepon seluler (ponsel), PDA ataupun smartphone. Aplikasi mobile yang dibangun dengan menggunakan XHTML MP ini selain dapat dibaca melalui browser yang ada di dalam ponsel juga dapat dibaca melalui internet browser pada komputer.

Aplikasi WAP *Customer Support* ini merupakan salah satu *touchpoint* sistem aplikasi yang dapat diakses oleh pelanggan melalui perangkat-perangkat mobile. Tidak ada kebutuhan khusus dari perusahaan atas penggunaan teknologi ini.

- **Short Message Service (SMS)**

Layanan pesan singkat atau SMS digunakan sebagai salah satu dari *touchpoint* sistem aplikasi yang disediakan. Pengguna dapat mengirimkan permasalahannya dengan mudah hanya dengan mengirim pesan pada nomor layanan SMS perusahaan. Tidak ada kebutuhan khusus perusahaan atas penggunaan teknologi ini.

3.1.3 Fungsi

Fungsi-fungsi sistem ini dibagi berdasarkan platform sistem aplikasi yang digunakan yaitu, aplikasi web, aplikasi WAP dan aplikasi SMS.

3.1.3.1 Aplikasi web sisi manajemen perusahaan

- **Administrasi**

Sebuah bentuk aplikasi manajemen harus memiliki fungsi dasar dalam masalah administrasi sistem. Administrasi yang dimaksud disini adalah pengelolaan pengguna seperti tambah, ubah profil, hapus pengguna.

- **Pengelolaan pesan**

Aplikasi *Customer Support* berfungsi sebagai sarana bagi pelanggan untuk menyampaikan permasalahannya pada produk tertentu dalam bentuk pesan kepada perusahaan. Pesan tersebut disimpan dalam bentuk data-data yang disimpan pada basis data. Oleh karena, itu diperlukan adanya suatu fungsi pengelolaan pesan pada aplikasi web sisi manajemen perusahaan guna membagi pesan berdasarkan properties yang dimilikinya (misal : pengirim, tanggal, subjek, status dan lain-lain).

Selain itu, pengelolaan pesan juga mencakup fungsi untuk membuat atau membalas dan mengirim pesan tertentu kepada pelanggan.

- **Pengelolaan informasi**

Aplikasi web *Customer Support* memiliki fungsi dasar pengelolaan informasi yang nantinya informasi tersebut diharapkan dapat membantu dan berguna bagi pelanggan.

3.1.3.2 Aplikasi web sisi pelanggan

- **Administrasi**

Fungsi ini merupakan fungsi dasar bagi pelanggan untuk mengelola profil yang dimilikinya.

- **Pengelolaan pesan**

Pada fungsi ini pelanggan dapat melakukan berbagai aktifitas yang berhubungan dengan pesan yang dimilikinya seperti membuat pesan baru, membalas, mengirim serta membaca pesan tertentu.

3.1.3.3 Aplikasi WAP

- **Buat / baca / balas pesan**

Fungsi ini merupakan bentuk lain dari fungsi yang dimiliki pada aplikasi web sisi pelanggan dimana pelanggan dapat melihat pesan yang dimilikinya.

Namun pada aplikasi ini tidak terdapat fungsi administrasi atau kelola profil karena keterbatasannya aplikasi berbasis WAP dalam menampilkan antarmuka dibandingkan dengan aplikasi berbasis web.

Aplikasi SMS

- **Buat / baca / balas pesan**

Fungsi ini merupakan salah satu fungsi dasar yang dimiliki pada perangkat telepon seluler yaitu aplikasi SMS. Pada aplikasi ini pelanggan dapat mengirim pesan baru, membaca serta membuat pesan balasan.

3.2 Deskripsi Umum

Aplikasi *Customer Support* ini adalah sebuah sistem yang membantu pelaksanaan proses penyampaian permasalahan pelanggan dalam bentuk pesan kepada pihak manajemen perusahaan, serta sebaliknya pihak manajemen dalam menyampaikan bantuan informasi maupun solusi terhadap permasalahan pelanggan tersebut. Selain itu aplikasi web *Customer Support* menyediakan berbagai informasi yang dapat membantu pengguna sistem khususnya pelanggan. Tujuan pembuatan aplikasi dengan sistem *multiple touchpoint* ini adalah untuk memberikan kemudahan bagi pihak pelanggan maupun manajemen perusahaan dalam menjalankan fungsi dan proses tersebut di atas.

Seperti dijelaskan sebelumnya bahwa pengguna sistem aplikasi *Customer Support* ini adalah pelanggan serta manajemen perusahaan dan diimplementasikan pada aplikasi web, SMS dan WAP.

Aplikasi web dan WAP *Customer Support* digunakan oleh pelanggan untuk menyampaikan permasalahan yang dimilikinya dalam bentuk kasus tertentu. Tiap pelanggan memiliki berbagai kasusnya masing-masing yang akan tampil pada aplikasi web dan WAP berdasarkan akun yang dimilikinya. Tiap kasus memiliki subjek kasus dan jenis produk yang dipermasalahkan atau akan dibahas. Ketika pelanggan membuat suatu kasus baru maka dapat dikatakan kasus tersebut merupakan suatu bentuk *thread* yang dapat dibahas secara terus menerus dengan pihak manajemen selama konteksnya masih berhubungan dengan subjek kasus yang dibuat. Ketika kasus tersebut telah selesai dibahas atau telah dapat diatasi maka pihak manajemen akan menutup kasus tersebut sehingga tidak akan muncul pada tampilan aplikasi sisi pelanggan.

Untuk aplikasi SMS, pesan yang disampaikan oleh pelanggan tidak terdapat ketentuan khusus dalam hal format penyampaiannya. Pelanggan dapat secara langsung menyampaikan permasalahan yang dimilikinya pada produk tertentu dengan mengirim SMS. Dan ketika pesan SMS terkirim atau sampai pada basis data SMS perusahaan,

pihak manajemen dapat secara langsung membaca dan membalas pesan tersebut melalui aplikasi web *Customer Support* sisi manajemen.

Pengelolaan pesan-pesan tersebut dipisahkan berdasarkan *touchpoint* yang digunakan pelanggan dalam cara penyampaiannya. Sehingga pihak manajemen dapat mengelola atau menindaklanjuti pesan-pesan tersebut secara terpisah. Sama halnya dengan pada pihak manajemen, pesan yang dimiliki oleh pelanggan dapat dilihat secara terpisah berdasarkan cara penyampaiannya. Namun dengan satu *account* yang dimilikinya pelanggan dapat mengakses sistem melalui berbagai *touchpoint* tersebut.

Pada aplikasi web *Customer Support* sisi manajemen perusahaan pihak manajemen dapat mengelola mengelola pengguna, pesan yang disampaikan melalui aplikasi web, SMS dan WAP serta pengelolaan informasi yang akan ditampilkan pada aplikasi web *Customer Support*.

Keberadaan aplikasi *Customer Support* ini diharapkan dapat membantu pelanggan dalam menyelesaikan permasalahannya pada produk tertentu. Selain itu juga diharapkan dapat membantu pihak manajemen dalam mengelola serta merespon permasalahan yang disampaikan pelanggan tersebut sehingga dapat meningkatkan kepuasan pelanggan terhadap pelayanan yang diberikan oleh perusahaan.

3.3 Arsitektur Sistem

Aplikasi utama dari sistem *Customer Support* ini adalah sebuah aplikasi berbasis web yang dikembangkan dengan menggunakan teknologi *Java*. Web Server yang digunakan adalah Apache Tomcat dengan bahasa pemrograman *Java Server Pages*. Pengembangan aplikasi ini juga didukung dengan Ext (*JavaScript Library*). Untuk koneksi basis data, aplikasi ini menggunakan JDBC standar dan basis data yang digunakan adalah MySQL. Gambar arsitektur aplikasi sistem *Customer Support* dapat dilihat pada (Gambar 2.1).

Untuk aplikasi *Customer Support* yang merupakan *touchpoint* yang disediakan adalah aplikasi web, SMS serta WAP.

Aplikasi web sisi pelanggan sama halnya dengan aplikasi web sisi manajemen perusahaan. Aplikasi SMS menggunakan software core untuk aplikasi SMS center sebagai suatu bentuk dari modul SMS gateway dimana basis data yang digunakan adalah MySQL. Dan untuk aplikasi WAP menggunakan bahasa pemrograman *Java Server Pages* dengan tipe bahasa markup XHTML.

3.4 Use Case Diagram

Perancangan diagram use case aplikasi sistem *Customer Support* ini merupakan gambaran fungsi pada fungsi-fungsi utama yang tersedia pada aplikasi web, WAP dan SMS dalam kegiatan pengelolaan pesan serta proses bisnis lainnya. Use case diagram sistem aplikasi dapat dilihat pada Gambar 3.1.

3.4.1 Deskripsi Aktor

Berikut ini adalah penjelasan dari aktor-aktor yang terlibat di dalam sistem sebagaimana terlihat pada keseluruhan diagram use case di atas.

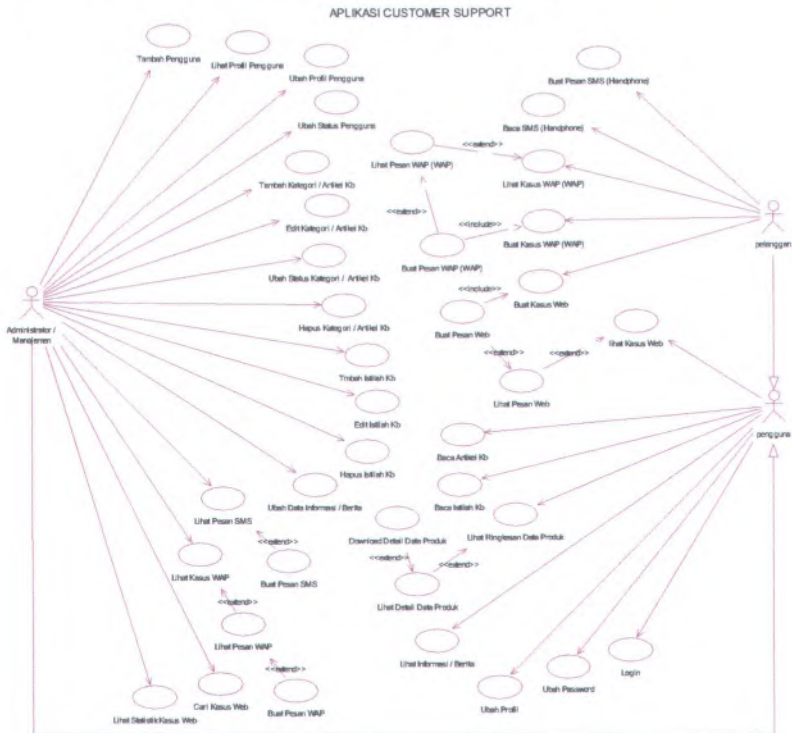
3.4.1.1 Pengguna

Aktor pengguna merupakan generalisasi dari semua aktor yang lain karena semua aktor bawahan dari aktor pengguna bisa melakukan login, mengubah profil, melihat informasi atau berita, melihat informasi dan download dokumen detail data produk tertentu dan mengubah password.

3.4.1.2 Administrator / Manajemen

Aktor ini mempunyai otorisasi untuk mengelola data pengguna. Selain itu aktor ini juga memiliki otoritas untuk mengelola semua data pesan yang masuk pada sistem melalui berbagai *touchpoint*

aplikasi yang digunakan dan mengelola berbagai informasi yang akan ditampilkan pada aplikasi web *Customer Support*.



Gambar 3.1 Use Case Diagram sistem Aplikasi

3.4.1.3 Pelanggan

Aktor ini memiliki otoritas untuk mengelola hanya data pesan yang dimilikinya sendiri. Pelanggan dapat membuat atau melihat pesan.



3.4.2 Deskripsi Use Case

Deskripsi use case adalah penjelasan dari spesifikasi-spesifikasi use case yang ada di aplikasi *Customer Support* berdasarkan use case diagram sistem aplikasi.

Deskripsi use case aplikasi *Customer Support* dijelaskan dalam bentuk tabel naratif use case yang terdapat pada lampiran A.

3.5 Sequence Diagram

Sequence diagram adalah penjelasan proses yang terjadi pada sistem dalam mengolah objek-objek pada sisi *back-end*. Perancangan sequence diagram mengacu pada diagram usecase yang telah didefinisikan sebelumnya.

Sequence diagram aplikasi *Customer Support* terdapat pada lampiran B.

3.6 Class Diagram

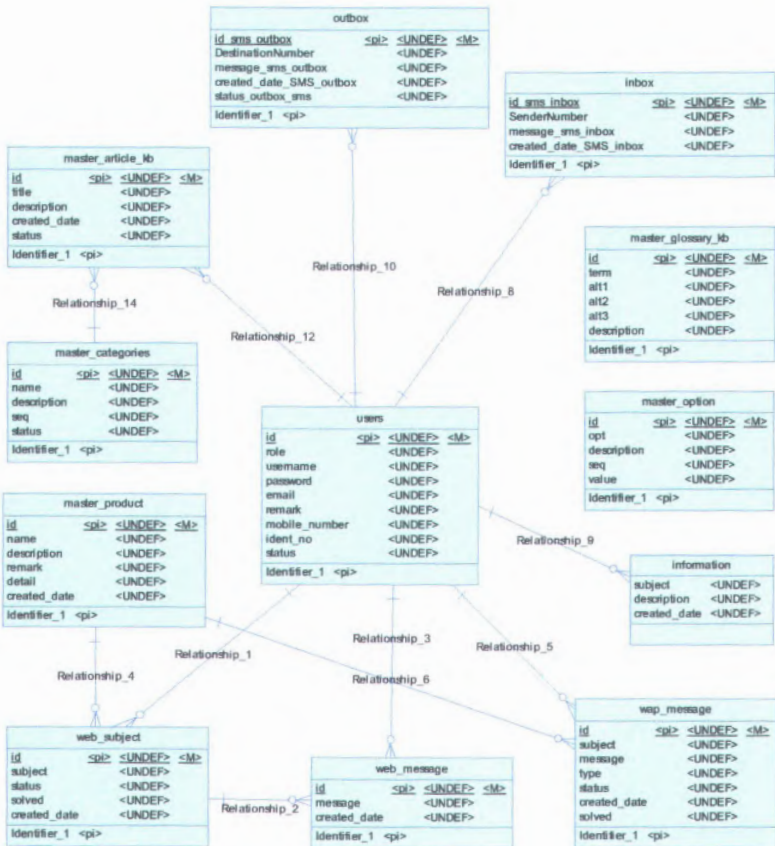
Class diagram adalah penggambaran kelas-kelas *Java* yang digunakan pada sisi *back-end* sistem aplikasi *Customer Support*.

Class diagram aplikasi *Customer Support* terdapat pada lampiran C.

3.7 Perancangan Basis Data

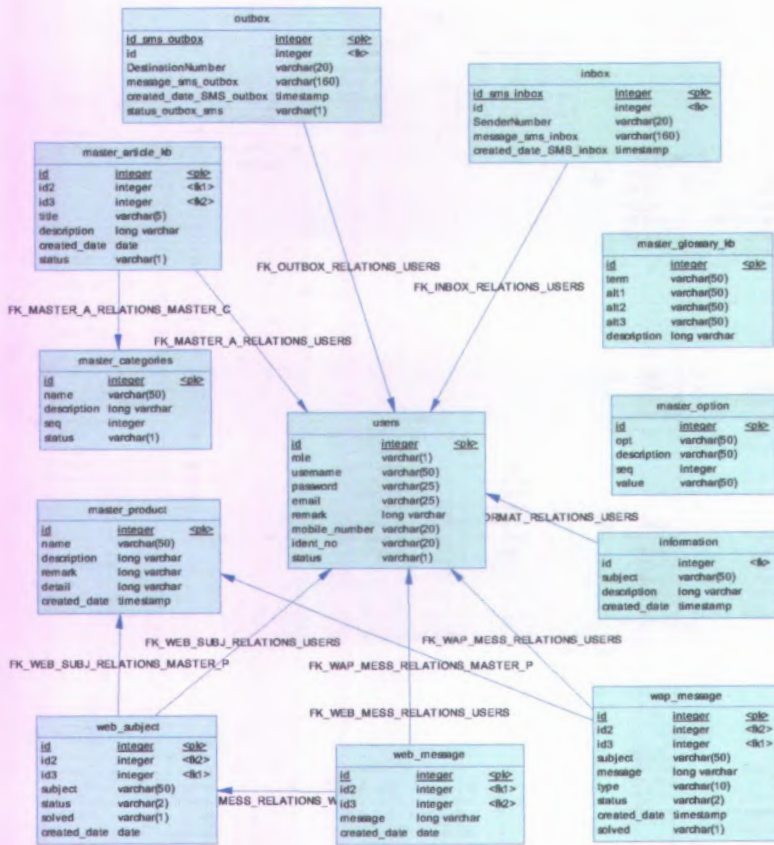
Pada bagian ini akan dijabarkan perancangan basis data aplikasi *Customer Support*. Perancangan basis data dilakukan dengan membuat dua jenis diagram/model yaitu Model Data Konseptual dan Model Data Fisik.

Untuk Model Data Konseptual dari aplikasi dapat dilihat pada gambar 3.2.



Gambar 3.2 Model Data Konseptual aplikasi Customer Support

Sedangkan untuk Model Data Fisik dapat dilihat pada gambar 3.3. Model Data Fisik menjelaskan bagaimana entitas-entitas bisnis di perusahaan direalisasikan ke dalam basis data.



Gambar 3.3 Model Data Fisik aplikasi Customer Support

Tabel-tabel yang digunakan pada basis data aplikasi ini adalah:

- users
Digunakan untuk menyimpan data pengguna.
- web_subject
Digunakan untuk menyimpan data subjek pesan web.

- web_message
Digunakan untuk menyimpan data pesan web berdasarkan subjek tertentu.
- wap_message
Digunakan untuk menyimpan data subjek dan pesan wap.
- master_product
Digunakan untuk menyimpan data produk.
- master_categories_kb
Digunakan untuk menyimpan data kategori artikel.
- master_article_kb
Digunakan untuk menyimpan data artikel.
- master_glossary_kb
Digunakan untuk menyimpan data istilah.
- master_option
Digunakan untuk menyimpan data pilihan pada comboboxes.
- information
Digunakan untuk menyimpan data informasi / berita.

Untuk table inbox dan outbox SMS menggunakan table hasil generate dari bawaan aplikasi GAMMU, dimana penulis tidak membangun model data fisik melainkan hanya konsep melalui model data fisik tabel yang disediakan aplikasi GAMMU.

3.8 Desain Antarmuka Aplikasi

Desain antarmuka dibagi menjadi dua bagian berdasarkan aplikasi platform aplikasi yang digunakan yaitu, aplikasi web dan wap.

3.8.1 Desain antarmuka aplikasi web Customer Support

Gambar 3.4 adalah desain antar muka aplikasi web *Customer Support*. Struktur halaman yang paling tinggi ditempati oleh halaman login karena semua pengguna harus melakukan login untuk menggunakan aplikasi ini. Ada 4 halaman / form yang bisa diakses oleh semua pengguna yaitu form lihat informasi / berita, form lihat data produk, form ubah profil dan form ubah password. Selanjutnya halaman terbagi ke dalam dua kelompok sesuai dengan otoritasnya masing-masing.

- **Halaman Administrator**

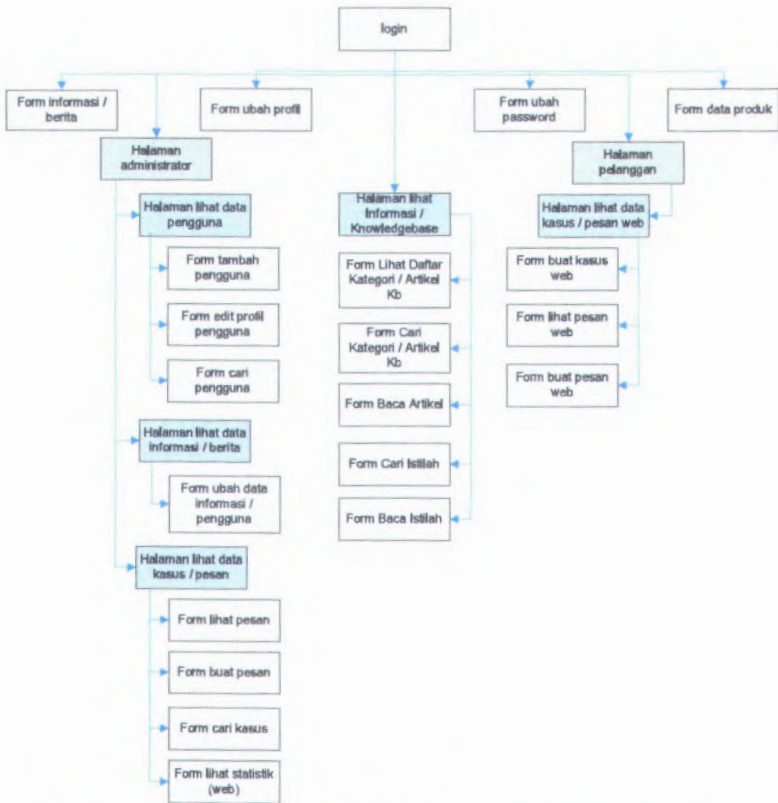
Halaman-halaman administrator digunakan untuk mengelola data pengguna, informasi dalam bentuk artikel maupun berita, dan manajemen pesan.

- **Halaman Pelanggan**

Halaman yang disediakan bagi pelanggan adalah form ubah data profil, ubah password serta manajemen pesan.

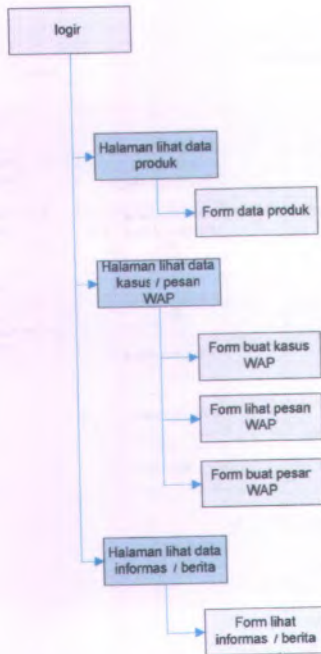
3.8.2 Desain antarmuka aplikasi WAP Customer Support

Gambar 3.5 adalah desain antar muka aplikasi WAP *Customer Support*. Struktur halaman yang paling tinggi ditempati oleh halaman login karena semua pengguna harus melakukan login untuk menggunakan aplikasi ini. Aplikasi ini diperuntukkan bagi pelanggan yang mengakses sistem melalui protokol WAP. Sehingga desain antarmuka disesuaikan desain fitur sebelumnya.



Gambar 3.4 Desain antarmuka aplikasi Web Customer Support

Untuk aplikasi WAP *Customer Support* antarmuka aplikasi hanya menyediakan halaman untuk mengelola data pesan serta menampilkan informasi ringkasan data produk.



Gambar 3.5 Desain antarmuka aplikasi WAP Customer Support

BAB IV IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini dijelaskan mengenai pembuatan aplikasi berdasarkan rancangan sistem yang telah dibuat. Penjelasan implementasi proses dilakukan dengan menggunakan kode sumber aplikasi. Tahap-tahap implementasi meliputi implementasi proses, basis data, antarmuka.

4.1 Lingkungan Implementasi

Sistem berbasis web ini dikembangkan pada suatu perangkat keras *Personal Computer Desktop* (PC). Spesifikasi lingkungan implementasi perangkat keras yang dilakukan pada pengembangan sistem aplikasi dapat dilihat pada tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat keras untuk implementasi sistem

Aplikasi	Perangkat Keras	Spesifikasi
Web - WAP	<i>Personal Computer</i> (Desktop)	Prosesor : Intel (R) Pentium (R) Dual CPU E2160 @ 1.80 GHz
		Memori : 1016 MB DDRAM
SMS	Modem	Sony Erricson w700i

Selain itu untuk implementasi pada lingkungan perangkat lunak, sistem berbasis web ini dikembangkan dengan menggunakan teknologi *Java Development Kit* 1.6. Tool utama yang digunakan adalah *Netbeans* 5.5. *Webserver* yang digunakan adalah *Apache Tomcat* 5.5.20. Sistem basis data yang digunakan adalah *MySQL*. Untuk koneksi basisdata, menggunakan *JDBC*. Antarmuka yang digunakan untuk berhubungan dengan pengguna adalah halaman-

halaman *Java Server Pages*. Tabel 4.1 berikut ini merangkum teknologi-teknologi yang digunakan untuk implementasi.

Tabel 4.2 Teknologi-teknologi yang Digunakan untuk Implementasi Sistem

Aplikasi	Teknologi	Versi
Web - WAP	JDK	JDK 1.6
	Editor	Netbeans 5.5
	Web Server	Apache Tomcat 5.5.20
	<i>JavaScript</i> Library	Ext 2.0
	Sistem Basis Data	MySQL
	Bahasa Pemrograman	<i>Java, Java Server Pages</i>
SMS	SMS Gateway	Gammu (GNU All Mobile Management Utilities) win 32
	Sistem Basis Data	MySQL

4.2 Implementasi *JavaScript* Library Ext

Dalam proses implementasi, *JavaScript library* Ext memegang peranan dalam beberapa proses. Seperti yang telah diuraikan di bab dua, bahwa Ext memiliki beberapa kontrol atau widget yang dapat digunakan untuk mengembangkan aplikasi berbasis web. Kontrol utama dari Ext yang digunakan dalam aplikasi web *Customer Support* ini adalah Grid Panel. Selain itu penulis juga menggunakan kontrol yang lain, seperti kontrol Tree, Tab Panel, Toolbar, Window, Comboboxes, dan beberapa kontrol yang lain.

Hal pertama yang dilakukan adalah meletakkan library Ext 2.0 pada lokasi tertentu di project aplikasi web. Pada dasarnya penggunaan library Ext sama dengan penggunaan script *JavaScript* yang diletakkan pada file tertentu. Namun, pada library Ext pemrogram hanya tinggal memanggil kontrol-kontrol yang sudah disediakan oleh library tersebut.

Umumnya penggunaan library Ext dalam pembuatan suatu website yang dinamis dikombinasikan dengan bahasa pemrograman php. Namun, pada tugas akhir ini penulis menggunakan bahasa pemrograman *Java*.

Untuk menampilkan halaman aplikasi pemrogram menggunakan halaman *Java Server Pages* sebagai tempat *JavaScript* Ext diletakkan.

Berikut penjelasan tentang penggunaan library Ext pada halaman *Java Server Pages*.

Membuat halaman web menggunakan halaman *Java Server Pages*, contoh .jsp. kode programnya seperti pada gambar 4.1.

```

1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
4     <title>Contoh Penggunaan Ext</title>
5
6     <!-- library dasar Ext -->
7     <link rel="stylesheet" type="text/css" href="js/ext-2.0/resources/css/ext-all.css" />
8     <script type="text/javascript" src="js/ext-2.0/adapters/ext/ext-base.js"></script>
9     <script type="text/javascript" src="js/ext-2.0/ext-all.js"></script>
10    <!-- end library -->
11
12    <!-- source kontrol Ext yang dibuat -->
13    <script type="text/javascript" src="js/web/contoh.js"></script>
14
15  </head>
16  <body>
17    <h1>Hello Ext ...!!</h1>
18    <p align="center">Ini contoh implementasi penggunaan library Ext.</p>
19    <p align="center">
20      <table>
21        <tr><td><div id="bee"></div></td></tr>
22      </table>
23    </p>
24  </body>
25 </html>

```

Gambar 4.1 Kode program halaman contoh.jsp

Keterangan :

Baris 7 – 9 merupakan lokasi tempat library Ext yang harus ada. Terdapat 3 file *JavaScript* yang difungsikan sebagai sumber *library* agar kontrol / *widget* Ext yang dibangun dapat berjalan.

Baris 12 merupakan lokasi file *JavaScript* dimana kontrol / *widget* yang dibangun oleh pemrogram.

Baris 20 merupakan proses pemanggilan kontrol / *widget* yang dibangun. Dasarnya tiap *widget* pada Ext mempunyai id dan id

tersebut dipanggil menggunakan tag `div` agar dapat tampil pada halaman web.

Kemudian membuat file *JavaScript* `contoh.js` dimana tempat kode program membuat widget yang akan dibuat. Pada `contoh.js` widget yang dibuat adalah tipe `Grid Panel`. Untuk membuat suatu kumpulan data statis maka dibuat suatu variabel yang menampung bentuk data `array`, seperti pada gambar 4.2.

```

1 Ext.onReady(function(){
2
3     Ext.state.Manager.setProvider(new Ext.state.CookieProvider());
4
5     var myData = [
6         ['Duncow',71.72,-0.02,-0.03,'9/1 12:00am'],
7         ['Wings',29.01,0.42,1.47,'9/1 12:00am'],
8         ['Danone',35.57,-0.39,1.11,'9/1 12:00am'],
9         ['Sony Ericsson',34.55,-0.94,1.22,'9/1 12:00am'],
10        ['Nokia',12.336,0.48,1.22,'9/1 12:00am'],
11        ['Sido Muncul',45.45,0.73,-1.63,'9/1 12:00am']
12    ];
13
14    // contoh fungsi untuk mengubah tampilan warna baris pada kolom perubahan
15    function change(val){
16        if(val > 0){
17            return '<span style="color:green;">' + val + '</span>';
18        }else if(val < 0){
19            return '<span style="color:red;">' + val + '</span>';
20        }
21        return val;
22    }
23

```

Gambar 4.2 Kode program membuat data statis pada file `contoh.js`

Kemudian, pada gambar 4.3 dibuat store yang merupakan kontrol yang berfungsi menampung data yang sebelumnya telah ditentukan.

```

34 // membuat data simple yang statis
35 var store = new Ext.data.SimpleStore({
36     fields: [
37         {name: 'perusahaan'},
38         {name: 'harga', type: 'float'},
39         {name: 'perubahan', type: 'float'},
40         {name: 'persen_perubahan', type: 'float'},
41         {name: 'lastChange', type: 'date', dateFormat: 'n/j h:ia'}
42     ]
43 });
44 store.loadData(myData);

```

Gambar 4.3 Kode program membuat store untuk menampung data pada file contoh.js

Pada gambar 4.4 setelah suatu data beserta store dibuat maka selanjutnya membuat Grid Panel dimana store yang digunakan merupakan store yang sebelumnya telah dibuat.

Keterangan, kode program pada gambar 4.4 :

Baris 5 - 12 merupakan deklarasi variable *JavaScript* berbentuk array berupa data statis yang akan diload oleh fungsi `Ext.data.SimpleStore` dan ditampilkan pada Grid Panel.

Baris 15 - 22 merupakan fungsi untuk memanipulasi tampilan dari baris berdasar kolom tertentu.

Baris 35 - 44 merupakan fungsi untuk menyiapkan data yang akan diload pada Grid Panel dimana data statis diambil pada variabel yang sebelumnya telah dideklarasikan.

Baris 47 - 84 merupakan fungsi untuk menampilkan kontrol / *widget* dengan tipe Grid Panel.

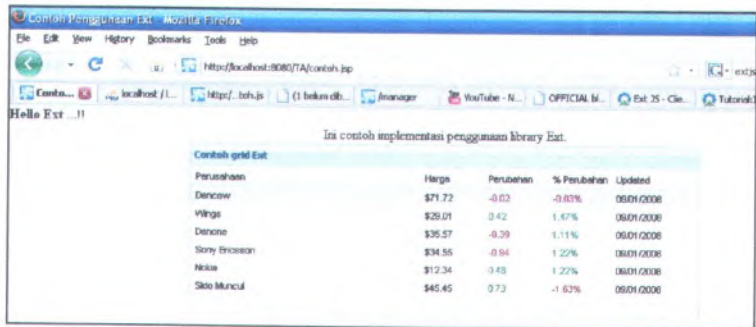
```

46 // membuat grid
47 var grid = new Ext.grid.GridPanel({
48     store: store,
49     columns: [
50         {
51             id: 'perusahaan',
52             header: "Perusahaan",
53             width: 160,
54             sortable: true,
55             dataIndex: 'perusahaan'
56         },
57         {
58             header: "Harga",
59             width: 75,
60             sortable: true,
61             renderer: 'usMoney',
62             dataIndex: 'harga'
63         },
64         ...
65         ...
66         ...
67         {
68             header: "Perubahan",
69             width: 75, sortable: true,
70             renderer: change,
71             dataIndex: 'perubahan'
72         }
73     ],
74     stripeRows: true,
75     autoExpandColumn: 'perusahaan',
76     height: 200,
77     width: 600,
78     title: 'Contoh grid Ext'
79 });
80
81 grid.render('tes');
82
83 grid.getSelectionModel().selectFirstRow();
84 });

```

Gambar 4.4 Kode program membuat Grid Panel pada file contoh.js

Berikut hasil tampilan dari halaman contoh.jsp :



Gambar 4.5 Halaman contoh.jsp

Pada file contoh.js diatas dapat dilihat bahwa kode program yang dibuat merupakan konsep dasar dari kode program *JavaScript*. Tiap Kontrol / *widget* memiliki tipe atribut atau properti tertentu berdasarkan jenis fungsinya. Untuk tipe atribut dapat dilihat dan dipelajari pada API Doc dari ExtJs. Selain itu kontrol / *widget* merupakan sebuah fungsi dimana tipe fungsi-fungsi tersebut telah disimpan pada *library* dan pemrogram tinggal memakai fungsi tersebut untuk ditampilkan.

Selanjutnya akan dijelaskan secara umum beberapa kontrol yang dipakai beserta kode programnya pada aplikasi web *Customer Support*.

4.2.1 Grid Panel

Grid panel hampir digunakan pada setiap menu untuk menampilkan data dari basis data. Dapat dikatakan bahwa Grid Panel telah menggantikan peran tabel seperti di html biasanya. Grid dapat menerima data baik berformat XML maupun berupa format JSON. Namun, dalam tugas akhir ini, penulis menggunakan format JSON

karena kemudahannya dalam hal generate data yang akan diparsingkan. Penulis lebih memilih menggunakan Grid Panel daripada menggunakan tabel biasa karena beberapa kelebihan yang dimiliki Grid, diantaranya:

- Adanya fitur sorting dimana dapat memudahkan dalam pengurutan data yang telah ditampilkan.
- Adanya fitur paging yang memudahkan pemrogram sehingga tidak perlu membuat fungsi paging sendiri (Gambar 4.10)
- Adanya kemudahan dalam tampilan kolom seperti adanya pilihan kolom mana saja yang ditampilkan, susunan urutan kolom dan *resize* ukuran kolom.

Itulah kelebihan utama Grid Panel yang dimiliki oleh Ext. Berdasarkan kelebihan itulah penulis lebih memilih menggunakan Grid Panel tersebut daripada tabel biasa. Untuk membuat Grid Panel, urutan prosesnya adalah mengambil data dari basis data, kemudian mengubah data tersebut kedalam format JSON. Seperti terlihat pada gambar 4.9. Setelah itu baru objek Grid Panel dibuat dengan mengisikan konfigurasi-konfigurasinya, termasuk data yang ingin ditampilkan dalam Grid Panel seperti pada gambar 4.10.

```

67
68 var webData = new Ext.data.Store({
69     proxy: new Ext.data.ScriptTagProxy({
70         url: '/TA/data/web/webdata.jsp', method: 'GET'
71     }),
72     baseParams: {type:0},
73     reader: new Ext.data.JsonReader({
74         root: 'rows',
75         totalProperty: 'results',
76         id: 'web_subject_id',
77         fields: [
78             {name: 'web_subject_id', mapping: 'web_subject_id'},
79             {name: 'web_subject', mapping: 'web_subject'},
80             {name: 'web_product_id', mapping: 'web_product_id'},
81             {name: 'web_created_user', mapping: 'web_created_user'},
82             {name: 'web_created_date', mapping: 'web_created_date'},
83             {name: 'web_status', mapping: 'web_status'},
84             {name: 'web_solved', mapping: 'web_solved'}
85         ]
86     }),
87     remoteSort: false
88 });
89

```

Gambar 4.6 Kode program membuat Data Store


```

<? page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="utf-8" >
<?page isport="lib.web">

<?
String msg = "";
String callback = request.getParameter("callback");
String id = request.getParameter("id");
int tipe = Integer.parseInt(request.getParameter("tipe"));
String start = request.getParameter("start");
String limit = request.getParameter("limit");
String id_login = request.getParameter("id_login");

if (tipe==0) {
    web showListActiveWeb = new web();
    msg = showListActiveWeb.showListActiveWebCad(callback,start,limit);
    showListActiveWeb = null;
} else if (tipe==1) {
    web showDetailWeb = new web();
    msg = showDetailWeb.showDetailWebCad(callback,id,start,limit);
    showDetailWeb = null;
} else if (tipe==2) {
    web showListNonActiveWeb = new web();
    msg = showListNonActiveWeb.showListNonActiveWebCad(callback,start,limit);
    showListNonActiveWeb = null;
} else if (tipe==3) {
    web showFormReplyWeb = new web();
    msg = showFormReplyWeb.showFormReplyWebCad(id,id_login);
    showFormReplyWeb = null;
}
    out.println(msg);
}

```

Gambar 4.7 Kode program halaman Java Server Pages untuk meletakkan hasil Query pada halaman Back-End

```

117 msg+=callback+("{ \"results\": "+total+", \"rows\": [";
118
119 sql = "select " +
120     "from web_subject " +
121     "where status='A' " +
122     "order by created_date limit "+start+", "+limit;
123
124 pg.execSQL(sql);
125 count = pg.count();
126 if (count>0) {
127     pg.first();
128     while (!pg.eof()) {
129         username = pgloop.getSingleResult("select username " +
130             "from users " +
131             "where id="+pg.getString("created_user")
132             +",username");
133
134         msg+="{\"web_subject_id\":\":" +pg.getString("id")+ "\",";
135         msg+="{\"web_subject\":\":" +pg.Q(pg.getString("subject"))+" \",";
136         msg+="{\"web_product_id\":\":" +pg.getString("product_id")+ "\",";
137         msg+="{\"web_created_user\":\":" +username+" \",";
138         msg+="{\"web_created_date\":\":" +pg.getString("created_date")+ "\",";
139         msg+="{\"web_status\":\":" +pg.getString("status")+ "\",";
140         if (itung!=count) msg+="{\"web_solved\":\":" +pg.getString("solved")+ "\",";
141         else msg+="{\"web_solved\":\":" +pg.getString("solved")+ "\",";
142         itung = itung + 1;
143         pg.next();
144     }
145 }
146 msg+="] }";

```

Gambar 4.8 Kode program file Java pada Back-End untuk membentuk String dengan format JSON

```

stcCallback1002({ "results": 2, "rows": [
  (
    "web_subject_id": "1",
    "web_subject": "bug",
    "web_product_id": "7",
    "web_created_user": "faizal",
    "web_created_date": "2008-07-05",
    "web_status": "A",
    "web_solved": "t"
  ),
  (
    "web_subject_id": "2",
    "web_subject": "tes lagi",
    "web_product_id": "3",
    "web_created_user": "faizal",
    "web_created_date": "2008-07-05",
    "web_status": "A",
    "web_solved": "t"
  )
] });

```

Gambar 4.9 Contoh hasil pengembalian data String JSON

```

165 var webActiveGrid = new Ext.grid.GridPanel({
166     store: webData,
167     columns: [
168         {id: 'web_subject_id', header: "ID Kasus", width: 60, sort
169         {header: "ID Produk", width: 60, sortable: true, dataIndex:
170         {header: "Subjek", width: 230, sortable: true, dataIndex:
171         {header: "Oleh", width: 100, sortable: true, dataIndex:
172         {header: "Tanggal", width: 90, sortable: true, dataIndex:
173         {header: "Solved", width: 45, sortable: true, renderer: w
174     ],
175     stripeRows: true,
176     sm: smWebActive,
177     ...
178     ...
179     ...
180     bbar: new Ext.PagingToolbar({
181         pageSize: 10,
182         store: webData,
183         ...
184         ...
185         displayInfo: true,
186         displayMsg: 'Menampilkan kasus {0} - {1} dari {2}',
187         emptyMsg: "Tidak ada kasus",
188         items: [
189             '-', {
190                 text: 'Buka kasus',
191                 enableToggle: true,
192                 toggleHandler: openWebDetails

```

Gambar 4.10 Kode program membuat Grid Panel dengan paging

4.2.2 Tree

Tree dapat digunakan sebagai daftar menu yang ada di sistem, seperti halnya pada *windows explorer*. Penulis menggunakan Tree untuk mengatur daftar menu yang ada pada sistem dimana Tree tersebut dapat menampilkan suatu submenu berdasarkan hieraki pengelompokannya

Untuk isi dari menu Tree penulis menggunakan file dengan extensi tersendiri yang disediakan oleh Ext. File dengan extensi *.json tersebut seperti halnya dijelaskan pada bab sebelumnya merupakan format data JSON yang dapat secara otomatis dibaca oleh TreeLoader. Gambar 4.11 adalah bagaimana kode program untuk membuat Tree. Dan gambar 4.12 merupakan data menu Tree menggunakan file dengan format data JSON.

```

161     var treePanelAdministration = new Ext.tree.TreePanel({
162         id: 'treeAdmin-panel',
163         split: true,
164         height: 600,
165         minSize: 150,
166         autoScroll: true,
167         draggable: false,
168         // konfigurasi khusus
169         rootVisible: false,
170         lines: false,
171         singleExpand: true,
172         useArrows: true,
173         loader: new Ext.tree.TreeLoader({
174             dataUrl: 'treeMenuAdmin/treeAdmin-data.json'
175         }),
176         root: new Ext.tree.AsyncTreeNode()
177     });

```

Gambar 4.11 Kode program untuk membuat Tree

```

1  [{
2      text:'Administrator',
3      //expanded: true,
4      children:[{
5          text:'Edit Profil',
6          id:'profile',
7          leaf:true
8      }},{
9          text:'Ganti Password',
10         id:'password',
11         leaf:true
12     }]
13 },{
14     text:'Pengguna',
15     children:[{
16         text:'Tambah Pengguna',
17         id:'create',
18         leaf:true
19     }],{

```

Gambar 4.12 Kode program JSON untuk membuat data menu untuk Tree

4.2.3 Toolbar

Pada Ext versi 1.1 Toolbar merupakan kontrol terpisah yang harus didefinisikan secara tersendiri, namun pada Ext 2.0 toolbar dapat secara langsung digabungkan dengan Grid, Window maupun Panel. Toolbar dapat diletakkan pada atas maupun bawah suatu kontrol lain (Grid Panel, Window ataupun Panel) yang menggunakannya. Penulis menggunakan Toolbar untuk meletakkan berbagai tombol *action* tertentu. Gambar 4.13 adalah bagaimana kode program untuk membuat toolbar yang muncul pada atas maupun bawah dari suatu Grid Panel.

```

468         closeAction: 'hide',
469         plain: false,
470         tbar: [
471             '<->', logo, '->'
472         ],
473         bbar: [
474             kirimWEB, '->', resetWEB, '<->', tutupWEB
475         ],
476         items: formReplyWeb
477     });
478 }

```

Gambar 4.13 Kode program membuat Toolbar

4.2.4 Accordion

Sama halnya dengan Toolbar, kontrol / *widget* accordion merupakan kontrol yang sudah tergabung pada tipe kontrol lain, yaitu panel. Dengan mendeklarasikan tipe layout dari panel yaitu dengan tipe 'accordion' maka tampilan accordion-pun akan dapat dihasilkan. Gambar 4.14 merupakan kode program untuk membuat panel dengan tipe layout Accordion.

4.2.5 Window

Pada tugas akhir ini selain Grid Panel, kontrol / *widget* yang sering digunakan adalah window. Window pada Ext layaknya sebuah window pada aplikasi desktop dimana pada window dapat digabungkan dengan fungsi lain di dalamnya. Kelebihan yang dapat diberikan pada window Ext ini adalah adanya atribut atau properti yang berfungsi untuk membuat window yang sedang dibuka menjadi fokus utama. Sehingga pengguna tidak dapat mengakses bagian belakang dari halaman aplikasi dibawah window selama window tersebut belum ditutup. Kelebihan dari fungsi tersebut tidak dapat dibuat pada web konvensional. Gambar 4.15 menunjukkan kode program untuk membuat kontrol window pada Ext.

```

354         width: 200,
355         minSize: 200,
356         maxSize: 200,
357         collapsible: true,
358         collapsed: true,
359         margins: '0 0 0 5',
360         border: false,
361         bodyStyle: {background: '#DFE8F7'}},
362         layout: 'accordion',
363         layoutConfig: {
364             animate: true
365         },
366         items: [{
367             title: ':: Administrasi ::',
368             iconCls: 'settings',
369             collapsed: true,
370             items: [treePanelAdministration]
371         }, {
372             title: ':: Manajemen WEB ::',
373             iconCls: 'nav',
374             collapsed: true,
375             items: [treePanelWeb]
376         }, {
377             title: ':: Manajemen WAP ::',

```

Gambar 4.14 Kode program membuat menu Accordion

```

560     var winChartProduk = new Ext.Window({
561         layout: 'fit',
562         title: 'Lihat Chart',
563         iconCls: 'chartpie',
564         width: 450,
565         modal: true,
566         autoScroll: true,
567         height: 500,
568         resizable: false,
569         bodyStyle: 'padding: 5px;',
570         closeAction: 'hide',
571         plain: true,
572         tbar: [
573             '-', '>', logo
574         ],
575         bbar: [
576             '-', '>', {
577                 text: 'Tutup',
578                 enableToggle: true,
579                 toggleHandler: function() {winChartProduk.hide();}
580             }
581         ],
582         items: [panelChartProduk]
583     });

```

Gambar 4.15 Kode program membuat Window

4.2.6 Viewport

Kontrol Viewport adalah sebuah kontrol yang dipakai untuk membuat layout yang standar pada suatu tampilan situs web. Dimana menu-menu telah dibagi kedalam bagian-bagian atau *region* sehingga pemrogram hanya perlu menentukan isi dari bagian tersebut. Gambar 4.16 menunjukkan kode program untuk membuat suatu layout menggunakan Viewport pada Ext.

```

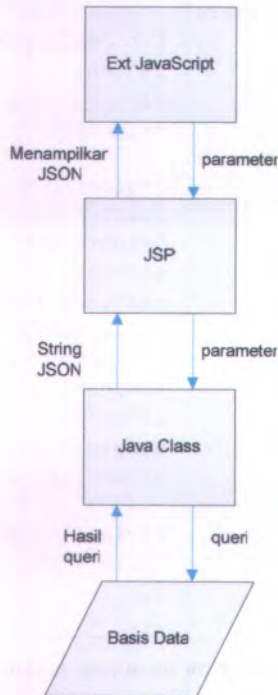
314     var viewport = new Ext.Viewport({
315         layout: 'border',
316         items: [
317             new Ext.BoxComponent({ // row
318                 region: 'north',
319                 el: 'north',
320                 height: 75
321             }), {
322                 region: 'south',
323                 split: true,
324                 height: 100,
325                 minSize: 100,
326                 maxSize: 100,
327                 collapsible: true,
328                 collapsed: false,
329                 title: 'Informasi / Berita',
330                 iconCls: 'info',
331                 margins: '0 0 0 0',
332                 items: [informationPanel]
333             }, {
334                 region: 'east',
335                 title: 'Produk',
336                 ...
337                 ...

```

Gambar 4.16 Kode program membuat layout web menggunakan Viewport

4.3 Implementasi Teknologi Java

Teknologi *Java* yang digunakan dalam pengembangan aplikasi ini dibagi menjadi dua bagian yaitu, *Java Server Pages* serta kelas *Java*. Halaman *Java Server Pages* di fungsikan sebagai halaman web back-end untuk mengambil parameter yang di-parsingkan melalui HTML headers melalui berbagai method GET maupun POST. Sedangkan kelas *Java* digunakan sebagai implementasi dari proses query pada basis data berdasarkan parameter dari *Java Server Pages*. Serta proses membuat string dengan format JSON. String dengan format JSON yang dibuat pada kelas tersebut akan dikembalikan pada halaman *Java Server Pages* dan akan didecode oleh JSON decoder yang ada pada Ext.



Gambar 4.17 Gambaran proses yang terjadi

Gambar 4.17 merupakan gambaran dari proses yang terjadi. File JSP (Gambar 4.19) difungsikan sebagai halaman yang menghasilkan data hasil format JSON yang berasal dari hasil query pada suatu kelas *Java* (Gambar 4.18).

```

10 package lib;
11 import lib.pgint;
12
13
14 public class user {
15
16     /** Creates a new instance of user */
17     public user() {
18     }
19
20     private pgint pg;
21     private String msg;
22
23     public String activatingUserCmd(String id) {
24         pgint pg = new pgint();
25         pg.initStaticConnection();
26         msg = "";
27         String status = "";
28
29         pg.execSQL("select status from users where id="+id);
30         pg.first();
31         status = pg.getString("status");
32         if (status.equals("U")) {
33             pg.execSQL("update users set status='A' where id="+id);
34             msg+="{ success: true }";
35         } else {
36             msg+="{ success: false," +
37                 " errors: { reason: 'Status pengguna adalah Aktif.' } }";
38         }
39         pg.close();
40         pg.closeStaticConnection();
41         return msg;
42     }

```

Gambar 4.18 Kelas Java

4.4 Implementasi Teknologi WAP

Aplikasi WAP *Customer Support* yang dibangun menerapkan teknologi yang sederhana. Penggunaan tipe dokumen XHTML MP merupakan teknologi standar pada teknologi WAP 2.0. Bahasa yang digunakan untuk membuat halaman WAP adalah *Java Server Pages* dimana tipe dokumen yang dideklarasikan adalah tipe XHTML MP.

Dengan menggunakan XHTML MP penulisan pada baris program pada aplikasi yang dibangun akan menjadi lebih ketat (*strict*). Untuk deklarasi tipe dokumen pada halaman aplikasi WAP dapat dilihat pada gambar 4.20.

```

1 <%@ page language="java" contentType="text/html; charset=ISO-885
2 <%@page import="lib.user"%>
3
4 <%
5     String msg = "";
6     String id = request.getParameter("id");
7     user activatingUser = new user();
8
9     msg = activatingUser.activatingUserCmd(id);
10    out.println(msg);
11    activatingUser = null;
12 %>

```

Gambar 4.19 Kode program halaman Java Server Pages menggunakan Method pada Kelas Java

```

1 <?xml version="1.0"?>
2 <!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
3 "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
4 <%@page import="lib.pgint"%>
5 <% response.setContentType("application/vnd.wap.xhtml+xml"); %>
6 <html xmlns="http://www.w3.org/1999/xhtml">
7     <head>
8         <title>WAP CSS - Detail</title>
9         <meta name="author" content="ronny"/>
10    </head>
11    <body>
12        <table cellpadding="1" cellspacing="1" width="100%">
13            <tr><td colspan="2" style="font-size:x-small;font-family:
14                Produk</b></td></tr>
15    <%
16        pgint pg= new pgint();
17        pg.initStaticConnection();
18        if (session.getAttribute("username")!=null) {

```

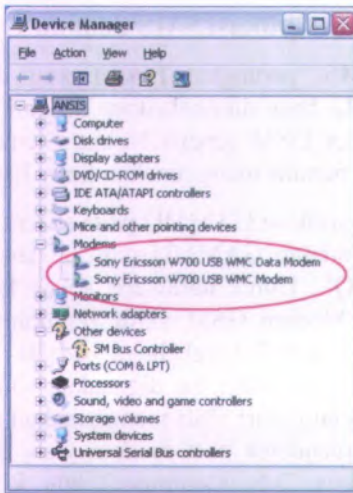
Gambar 4.20 Deklarasi tipe dokumen XHTML MP pada halaman Java Server Pages

4.5 Implementasi Aplikasi SMS dengan GAMMU

Untuk aplikasi SMS, perangkat lunak yang digunakan adalah GAMMU. GAMMU bisa dikoneksikan dengan berbagai macam merek telepon seluler GSM seperti Nokia, Siemens dan lain-lain. Pada tugas akhir ini penulis menggunakan Sony Ericsson W 700i.

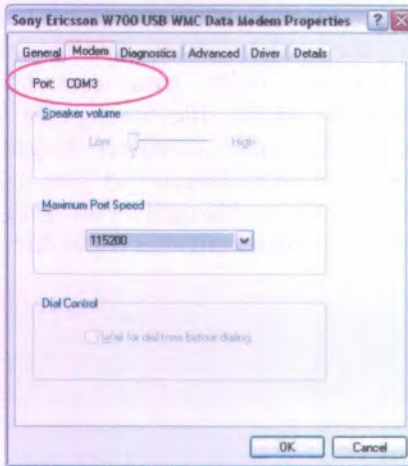
Untuk menjalankan aplikasi GAMMU pada sistem operasi Windows versi yang dipakai adalah GAMMU win 32 dan untuk basis data menggunakan MySQL. Untuk hardware, selain komputer tentunya perlu dipersiapkan Modem GSM yaitu Handphone dengan kabel data port USB. Langkah-langkahnya adalah menghubungkan modem atau dalam tugas akhir ini digunakan sony ericsson w700i dengan komputer melalui port USB menggunakan kabel data. Proses selanjutnya adalah mendeteksi port modem yang digunakan dengan cara klik kanan pada "My Computer" lalu klik pada "Device Manager" maka, akan tampil window untuk melihat hardware yang terhubung pada komputer (Gambar 4.21). Pada modems pilih sony ericsson w700 USB WMC Data Modem klik dua kali. Untuk melihat port yang digunakan oleh modem (Gambar 4.22).

Setelah itu ekstrak file `gammu_win32.zip` (yang bisa didapat pada www.gammu.org) ke salah satu folder, misalnya pada `c:\gammu_win32`. Dalam folder `gammu_win32` terdapat file `gammurc`, file tersebut adalah file konfigurasi `gammu` untuk melakukan proses koneksi antara komputer dengan modem, baris port dan connection disesuaikan dengan port yang didefinisikan oleh komputer pada menu *Device Manager*. Port yang diubah untuk disesuaikan dengan konfigurasi komputer dapat dilihat pada gambar 4.23.



Gambar 4.21 Window Device manager

Untuk melihat Port modem yang diinisialisasikan oleh computer dapat dilihat dengan memilih modem dan klik dua kali.



Gambar 4.22 Port modem yang digunakan

```

port = com3:
#model = 6110
connection = at115200
#synchronizetime = yes
#logfile = gammulog
#logformat = textall
#use_locking = yes
#gammuloc = locfile
#startinfo = yes
#gammucoding = utf8
#rsslevel = teststable
#usephonedb = yes

```

Gambar 4.23 Konfigurasi port pada file gammurc

Setelah semua konfigurasi telah dilakukan maka aplikasi dapat dijalankan. GAMMU merupakan aplikasi yang bersifat *service* dimana aplikasi akan mendeteksi *inbox* dari modem yang digunakan. Data SMS yang ada pada Handphone akan dimasukkan pada tabel yang telah ditentukan. Perintah untuk mendetek bahwa gammu telah berhasil terhubung dengan modem adalah dengan perintah `c:\gammu --identify`, sedangkan untuk menjalankan service gammu dengan perintah `c:\gammu --smsd MYSQL smsdrc..`

Sebelum semua proses diatas dilakukan maka terlebih dahulu dilakukan pembuatan tabel. Tabel yang digunakan untuk aplikasi GAMMU adalah tabel tersendiri yang disediakan pada file zip GAMMU yang didapat. Stabel dapat dibangun dengan mengekstrak file sql yang terdapat pada file zip tersebut. Setelah meng-import tabel-tabel GAMMU maka perlu ada penyesuaian terhadap tabel-tabel yang digunakan pada aplikasi sehingga dapat dibangun sebuah basis data aplikasi yang telah terintegrasi dengan tabel-tabel aplikasi *Customer Support*.

```

C:\WINDOWS\system32\cmd.exe

C:\gammu_win32\win32>gammu --identify
[Gammu      - 1.09.00 built 22:03:05 Dec  6 2006 in MS UC .NET 2005]
[Connection - "at115200"]
[Model type  - "com1:"]
[Device      - "Win XP Pro SP2]
[Run on     - "\.\COM4]
Serial device: DTR is up, RTS is up
[Module     - "auto"]
Escaping SMS mode
SENDING frame type 0x00/length 0x02/2          ..
1B 10D
Sending simple AT command to wake up some devices
SENDING frame type 0x00/length 0x03/3          AT.
41A154T10D
1 "AT"
2 "OK"
RECEIVED frame type 0x00/length 0x09/9          AT...OK..
41A154T10D 10D 10A 14F014BR10D 10A
Enabling echo
SENDING frame type 0x00/length 0x05/5          ATE1.
41A154T145E13110D
1 "ATE1"

```

Gambar 4.24 Identifikasi lingkungan sistem GAMMU berjalan

Pada gambar 4.24 dillihatkan proses deteksi modem dan lingkungan implementasi GAMMU berjalan.

Untuk melihat pesan SMS yang masuk pada basis data maka dapat dilihat pada tabel inbox, sedangkan untuk mengirim pesan yang perlu dilakukan hanyalah menginsert satu record data pesan pada tabel outbox.

4.6 Implementasi Proses

Sesuai dengan rancangan yang telah dibuat sebelumnya, maka dalam implementasi proses ini akan dijelaskan beberapa fungsi-fungsi penting yang digunakan. Untuk menyederhanakan penjelasan, kode program akan diganti deskripsi proses singkat melalui comment program.

4.6.1 Proses Koneksi dengan Basis Data

Koneksi database menggunakan JDBC standar. Aktivitas ini ditangani oleh sebuah kelas dengan nama `pgint`. Setiap proses yang membutuhkan koneksi dengan database bisa membuka koneksi

dengan menggunakan kelas `pgint` dan memanggil method-method yang dimiliki oleh kelas `pgint`.

```

boolean connect() {
    try {
        String url = "jdbc:mysql://" + DB_HOST + "/" + DB_NAME;
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url,DB_USR,DB_PWD);
        stat = conn.createStatement();
        return true;
    } catch (ClassNotFoundException e) {
        msg = "Database driver not found !";
        return false;
    } catch (Exception e) {
        msg = "Connection to database failed !";
        return false;
    }
}

```

Gambar 4.25 Proses pembuatan koneksi Basis Data

Gambar 4.25 menunjukkan bagaimana sebuah koneksi basis data dibuka. Sedangkan Gambar 4.26 menunjukkan bagaimana sebuah koneksi basis data ditutup

```

75 public boolean closeStaticConnection() {
76     try {
77         if (stat != null) {
78             stat.close();
79             stat = null;
80         }
81
82         if (conn != null) {
83             conn.close();
84             conn = null;
85             isConnected = false;
86         }
87         return true;
88     } catch (SQLException e) {
89         msg = "Close connection failed !";
90         return false;
91     }
92 }

```

Gambar 4.26 Proses penutupan koneksi Basis Data



4.6.2 Proses Login

Proses login adalah proses untuk meng-autentifikasi pengguna. Semua pengguna yang ingin menggunakan aplikasi ini harus melakukan login terlebih dahulu.

Proses autentifikasi dilakukan oleh kelas `active.java`. Kelas ini mencari data pengguna berdasarkan `username` dan `password` yang dimasukkan di form login. halaman `logon.jsp` berfungsi untuk menerima data masukan dari form login sekaligus melakukan validasi untuk memastikan apakah kolom sudah diisi. Jika proses autentifikasi gagal, baik disebabkan kesalahan `username` atau kesalahan `password`, maka pengguna akan dikembalikan ke halaman login. Gambar 4.27 berikut ini menunjukkan kode program utama untuk melakukan login. Gambar 4.28 menunjukkan halaman login.

```

22 public boolean setLogon(String usr, String pwd) {
23     boolean isSuccess = false;
24     pgint beanPg = new pgint();
25     beanPg.initStaticConnection();
26     try {
27         beanPg.execSQL("select * " +
28             "from users " +
29             "where username='" + usr + "' " +
30             "and password='" + pwd + "'");
31         if (beanPg.count()>0) {
32             beanPg.first();
33             strUserId = beanPg.getString("id");
34             strUserName = beanPg.getString("username");
35             strUserMail = beanPg.getString("email");
36             strPrivilege = beanPg.getString("role");
37             isSuccess = true;
38         }
39         setComplete(true);
40     }
41     catch(Exception e) {
42         e.printStackTrace();
43     }
44     beanPg.close();
45     beanPg.closeStaticConnection();
46     return isSuccess;
47 }

```

Gambar 4.27 Kode Program untuk melakukan proses login



Gambar 4.28 Form Login pada halaman `login.jsp`

4.6.3 Proses Menampilkan Data

Salah satu diantara beberapa proses utama dalam aplikasi ini adalah proses menampilkan data. Hampir semua entitas bisnis menggunakan proses ini untuk menampilkan data. Karena tahapan proses yang dilakukan untuk proses menampilkan data hampir sama, penulis hanya akan menunjukkan salah satu contoh.

Tampilan data yang disajikan diletakkan pada sebuah kontrol Grid. Untuk menampilkan data dalam bentuk Grid, diperlukan sebuah *JavaScript* yang berfungsi untuk mengolah data, membuat Grid Panel, serta menampilkan Grid Panel tersebut di halaman html. Untuk dapat menggunakan Ext, penulis harus me-load sebuah file *JavaScript* utama yang dimiliki Ext dalam halaman html yakni file `ext-all.js`. Setelah itu baru penulis dapat membuat objek-objek yang diperlukan, salah satunya adalah objek kontrol / widget Grid yang akan digunakan untuk menampilkan data. Untuk menampilkan data dalam bentuk Grid, penulis harus membuat sebuah div di halaman html sebagai penampung dari Grid Panel yang akan dibuat.

Setiap file *JavaScript* yang menggunakan library Ext, maka diawal penulisan kode program *JavaScript* akan selalu diawali dengan `Ext.onReady()`. Seperti yang telah dijelaskan sebelumnya. Proses penampilan data dimulai dengan pengambilan data. Gambar 4.29 berikut menunjukkan proses pengambilan data dari basis data.

```

161     var userData = new Ext.data.Store({
162         proxy: new Ext.data.ScriptTagProxy({
163             url: '/TA/data/user/user.jsp', method: 'GET'
164         }),
165         reader: new Ext.data.JsonReader({
166             root: 'rows',
167             totalProperty: 'results',
168             id: 'id',
169             fields: [
170                 {name: 'id', mapping: 'id'},
171                 {name: 'username', mapping: 'username'},
172                 {name: 'email', mapping: 'email'},
173                 {name: 'remark', mapping: 'remark'},
174                 {name: 'mobile_number', mapping: 'mobile_number'},
175                 {name: 'ident_no', mapping: 'ident_no'}
176             ]
177         }),
178         remoteSort: false
179     });

```

Gambar 4.29 Kode program mengambil data dari Basis Data dengan Ext

Dalam kode program diatas, *JavaScript* memanggil file `user.jsp`, dimana fungsi dari file ini adalah mendapatkan data dari basis data dengan memanggil kelas `setting.java` dan proses query terhadap basis data oleh kelas `pgint.java` lalu kelas `setting.java` mengolahnya menjadi sebuah data string dengan format JSON, dan mengembalikan data string tersebut dalam bentuk text hasil pada suatu halaman *Java Server Pages*.

```

506         sql = "select * " +
507             "from users " +
508             "where status='A' and role<0 limit "+start+", "+limit;
509
510         pg.execSQL(sql);

```

Gambar 4.30 Kode program mengambil data yang dilakukan pada file Kelas setting

Untuk mendapatkan bentuk format tampilan sesuai dengan yang penulis inginkan, maka penulis membuat sebuah fungsi untuk mengatur tampilan disetiap kolom dengan memanfaatkan konfigurasi renderer yang ada dalam kolom (Gambar 4.32).

```

616     var gridSearchUser = new Ext.grid.GridPanel({
617         store: userSearchData, // data store
618         columns: [
619             {
620                 id: 'user_id',
621                 header: "ID",
622                 width: 50,
623                 sortable: true,
624                 dataIndex: 'id'
625             },
626             {
627                 header: "No. Identitas",
628                 width: 80, sortable: true,
629                 resizable: false,
630                 dataIndex: 'ident_no'
631             },
632             ...
633             ...
634             {
635                 header: "Aktif",
636                 width: 40,
637                 sortable: true,
638                 renderer: warnaStatus,
639                 dataIndex: 'status'
640             }
641         ],

```

Gambar 4.31 Kode program tampilan kolom pada Grid Panel

```

606     function warnaStatus(val){
607         if (val == 'A') {
608             return '<span style="color:green;"><b>#10004;</b></span>';
609         } else {
610             return '<span style="color:red;"><b>#10006;</b></span>';
611         }
612     }

```

Gambar 4.32 Kode program fungsi Render tampilan kolom Grid Panel

Dengan menampilkan data menggunakan Grid, penulis tidak mengalami banyak masalah ketika melakukan paging halaman, karena fungsi ini sudah disediakan oleh Ext (Gambar 4.34). Untuk menambahkan fungsi paging, maka ditambahkan kode program seperti terlihat dalam gambar 4.33 berikut ini.

```

221     )
222     ],
223     bbar: new Ext.PagingToolbar({
224         pageSize: 10,
225         store: userData,
226         displayInfo: true,
227         displayMsg: 'Menampilkan pengguna {0} - {1} dari {2}',
228         emptyMsg: "Tidak ada pengguna",
229         items: [
230             '-', {
231                 enableToggle: true,
232                 text: 'Preview',
233                 toggleHandler: previewUser
234             }
235         ]
236     });

```

Gambar 4.33 Kode program untuk menambahkan fungsi Paging

Daftar pengguna

Edit Pengguna Non-Aktifkan Pengguna

ID Pengguna	No. Identitas	Username	No. Handphone	E - mail
49	5204100013	rizka	+6285655338021	rizka_zulfetri@yahoo.com
Catatan : ini pengguna rizka				
48	5204100032	segaf	+6285232642001	oyex10@yahoo.com
Catatan : ini pengguna segaf				
47	5204100078	noval	+6285648453677	noval78@yahoo.com
Catatan : ini pengguna noval				
46	5204100033	nandha	+628563451206	nandha@yahoo.com
Catatan : ini pengguna nandha				
41	5204100056	taizal	+623160613665	fmahananto@yahoo.com
Catatan : ini pengguna taizal				

Page 1 of 3 Preview Menampilkan pengguna 1 - 10 dari 21

Gambar 4.34 Tampilan grid dengan paging

4.6.4 Proses Menambahkan Data

Proses penambahan data dilakukan oleh pengguna sesuai dengan fungsinya masing-masing. Karena semua proses penambahan data memiliki mekanisme yang mirip, maka penulis akan menjelaskan proses ini dengan mengambil satu contoh saja.

Penambahan data dilakukan dengan mengisi form yang telah disediakan.

Data yang telah dimasukkan ke dalam form masukan akan divalidasi berdasarkan kolom-kolom yang sifatnya *mandatory*. Validasi ini dilakukan oleh Ext. Pada Kontrol dengan tipe form pemrograman tidak perlu membuat fungsi tersendiri untuk memvalidasi isian dari tiap textfield yang digunakan (Gambar 4.35). Ext telah menyediakan atribut tertentu pada kontrol text field yang mempunyai fungsi untuk validasi terhadap inputan yang akan dimasukkan. Jenis tipe inputan-pun bisa diatur sesuai dengan keinginan, nomer, huruf maupun tipe jenis seperti password atau alamat e-mail.

```

4   var create_ident_no = new Ext.form.NumberField({
5       fieldLabel: 'No. Identitas ',
6       name: 'create_ident_no',
7       allowDecimals:false,
8       allowBlank:false,
9       maxValue:9999999999999999,
10      maxLength:15,
11      maxLengthText: 'Mohon masukkan nomor identitas dengan benar',
12      width:125,
13      emptyText: 'No. ID...',
14      labelSeparator:''
15  });

```

Gambar 4.35 Kode program untuk validasi data masukan

Gambar 4.36 Tampilan pesan kesalahan validasi inputan

Jika data yang dimasukkan sudah valid, maka controller akan memanggil method `updateUserCmd()` untuk menyimpan data tersebut ke dalam basis data. Proses menyimpan data ke database dapat dilihat pada kode program berikut.

```

53   pg.execSQL("INSERT INTO users (" +
54       "role,username,password,email," +
55       "remark,mobile_number,ident_no)" +
56       " VALUES ('1','"+pg.Q(create_username)+"'" +
57       "','"+pg.Q(create_password)+"','"+create_email+"'" +
58       "','"+pg.Q(create_remark)+"','"+create_num_mobile+"'" +
59       "','"+create_ident_no+"')");
60   msg+="{ success: true }";

```

Gambar 4.37 Kode program untuk menambahkan data

4.6.5 Proses Mengubah Data

Untuk melakukan proses pengubahan data, terlebih dahulu sistem mengambil entitas yang datanya akan diubah. Untuk data yang ditampilkan dalam bentuk Grid Panel, proses pengambilan data ini dapat dilakukan dengan menggunakan Ext. Caranya dengan memilih salah satu baris dalam Grid Panel dan menekan tombol edit. Karena tahapan proses ini hampir sama pada tiap entitas bisnis, maka penulis akan mengambil salah satu contoh yaitu mengedit data pelanggan. Diawali dengan menekan tombol edit, maka sistem akan menampilkan kontrol / *widget* berupa window dimana dalam window tersebut terdapat form untuk mengubah data pengguna.

```

240 function editUser(btn, pressed){
241     if (selectedId!=null) {
242         if (!winEditUser) {
243             winEditUser = new Ext.Window({
244                 layout: 'form',
245                 title: 'Edit Pengguna',
246                 iconCls: 'window_edit',
247                 modal:true,
248                 width:505,
249                 autoHeight:true,
250                 resizable:false,
251                 border:false,
252                 closeAction:'hide',
253                 plain:false,
254                 tbar: [
255                     '->',logo,'-'
256                 ],
257                 hbar: [
258                     bbupdateSave,'-',bbresetProUpdateFS,'->',bbtutupEDIT
259                 ],
260                 items: formUser
261             });
262         }
263         formUser.getForm().reset();
264         winEditUser.show();
265         proUpdateFSData.load({params:{id:selectedId}});
266         formUser.getForm().load({waitMsg:'Loading data, Mohon tunggu...'});
267     } else {
268         Ext.Msg.alert('Peringatan','Tidak ada pengguna yang dipilih');
269     }
270 }

```

Gambar 4.38 Kode program untuk menampilkan Window Form Ubah data pengguna

Penjelasan kode program diatas adalah sebuah fungsi dari tombol pada Toolbar dari suatu Grid Panel pengguna. Fungsi akan melakukan validasi apakah sudah benar ada baris yang dipilih. Jika pengguna belum memilih salah satu data, maka sistem akan mengeluarkan pesan peringatan. Jika proses pemilihan data benar, maka sistem akan menampilkan window dan form dimana isi dari form merupakan data store yang diload dan mengambil data pada database melalui kelas java. Setelah pengguna memasukkan data, maka data inputan yang dimasukkan ke dalam form tersebut akan divalidasi dahulu oleh Form Ext sebelum diproses oleh kelas java. Proses ini mirip dengan proses tambah data.

Untuk melakukan proses perubahan data, kelas controller membuat objek kemudian memanggil `doSQL()` milik kelas `pgint.java`.

Proses mengubah data yang digunakan pada sistem ini memiliki mekanisme yang mirip, sehingga penjelasan proses ini akan dilakukan sekali saja. Gambar 4.39 berikut ini menunjukkan proses mengubah data account yang dilakukan oleh kelas `user.java`.

```

129     pg.doSQL("update users set " +
130             "username='"+pg.Q(update_username)+"'" +
131             ",email='"+pg.Q(update_email)+"'" +
132             "remark='"+pg.Q(update_remark)+"'" +
133             ",mobile_number='"+update_num_mobile+"'" +
134             "ident_no='"+update_ident_no+"'" +
135             "where id='"+id);
136     msg+="{ success: true }";

```

Gambar 4.39 Kode program untuk mengubah data

4.6.6 Proses Mencari Data

Proses mencari data pada dasarnya memiliki mekanisme yang mirip antara satu fungsi dengan yang lainnya. Cara yang digunakan bisa bermacam-macam. Pada tugas akhir ini penulis menggunakan dua fungsi kontrol dari Ext yaitu Form dan Grid. Form difungsikan untuk menampung berbagai filter pencarian sedangkan Grid untuk

menampilkan hasil pencarian. Untuk tampilan form dapat dilihat pada gambar 4.40. Dan kode program untuk membuat form tersebut dapat dilihat pada gambar 4.41.

Gambar 4.40 Form filter pencarian

Untuk membuat form filter pencarian diatas, penulis menggunakan kontrol FormPanel.

```

351     var panelSearchUser = new Ext.FormPanel({
352         title: 'Filter',
353         id: 'userSearchPanel',
354         autoHeight: true,
355         labelWidth: 125,
356         frame: true,
357         border: false,
358         defaults: {width: 210},
359         defaultType: 'textfield',
360         bbar: [tombolResetCari, '-', tombolCari],
361         items: [
362             new Ext.form.FieldSet({
363                 autoHeight: true,
364                 autoWidth: true,
365                 border: false,
366                 defaultType: 'textfield',
367                 items: [
368                     fieldCariID,
369                     fieldCariNama,
370                     fieldCariNoHP
371                 ]
372             })
373         ]
374     });

```

Gambar 4.41 Kode Program untuk fungsi pencarian

Data yang dimasukkan pada form akan mengambil parameter input untuk filter pencarian. Parameter akan diparsing pada file *Java Server Pages* dan menggunakan method pada kelas user.

```

85     String sql = "select * " +
86                 "from users " +
87                 "where ident_no like '%" + queryid + "%' " +
88                 "and username like '%" + querynama + "%' " +
89                 "and mobile_number like '%" + queryhp + "%' " +
90                 "and role >= '0' " +
91                 "order by id";
92     pg.execSQL(sql);

```

Gambar 4.42 Kode program untuk Query pada basis data

4.6.7 Proses Menampilkan Halaman Dinamis

Proses menampilkan halaman dinamis merupakan sebagian dari proses menampilkan data informasi yang dimiliki perusahaan. Ext menyediakan fungsi dimana kontrol seperti panel, window atau tab dapat menampilkan halaman dinamis didalamnya. Pada kontrol tersebut proses yang terjadi dasarnya sama..

Berikut contoh dari kontrol window dimana di dalamnya terdapat panel yang menampilkan halaman *Java Server Pages* berdasarkan parameter yang sebelumnya telah diparsingkan. Halaman *Java Server Pages* pada panel dapat menampilkan daftar pengguna. Berikut kode program dari fungsi yang men-update isi halaman *Java Server Pages*.

```

733     // fungsi untuk mengupdate isi halaman pada panel
734     winDetailProfile.show();
735     var u = Ext.getCmp('profileDetailPanel').getUpdater();
736     u.update({
737         url: '/TA/data/user/infoDetailProfile.jsp',
738         method: 'post',
739         scripts: true,
740         params: {id:selectedSearchID},
741         showLoadIndicator : true
742     });

```

Gambar 4.43 Kode program untuk update isi dari Panel

Setiap panel Ext ditampilkan maka, parameter akan diparsing untuk mengupdate isi tampilan dari halaman JSP sehingga tampilan dapat berubah sesuai dengan parameter dari kontrol Ext.

```

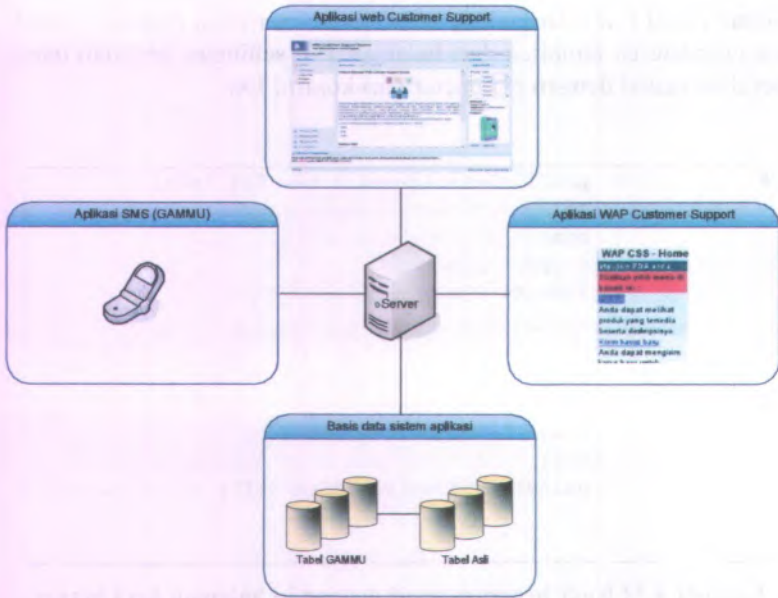
689     var profileDetailPanel = new Ext.Panel({
690         id: 'profileDetailPanel',
691         name: 'profileDetailPanel',
692         xtype: 'panel',
693         layout: 'fit',
694         width: 505,
695         height: 310,
696         bodyStyle: 'padding: 5px;',
697         autoScroll: true,
698         html: Ext.Ajax.request({
699             url: '/TA/data/user/infoDetailProfile.jsp',
700             params: {id: selectedSearchID} // parameter
701         })
702     });

```

Gambar 4.44 Kode program panel dengan isi halaman Java Server Pages

4.7 Implementasi Basis Data

Pembuatan basis data dilakukan dengan menggabungkan model fisik dari model diagram konseptual sistem aplikasi dengan basis data bawaan aplikasi GAMMU. Untuk tabel-tabel pada aplikasi GAMMU tidak diubah karena merupakan tabel yang secara default nantinya digunakan oleh *service* dari aplikasi GAMMU itu sendiri. Pada gambar 4.45 menunjukkan implementasi basis data terhadap sistem aplikasi *Customer Support*.



Gambar 4.45 Gambaran implementasi basis data pada sistem aplikasi

BAB V

UJI COBA DAN EVALUASI

Pada bab ini dijelaskan mengenai hasil penerapan *JavaScript library Ext* dalam aplikasi *Customer Support* dan proses uji coba aplikasi dengan menggunakan beberapa skenario uji coba. Uji coba ini akan menjadi bahan evaluasi yang nantinya akan dapat digunakan untuk perbaikan sistem dan pengembangan selanjutnya.

5.1 Hasil Penerapan JavaScript Library Ext

Dengan diterapkannya *JavaScript library Ext* dalam pengembangan tugas akhir ini, aplikasi yang dihasilkan menjadi lebih menarik dari pada pengembangan aplikasi berbasis web secara konvensional. Penulis pun dapat mengembangkan aplikasi dengan lebih cepat, karena kontrol-kontrol yang telah disediakan oleh Ext. Tidak semua kontrol dimanfaatkan penulis dalam pengembangan tugas akhir ini. Hal ini disebabkan karena faktor kebutuhan dan juga disebabkan keterbatasan Ext.

Faktor kebutuhan muncul karena kebutuhan penulis sudah dapat diakomodasi oleh Grid, Form, Window dan sebagainya seperti yang telah dijelaskan di bab empat. Sedangkan faktor keterbatasan muncul lebih dikarenakan ada kontrol yang belum stabil jika diterapkan dalam pengembangan aplikasi serta adanya perlunya pemahaman yang lebih mendalam tentang kontrol yang dikembangkan. Permasalahan yang muncul antara lain:

- Terdapat fungsi yang harus dibuat dan didefinisikan terlebih dahulu oleh pemrogram sehingga beberapa atribut yang tidak dapat dimiliki oleh kontrol tertentu tersebut dapat berjalan semestinya. Hal ini disebabkan belum semua kontrol yang dimiliki Ext dapat mengatasi berbagai kebutuhan dari proses bisnis yang diinginkan. Contoh dapat ditemui pada tipe kontrol

input HTML editor dimana untuk proses validasi inputan kosong maka pemrogram harus membuat fungsi tersendiri. Berbeda untuk tipe inputan biasa yang dapat dibuat melalui pendefinisian atribut `allowBlank` yang dimilikinya.

- Tidak semua browser mendukung Ext. Pada saat pengembangan aplikasi web *Customer Support*, penulis menggunakan browser mozilla firefox versi 3 sebagai media untuk menjalankan sistem. Penulis juga sudah mencoba pada mozilla firefox versi 2.0 dan aplikasi web *Customer Support* dapat berjalan dengan baik. Pada browser tertentu fungsi dari kontrol Ext tidak dapat berjalan, seperti pada browser internet explorer. Penulis mengatakan bahwa ini bukan suatu kesalahan namun merupakan suatu keterbatasan. Berdasarkan uji coba yang telah dilakukan penulis, maka aplikasi web *Customer Support* dapat berjalan dengan maksimal pada lingkungan browser mozilla firefox versi 2.0 keatas. Pada sub bab berikutnya akan dijelaskan lebih detail mengenai lingkungan dan skenario uji coba yang dilakukan penulis terhadap aplikasi web *Customer Support* yang telah dikembangkan.

5.2 Lingkungan Uji Coba

Pada bagian ini akan dijelaskan mengenai lingkungan pengujian aplikasi, yang meliputi perangkat lunak dan perangkat keras. Spesifikasi dari masing-masing perangkat yang digunakan dalam uji coba aplikasi web *Customer Support* tampak pada tabel berikut:

Tabel 5.1 Lingkungan Pengujian Aplikasi

Perangkat Keras	<p>Prosesor : Intel(R) Pentium (R) Dual CPU E2160 @ 1.80 Ghz</p> <p>Memori : 1016 MB DDRAM</p> <p>Modem : Sony Erricson w700i</p>
Perangkat Lunak	<p>Sistem Operasi : Windows XP SP 2.</p> <p>Web Server: Apache Tomcat 5.5.20</p>

Web Browser: Mozilla Firefox Version 3.0 Basis Data : MySQL 5.0.37 Simulator WAP : Openwave V7 Simulator, version 7.0.107

5.3 Skenario Uji Coba

Uji coba dilakukan untuk menguji jalannya aplikasi mulai dari proses input sampai dengan proses output dan mengetahui kesesuaian antara desain dengan kenyataan. Data-data yang digunakan dalam proses uji coba ini menggunakan data dummy (rekayasa).

Aplikasi web *Customer Support* bisa dikatakan lulus uji coba apabila menu-menu yang dipilih bisa berjalan dengan baik dan tidak ada pesan error yang muncul.

Skenario uji coba yang akan dilakukan adalah sebagai berikut:

1. Login dan logout.
2. Mengelola data pengguna.
3. Mengelola data pesan.
4. Mengelola data knowledgebase.
5. Menampilkan informasi yang disediakan oleh perusahaan.

5.4 Pelaksanaan Skenario Uji Coba

Pada aplikasi ini terdapat 5 skenario utama. Penjelasan pelaksanaan skenario uji coba akan disertai dengan gambar yang menunjukkan keadaan sistem yang sebenarnya.

5.4.1 Login dan Logout

Untuk menggunakan aplikasi web *Customer Support*, semua pengguna harus melakukan proses Login terlebih dahulu. Setelah membuka URL Sistem di browser, maka form login akan terbuka (Gambar 5.1).

Pengguna memasukkan *username* dan *password* kedalam kolom yang ada di form login. Lalu pengguna menekan tombol login. Sistem akan melakukan autentifikasi berdasarkan data masukan tersebut. Jika data pengguna dengan *username* yang dimasukkan tidak ditemukan atau apabila *password* salah, akan pesan kesalahan sebagaimana Gambar 5.2.

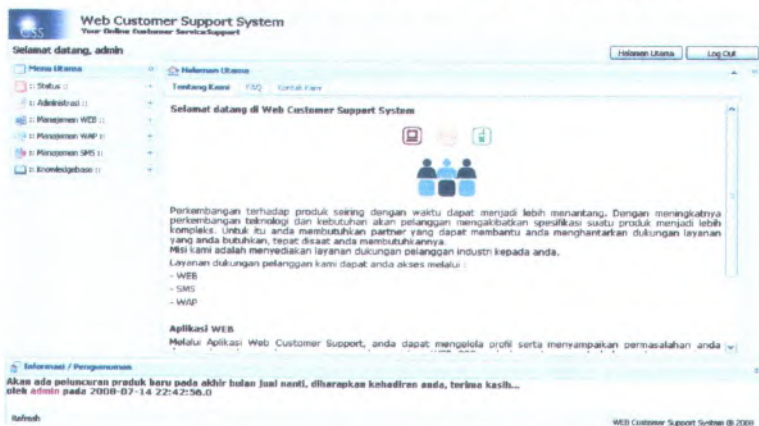


Gambar 5.1 Halaman Form Login



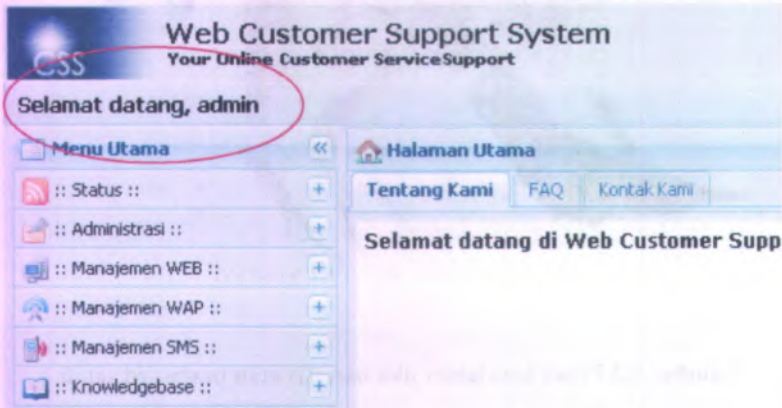
Gambar 5.2 Pesan kesalahan jika user ID atau password salah

Jika Proses Login Sukses, maka pengguna akan disugahi halaman utama (Gambar 5.3). Halaman utama ini memiliki menu yang berbeda-beda sesuai dengan otoritas masing-masing pengguna.



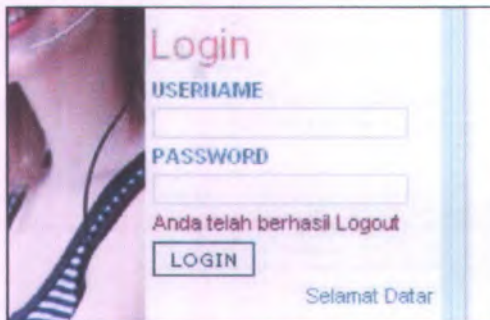
Gambar 5.3 Halaman utama admin

Di halaman utama akan muncul personalisasi berupa salam ke pengguna (Gambar 5.4).



Gambar 5.4 Personalisasi pengguna

Setelah menu Log Out dipilih, pengguna akan dibawa ke halaman Login dengan pesan yang menunjukkan bahwa Log Out telah berhasil dilakukan (Gambar 5.5).



Gambar 5.5 Logout berhasil

5.4.2 Mengelola Data Pengguna

Uji coba skenario mengelola data pengguna terdiri dari tambah data pengguna, lihat data profil pengguna, ubah data profil pengguna, non-aktifkan pengguna, dan cari data pengguna.

Untuk menambah data pengguna, pengguna bisa memilih menu Administrasi – Pengguna – Tambah Pengguna. Aplikasi akan menampilkan form tambah pengguna. Lalu pengguna bisa memasukkan data identitas atau profil dari pengguna, dalam hal ini data pelanggan. Form yang dibangun oleh kontrol form pada Ext dapat melakukan proses validasi terhadap inputan yang dimasukkan oleh pengguna secara langsung tanpa harus melakukan proses submit form.

Contoh dapat dilihat pada proses memasukkan nomer identitas atau nomer handphone dimana input yang boleh dimasukkan hanyalah inputan angka bukan huruf. Selain itu untuk inputan yang bersifat *mandatory* atau harus diisi akan menghasilkan tanda atau pesan kecil yang memperingatkan pengguna untuk mengisi inputan tersebut. Jenis-jenis inputan dapat diatur melalui pendefinisian atribut yang dimiliki oleh tiap-tiap kontrol text field yang dibuat pada kode program. Tipe inputan nomor atau huruf, alamat e-mail, panjang string, tipe *password*, inputan *mandatory* merupakan jenis atribut yang dapat diatur pada sebuah kontrol input pada form Ext. *Behavior* tersebut pada umumnya dapat ditangani dengan menambahkan syntax *JavaScript* pada suatu halaman web aplikasi, namun pada Ext, text field merupakan sebuah kontrol pada form Ext yang dapat diatur atributnya secara tersendiri sehingga dapat menangani masalah *error handling* inputan.

Tambah Pengguna

No. Identitas: 0111343434343434 ⓘ

Username: anis ⓘ

Password: ●●●●

No. Handphone: 62316030447033432 ⓘ

e - mail: tes@tes.com

Catatan: catatan tentang pengguna ..

Simpan Reset

ⓘ Mohon masukan nomor identitas dengan benar

Gambar 5.6 Form menampilkan pesan peringatan bahwa nomor identitas tidak sesuai dengan ketentuan

Tambah Pengguna

No. Identitas: 5204100111

Username: anis

Password: ●●●●

No. Handphone: 623160304470

e - mail: tes ⓘ

Catatan: catatan tentang pengguna ..

Simpan Reset

ⓘ This field should be an e-mail address in the format "user@domain.com"

Daftar Prod	
ID Produk	Nam
1	Prod
2	Prod
3	Prod
4	Prod
5	Prod

Gambar 5.7 Form menampilkan pesan peringatan input alamat email tidak sesuai dengan format alamat email yang valid

Ketika pengguna melakukan proses penyimpanan data ketika form belum valid maka form akan menampilkan pesan peringatan bahwa form belum sepenuhnya diisi dengan benar sehingga proses penyimpanan data pelangan tidak akan dilakukan (Gambar 5.8).

The screenshot shows a web form titled "Tambah Pengguna" with the following fields and values:

- No. Identitas: 5204100111343434C (with a red error icon)
- Username: anis
- Password: *****
- No. Handphone: 6231603044703343c (with a red error icon)
- e - mail: (empty)
- Catatan: (empty)

A modal dialog box titled "Peringatan" is displayed in the center, containing the text "Mohon masukkan data dengan benar !" and an "OK" button. At the bottom right of the form, there are "Simpan" and "Reset" buttons.

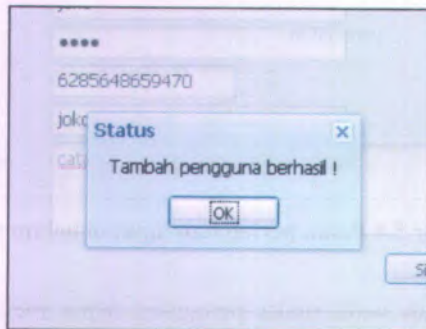
Gambar 5.8 Pesan peringatan inputan belum valid

Ketika form telah valid maka pengguna dapat melakukan proses penyimpanan. Untuk proses pada insert data pada basis data ketika terdapat data yang telah sama maka proses ditangani oleh kelas java. Ketika proses gagal maka kelas java akan mengembalikan nilai *false* kepada form dan form Ext akan menampilkan pesan respon gagal melakukan penyimpanan data (gambar 5.9). Pesan respon dapat ditentukan oleh pemrogram sesuai dengan kejadian yang ada pada kegagalan proses penyimpanan yang terjadi.

The screenshot shows a modal dialog box titled "Tambah pengguna gagal !" with the text "No identitas telah ada." and an "OK" button. The background shows a blurred view of the "Tambah Pengguna" form.

Gambar 5.9 Pesan peringatan gagal menyimpan data

Ketika pengguna telah memasukkan data yang benar-benar sudah valid dimana pada basis data data merupakan data yang baru atau belum ada, maka akan muncul pesan status bahwa proses penyimpanan data telah berhasil (Gambar 5.10).



Gambar 5.10 Pesan status sukses menyimpan data

Untuk melihat data, pengguna bisa memilih menu Administrasi – Pengguna – Profil Pengguna pada daftar menu di bagian kiri halaman. Selanjutnya sistem akan menampilkan daftar pengguna (Gambar 5.11).

Untuk mengubah data pelanggan, administrator terlebih dahulu memilih salah satu data pelanggan yang terdapat pada Grid Panel daftar pengguna (Gambar 5.12) dan menekan tombol Edit Pengguna. Jika administrator belum memilih salah satu data maka akan muncul pesan peringatan seperti pada gambar 5.13. Jika pengguna sudah memilih data dengan benar dari Grid Panel lalu menekan tombol Edit Pengguna, maka sistem akan menampilkan window dimana pada window tersebut terdapat form ubah data profil pelanggan (Gambar 5.14). Fungsi ubah data pengguna disediakan untuk mengubah data profil pelanggan. Gambar 5.15 menunjukkan pesan status setelah proses ubah data pengguna berhasil.

Menu Manajemen Pengguna

Daftar pengguna

Edit Pengguna Non-Aktifkan Pengguna

ID Pengguna	No. Identitas	Username	No. Handphone	E - mail
49	5204100013	rizka	+6285655338021	rizka_zuffitri@yahoo.com
Catatan : ini pengguna rizka				
48	5204100032	segaf	+6285232642001	oyex10@yahoo.com
Catatan : ini pengguna segaf				
47	5204100078	noval	+6285648453677	noval78@yahoo.com
Catatan : ini pengguna noval				
46	5204100033	nandha	+628563451206	nandha@yahoo.com
Catatan : ini pengguna nandha				
41	5204100056	faizal	+623160613665	fmahananto@yahoo.com
Catatan : ini pengguna faizal				

Page 1 of 3 Preview Menampilkan pengguna 1 - 10 dari 22

Gambar 5.11 Daftar pengguna

Gambar 5.12 menunjukkan proses memilih pengguna pada grid dan memilih tombol Edit Pengguna untuk mengubah profil

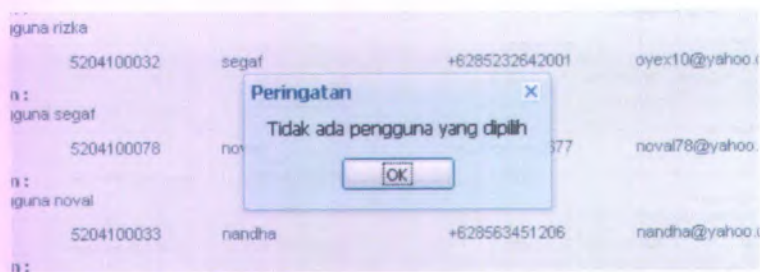
Daftar pengguna

Edit Pengguna Non-Aktifkan Pengguna

ID Pengguna	No. Handphone	E - mail
49	+6285655338021	rizka_zuffitri@yahoo.com
Catatan : ini pengguna rizka		
48	+6285232642001	oyex10@yahoo.com
Catatan : ini pengguna segaf		
47	+6285648453677	noval78@yahoo.com
Catatan : ini pengguna noval		
46	+628563451206	nandha@yahoo.com
Catatan : ini pengguna nandha		
41	+623160613665	fmahananto@yahoo.com
Catatan : ini pengguna faizal		

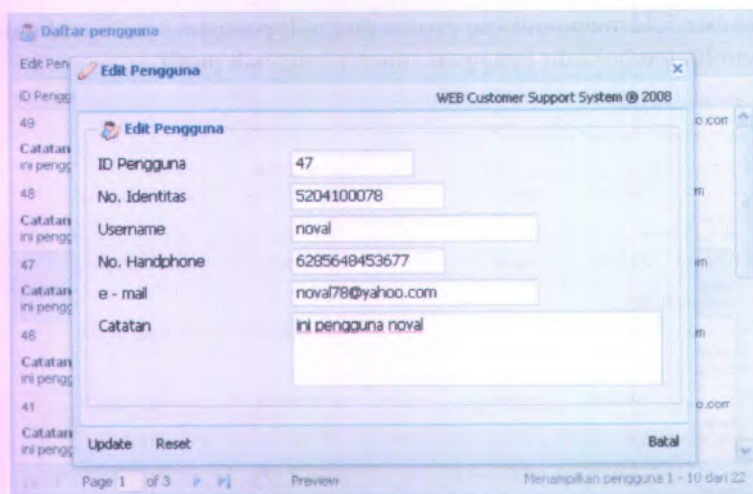
Page 1 of 3 Preview Menampilkan pengguna 1 - 10 dari 21

Gambar 5.12 Memilih pengguna untuk diubah

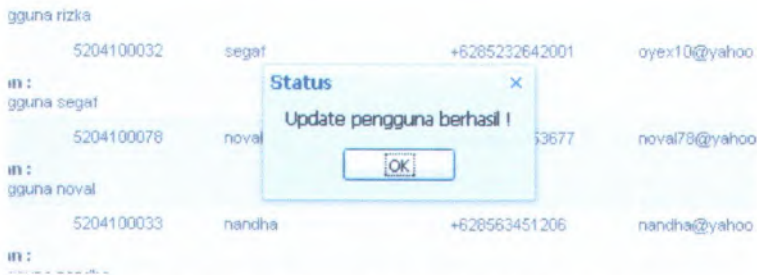


Gambar 5.13 Pesan peringatan ketika tidak ada data yang dipilih

Gambar 5.13 menunjukkan pesan peringatan sistem jika pengguna memilih tombol edit pengguna namun sebelumnya tidak ada pengguna yang dipilih untuk diedit. Sedangkan pada gambar 5.14, jika pengguna memilih salah satu pengguna pada grid dan menekan tombol Edit Pengguna maka akan muncul form Edit Pengguna.



Gambar 5.14 Form edit data pengguna

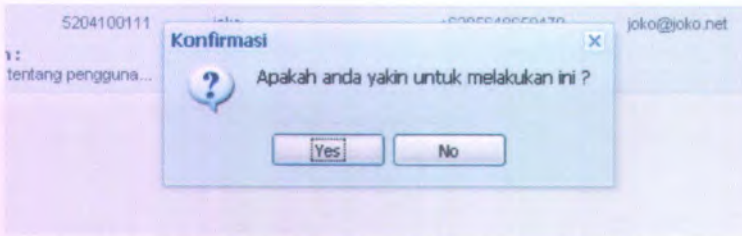


Gambar 5.15 Pesan status data berhasil diubah

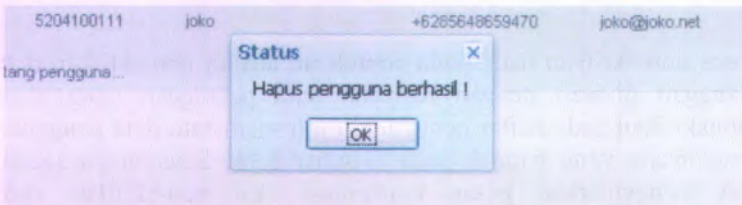
Proses non-aktifkan data, pada contoh ini adalah non-aktifkan data pelanggan diawali dengan memilih data pelanggan yang akan dinonaktifkan pada daftar pengguna. Pilih salah satu data pengguna sebagaimana yang tampak pada Gambar 5.16. Selanjutnya sistem akan mengeluarkan pesan konfirmasi aksi non-aktifkan data pelanggan untuk memastikan apakah pengguna benar-benar akan menon-aktifkan data pelanggan (Gambar 5.17). Jika konfirmasi telah disetujui oleh pengguna, maka sistem akan mengubah status data pelanggan yang dimaksud lalu menampilkan pesan status yang memberitahu bahwa proses non-aktifkan data pelanggan telah berhasil (Gambar 5.18).

Daftar pengguna				
Edit Pengguna		Non-Aktifkan Pengguna		
ID Pengguna	No. Identif	Ubah status pengguna menjadi tidak aktif	No. Handphone	E - mail
53	5204100100	sulton	+623160327003	sulthon@abc.com
Catatan : pengguna sulton				
54	5204100111	joko	+6285648659470	joko@joko.net
Catatan : catatan tentang pengguna...				

Gambar 5.16 Memilih pelangan untuk dinon-aktifkan

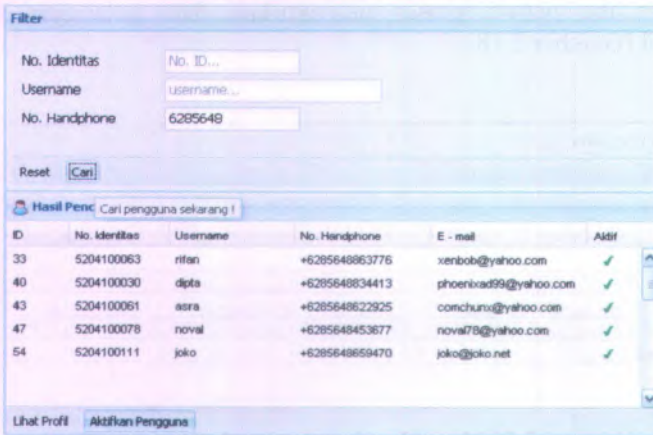


Gambar 5.17 Pesan konfirmasi aksi non-aktifkan pelanggan



Gambar 5.18 Pesan status aksi non-aktifkan pengguna berhasil

Untuk proses pencarian data pelanggan pengguna dapat memilih menu Administrasi – Pengguna – Cari Pengguna.



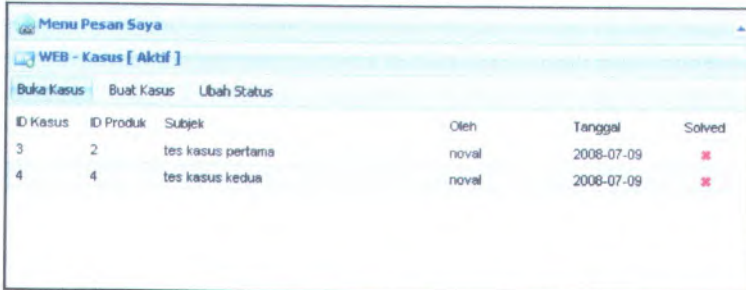
Gambar 5.19 Form untuk melakukan pencarian data pelanggan

Proses mencari data pelanggan dilakukan dengan terlebih dahulu memasukkan parameter filter pencarian pada form Filter (Gambar 5.19). Parameter filter pencarian merupakan suatu pilihan yang dapat dilakukan oleh pengguna. Pengguna dapat melakukan pencarian tanpa harus memasukkan kata kunci untuk filter pencarian.

5.4.3 Mengelola Data Pesan

Uji coba skenario mengelola data pesan dibagi menjadi tiga bagian berdasarkan *multiple touchpoint* aplikasi yang digunakan. Pada dasarnya proses pengolahan data pesan merupakan proses yang sama, perbedaan hanyalah pada langkah dan ketentuan yang harus dilakukan dalam mengelola data pesannya. Secara umum proses yang ada adalah lihat data pesan, buat atau balas data pesan.

Pada aplikasi *touchpoint* yaitu web dan WAP untuk membuat pesan baru sebelumnya pelanggan terlebih dahulu harus membuat kasus. Hal ini dimaksudkan untuk menentukan pengelompokan permasalahan berdasarkan kasus atau permasalahan yang dimiliki oleh pelanggan itu sendiri. Kasus memiliki atribut subjek dan produk yang dipermasalahkan. Untuk itu pada aplikasi web *Customer Support* sisi pelanggan, pelanggan dapat memilih menu Pesan Saya – Kasus Aktif pada daftar menu di bagian kiri halaman. Selanjutnya sistem akan menampilkan daftar kasus aktif yang dimiliki oleh pelanggan tersebut (Gambar 5.20).



ID Kasus	ID Produk	Subjek	Oleh	Tanggal	Solved
3	2	tes kasus pertama	novel	2008-07-09	✘
4	4	tes kasus kedua	novel	2008-07-09	✘

Gambar 5.20 Daftar kasus web aktif yang dimiliki pelanggan



Untuk membuat kasus baru pelanggan dapat memilih tombol Buat Kasus. Lalu sistem akan menampilkan window dimana terdapat form untuk membuat kasus baru beserta pesan pertama pada kasus tersebut (Gambar 5.21).

Gambar 5.21 Form buat kasus dan pesan baru

Sebelum dimasukkan kedalam basis data, data-data tersebut akan divalidasi terlebih dahulu. Validasi yang dilakukan adalah mengecek apakah kolom-kolom yang wajib diisi tidak kosong. Jika data masukan tidak benar, maka akan ada pesan kesalahan. Seperti dijelaskan sebelumnya kontrol HTML editor belum dapat secara langsung dapat mengatasi inputan kosong. Maka penulis menggunakan proses validasi data pada backend dari proses. Pada halaman *Java Server Pages* diletakkan fungsi untuk melakukan cek terhadap input pesan. Jika pesan kosong maka proses tidak dapat dilanjutkan dan mengembalikan nilai *false* pada form sehingga form menampilkan pesan kesalahan inputan data pesan. Ketika form telah memenuhi syarat untuk diproses insert pada basis data, maka pesan status penambahan data akan ditampilkan untuk menunjukkan

bahwa proses penambahan data kasus dan pesan baru telah berhasil dilakukan (Gambar 5.22).

ID Produk	Subjek	Oleh	Tanggal
2	tes kasus pertama	noval	2008-07-09
4	tes kasus kedua		2008-07-09

Status X

Mengirim pesan berhasil !

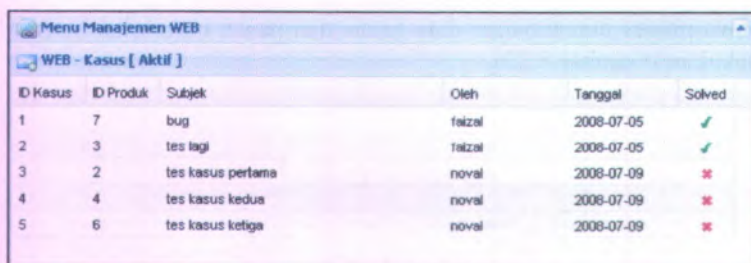
[OK]

Gambar 5.22 Pesan status pengiriman atau buat kasus dan pesan baru berhasil

Untuk melihat data pesan yang dikirimkan oleh pelanggan maka, administrator dapat melakukan pengecekan pesan baru dengan status aktif pada daftar pesan web. Pada daftar pesan kasus web tersebut dapat dilihat kasus yang sebelumnya telah dibuat oleh pelanggan. Kasus tersebut memiliki status belum *solved*, atau dalam kata lain belum terselesaikan.

Kasus dapat terselesaikan atau mempunyai status *solved* ketika pelanggan merasa kasus tersebut telah selesai didiskusikan dengan pihak manajemen perusahaan, dalam hal ini adalah administrator dan pelanggan mengubah status kasus tersebut menjadi *solved*.

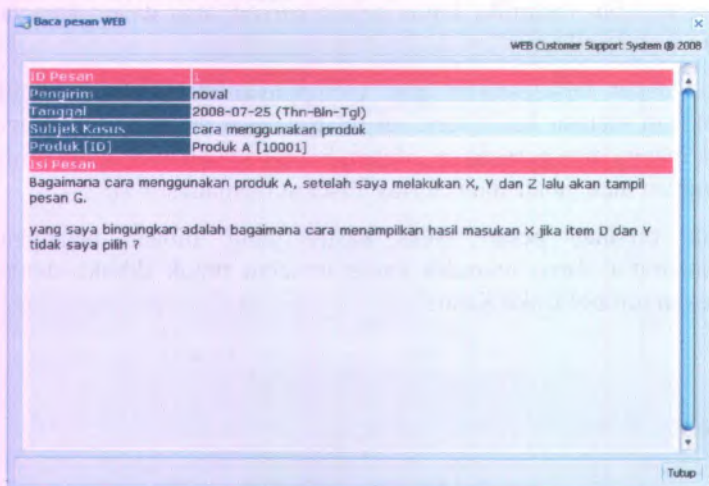
Untuk melihat pesan pada kasus yang dibuat pelanggan, administrator dapat memilih kasus tersebut untuk dibuka dengan menekan tombol Buka Kasus.



ID Kasus	ID Produk	Subjek	Oleh	Tanggal	Solved
1	7	bug	faizal	2008-07-05	✓
2	3	tes lagi	faizal	2008-07-05	✓
3	2	tes kasus pertama	noval	2008-07-09	✗
4	4	tes kasus kedua	noval	2008-07-09	✗
5	6	tes kasus ketiga	noval	2008-07-09	✗

Gambar 5.23 Kasus pesan baru tampil pada daftar pesan

Sistem akan menampilkan daftar pesan yang terdapat pada kasus tersebut. Pada daftar tersebut administrator dapat melihat atribut dari pesan serta preview dari isi pesan. Jika administrator ingin melihat pesan tertentu secara menyeluruh maka administrator dapat memilih pesan menekan tombol Baca Pesan. Sistem akan menampilkan window baru dimana window tersebut akan menampilkan halaman dinamis. Halaman dinamis tersebut berisi keseluruhan isi dari pesan yang dipilih. Halaman dinamis pada window baca pesan merupakan halaman *Java Server Pages*.



Gambar 5.24 Window baca pesan

Untuk melakukan pengiriman pesan balasan, administrator dapat membuat data pesan baru. Dimana administrator harus memilih salah satu pesan yang akan dibalas dan menekan tombol Tulis Pesan. Proses ini sama persis dengan proses yang dilakukan oleh pelanggan pada saat pembuatan data kasus dan pesan baru. Perbedaan terletak pada inputan yang dimasukkan. Adminstrator tidak perlu memasukkan subjek kasus dan produk yang dipermasalahkan karena proses yang dilakukan adalah membalas data pesan.

Jika pesan telah berhasil dibuat dan diproses oleh pengguna maka sistem akan menyimpan data pesan tersebut pada basis data. Sehingga data pesan tersebut akan muncul pada daftar pesan pada kasus yang dimiliki oleh pelanggan yang membuat kasus tersebut serta pada pihak administrator.

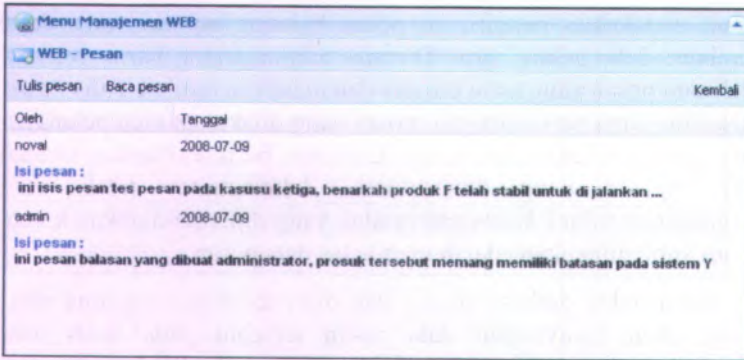
Dapat dilihat bahwa proses buat kasus dan pesan baru dengan proses balas pesan adalah proses yang serupa namun hanya berbeda pada langkah-langkah yang harus dilakukan.

The screenshot shows a web browser window titled "Kirim pesan WEB" with a subtitle "WEB Customer Support System @ 2008". The main content area is titled "Tulis pesan baru" and contains the following form fields:

- Pengirim:** admin
- ID Produk:** 6
- Subjek:** tes kasus ketiga
- Pesan balasan:** A rich text editor containing the text: "ini pesan balasan yang dibuat administrator, produk tersebut memang memiliki batasan pada sistem Y".

At the bottom of the form, there are three buttons: "Kirim", "Reset", and "Batal".

Gambar 5.25 Form tulis pesan baru untuk membalas



Gambar 5.26 Pesan balasan tampil pada daftar pesan kasus

Ketika pelanggan merasa kasus tertentu yang dimilikinya telah terselesaikan, maka pelanggan dapat mengubah status kasus tersebut menjadi *solved*. Kasus dengan status *solved* akan ditindaklanjuti oleh administrator untuk ditutup atau dengan kata lain dinon-aktifkan. Kasus yang telah dinon-aktifkan tidak akan muncul pada daftar kasus aktif pada daftar kasus pelanggan maupun administrator (Gambar 5.29).

ID Kasus	ID Produk	Subjek	Ubah status kasus	Oleh	Tanggal	Solved
3	2	tes kasus pertama		noval	2008-07-09	✘
4	4	tes kasus kedua		noval	2008-07-09	✘
5	6	tes kasus ketiga		noval	2008-07-09	✘

Gambar 5.27 Memilih data kasus untuk diubah statusnya



Gambar 5.28 Pesan status Ubah Status data kasus berhasil

WEB - Kasus [Aktif]						
Buka Kasus	Buat Kasus	Ubah Status				
ID Kasus	ID Produk	Subjek	Ubah status kasus	Oleh	Tanggal	Solved
3	2	tes kasus pertama		noval	2008-07-09	✘
4	4	tes kasus kedua		noval	2008-07-09	✘
5	6	tes kasus ketiga		noval	2008-07-09	✔

Gambar 5.29 Data kasus yang telah diubah statusnya

Untuk proses memnon-aktifkan dan melakukan pencarian kasus (Gambar 5.30), proses yang dilakukan sama dengan proses menon-aktifkan dan melakukan pencarian data pelanggan, perbedaan hanya pada data dan jenis data yang diolah atau proses.

Pada Gambar 5.31 menunjukkan proses mengubah status kasus menjadi tidak aktif berhasil dilakukan. Untuk mengubah status kasus maupun mengubah status pengguna dari aktif menjadi tidak aktif ataupun sebaliknya dapat dilakukan pada menu pencarian. Pada menu pencarian terdapat fungsi untuk mengubah status pengguna ataupun kasus menjadi aktif atau tidak aktif.

The screenshot shows a web interface for searching cases. At the top, there is a 'Filter' section with several input fields: 'Subjek' (with a text input 'subjek...'), 'Produk' (with a dropdown arrow), 'Oleh' (with a text input), and 'Urutkan Berdasarkan' (with a dropdown arrow). Below these are 'Reset' and 'Cari' buttons. A 'Hasil Pencarian Kasus' section contains a table with columns: 'ID Kasus', 'ID Produk', 'Subjek', 'Oleh', 'Tanggal', 'Aktif', and 'Solved'. A dropdown menu is open over the 'Produk' field, listing 'Produk A' through 'Produk G'. At the bottom of the interface are 'Buka Kasus' and 'Ubah Status' buttons.

ID Kasus	ID Produk	Subjek	Oleh	Tanggal	Aktif	Solved
1	10001	cara menggunakan produk	noval	2008-07-25	✓	X
2	10007	serial number aplikasi tidak berjalan	citra	2008-07-25	X	✓
3	10003	proses simpan tidak berhasil	citra	2008-07-27	X	✓
4	10005	mengirim data	citra	2008-07-27	✓	X

Gambar 5.30 Form pencarian kasus

Pada gambar 5.30, menu pencarian kasus akan menampilkan form pencarian kasus dimana terdapat fungsi ubah status kasus.

The screenshot shows a table of cases with a dialog box open. The table has columns for ID, Produk, Subjek, Oleh, and Tanggal. A dialog box titled 'Status' is open, displaying the message 'Menutup kasus berhasil !' and an 'OK' button. The table shows two rows of cases, with the second row having a green checkmark in the 'Solved' column.

ID	Produk	Subjek	Oleh	Tanggal	Solved
4	4	tes kasus kedua	noval	2008-07-09	X
5	6	tes kasus ketiga	noval	2008-07-09	✓

Gambar 5.31 Pesan status menutup / non-aktikan kasus berhasil

Menu Manajemen WEB						
WEB - Kasus [Tidak Aktif]						
ID Kasus	ID Produk	Subjek	Oleh	Tanggal	Solved	
5	6	tes kasus ketiga	noval	2008-07-09	✓	

Gambar 5.32 Daftar data kasus tidak aktif

Untuk uji coba skenario pengiriman pesan melalui *touchpoint* aplikasi WAP, langkah yang harus dilakukan pelanggan sama pada proses pembuatan data kasus dan pesan melalui *touchpoint* web. Perbedaan terletak pada antarmuka halaman WAP yang ditampilkan. Pada tugas akhir ini penulis menggunakan simulator aplikasi WAP yang bersifat *freeware* milik *Openwave System Inc.*



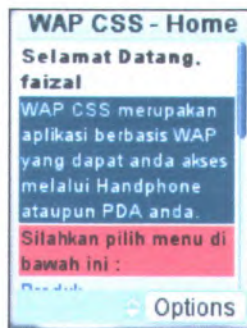
Gambar 5.33 Openwave Simulator

Untuk mengelola data pesan yang dimilikinya pelanggan harus terlebih dahulu login pada sistem. Kemudian memilih menu Kirim Kasus baru. Sistem akan menampilkan Form Kirim Kasus dimana form yang ditampilkan sama dengan form Buat Kasus pada aplikasi web *Customer Support*. Untuk prosedur uji coba selanjutnya langkah yang dilakukan oleh pengguna yaitu pelanggan dengan administrator sama dengan yang dilakukan pada aplikasi web *Customer Support*.

The image shows a mobile web application (WAP) login screen. At the top, the title is 'LOGIN'. Below the title is a header 'Form Login'. There are two input fields: 'Username :' with the text 'faisal' entered, and 'Password :' with '*****' entered. At the bottom of the form, there are four buttons: 'Login', 'Reset', 'Select', and 'Options'.

Gambar 5.34 Form Login aplikasi WAP Customer Support

Pada gambar 5.34 dapat dilihat proses login dimana pelanggan memasukkan *username* dan *password* yang dimilikinya. Setelah melakukan login maka akan ditampilkan halaman utama aplikasi WAP *Customer Support* seperti yang ditampilkan pada gambar 5.35. Pada halaman utama sistem menyediakan menu bagi pelanggan untuk melihat data produk, buat kasus baru atau melihat daftar pesan. Dimana pada menu daftar pesan pelanggan dapat melakukan berbagai aktivitas yang berhubungan dengan pengelolaan pesan WAP yang dimilikinya.



Gambar 5.35 Halaman utama aplikasi WAP Customer Support

Hal yang sama pada aplikasi web dapat dilakukan pelanggan untuk membuat kasus baru, pelanggan harus melakukan pengisian form buat kasus dengan inputan-inputan subjek kasus, produk serta isi pesan pertama yang akan dibuat (Gambar 5.36).

Gambar 5.36 Form buat data kasus baru WAP Customer Support

The screenshot shows a mobile interface titled "WAP CSS - Kasus". It displays a list of active cases. The first case is selected, with "Tes" highlighted in blue. The list has columns for "Subjek kasus", "Oleh", and "Solved".

Subjek kasus	Oleh	Solved
1 Tes produk A	faizal	[X]
2 Akmjd im tes	faizal	[V]

At the bottom, there are "Select" and "Options" buttons.

Gambar 5.37 Daftar data kasus aktif WAP Customer Support

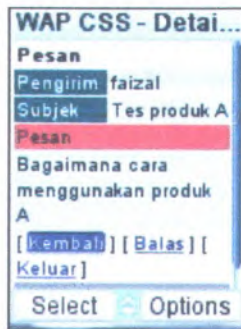
Pelanggan dapat melihat daftar kasus serta daftar pesan yang dimilikinya. Gambar 5.37 menunjukkan daftar data kasus dengan status aktif yang dimilikinya. Sedangkan gambar 5.38 merupakan daftar data pesan pada kasus tersebut.

The screenshot shows a mobile interface titled "WAP CSS - Pesan". It displays details for a selected case. The "Subjek Kasus" is "Tes produk A". Below this is a table with columns "Oleh" and "Tanggal".

Oleh	Tanggal
1 faizal	2008-07-22 21:06:52.0

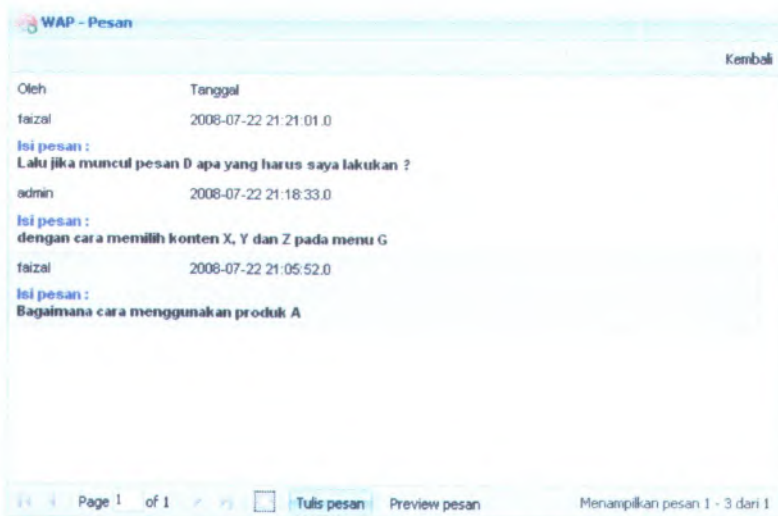
Below the table, there are links "[Kembali]" and "[Keluar]". At the bottom, there is a status bar "WAP CSS © 2008" and "Select" and "Options" buttons.

Gambar 5.38 Daftar data pesan pada kasus yang dipilih



Gambar 5.39 Halaman isi pesan WAP Customer Support

Pesan yang dikirimkan pelanggan akan muncul pada menu pesan WAP pada aplikasi web Customer Support sisi manajemen (Gambar 5.40).



Gambar 5.40 Daftar data pesan WAP pada aplikasi web

WAP CSS - Pesan	
produk A	
Oleh	Tanggal
1 admin	2008-07-22 21:24:49.0
2 faizal	2008-07-22 21:21:01.0
3 admin	2008-07-22 21:18:33.0
4 faizal	2008-07-22 21:06:52.0
Select	Options

Gambar 5.41 Daftar data pesan WAP pada aplikasi WAP

Uji coba skenario pada pengelolaan data pesan SMS pada aplikasi web berbeda dengan pengelolaan pesan pada kedua *touchpoint* sebelumnya. Pada aplikasi SMS tidak terdapat ketentuan dimana pesan yang dikirim harus dikelompokkan pada suatu kasus tertentu. Pelanggan dapat secara bebas mengirim pesan tanpa adanya format isi pesan tertentu. Pelanggan dapat mengirimkan pesan melalui perangkat handphonenya ke nomor SMS yang dipakai oleh modem pada perusahaan. Setelah pelanggan mengirimkan pesan maka pesan akan tampil pada daftar pesan di aplikasi web *Customer Support* sisi manajemen atau administrator. Pesan yang tampil pada daftar pesan adalah hanya pesan yang dikirim oleh pelanggan yang nomornya terdaftar pada nomor pengguna sistem atau pelanggan terdaftar.

Pada aplikasi handphone pengguna dapat mengirimkan pesan SMS dengan cara biasa seperti pada pengiriman SMS umumnya. Pada gambar 5.43 dapat dilihat proses pengiriman pesan ke nomor SMS aplikasi web *Customer Support*.



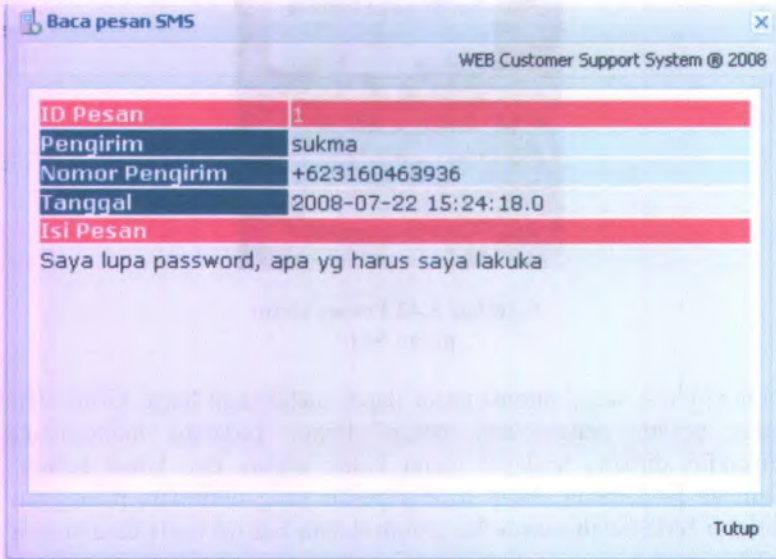
Gambar 5.42 Proses kirim pesan SMS

Pada aplikasi web, administrator dapat melakukan baca, kirim atau balas pesan, pengelolaan pesan dibagi kedalam menu-menu tersendiri dimana terdapat menu kotak masuk dan kotak keluar. Gambar 5.43 menunjukkan bahwa pesan yang dikirimkan pelanggan melalui SMS telah masuk ke sistem dalam hal ini basis data sistem aplikasi web *Customer Support*.

SMS - Kotak Masuk [Baru]			
Baca Pesan		Balas	
ID SMS	Pengirim	Nomor	Tanggal Terima
1	sukma	+623160463936	2008-07-22 15:24:18.0
Pesan :			
Saya lupa password, apa yg harus saya lakukan			

Gambar 5.43 Daftar pesan SMS baru

Administrator / Manajemen dapat membaca pesan pada window baru dengan menekan tombol baca Pesan (Gambar 5.44).



Gambar 5.44 Baca Pesan SMS

Untuk melakukan balas pesan administrator dapat menekan tombol Balas dan mengisi form balas pesan SMS yang ditampilkan sistem (Gambar 5.45). Pada form terdapat field masukan pengirim, nomor dan isi pesan pengirim SMS. Kemudian terdapat field masukan pesan untuk membalas pesan yang dipilih.



Balas Pesan SMS

WEB Customer Support System @ 2008

Tulis pesan SMS

Pengirim: sukma

Nomor: 623160463936

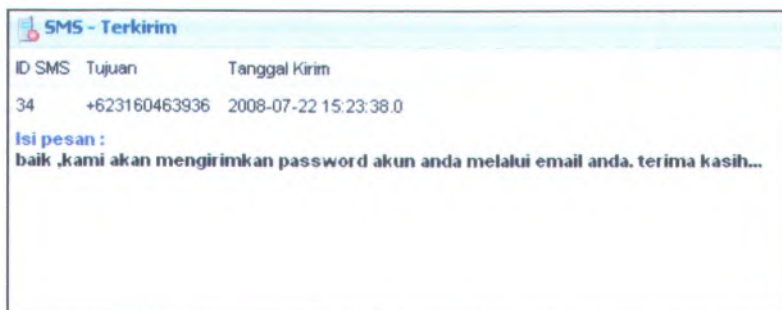
Isi pesan: Saya lupa password, apa yg harus saya lakuka

Isi pesan balasan: baik ,kami akan mengirimkan password akun anda melalui email anda. terima kasih...

Kirim Reset Batal

Gambar 5.45 Form balas pesan SMS

Pesan yang telah terkirim akan muncul pada menu SMS – Terkirim (Gambar 5.46).



ID SMS	Tujuan	Tanggal Kirim
34	+623160463936	2008-07-22 15:23:38.0

Isi pesan :
baik ,kami akan mengirimkan password akun anda melalui email anda. terima kasih...

Gambar 5.46 Daftar pesan SMS terkirim

Pada gambar 5.47 dapat dilihat pesan yang dikirimkan melalui aplikasi web *Customer Support* dapat sampai pada pelanggan..



Gambar 5.47 Baca pesan SMS pada handphone

5.4.4 Mengelola Data Knowledgebase

Uji coba skenario mengelola data knowledgebase dibagi menjadi dua bagian berdasarkan jenis informasi knowledgebase yang disediakan, antara lain mengelola data artikel dan mengelola data istilah. Pada dasarnya proses pengolahan data knowledgebase merupakan proses yang sama dengan proses yang sebelumnya telah dijelaskan pada pengelolaan data pengguna dan pengelolaan data pesan, perbedaan hanyalah pada langkah dan ketentuan yang harus dilakukan dalam mengelola data artikel dan data istilah. Secara umum proses yang ada adalah tambah, edit, ubah status dan hapus data kategori artikel maupun data artikel serta data istilah.

Pada proses buat kategori dilakukan pembuatan atau penambahan kategori artikel baru. Pada gambar 5.48 dapat dilihat daftar kategori artikel knowledgebase.

The screenshot shows a web application interface for managing knowledgebase categories. At the top, there are two tabs: "Menu Manajemen Knowledgebase" and "Daftar Kategori Knowledgebase". Below the tabs are four action buttons: "Tambah", "Edit", "Ubah Status", and "Hapus". The main content is a table with the following data:

ID	Kategori	Jumlah Artikel	Aktif
7	Baru baru baru baru	1	✓
6	Harga Produk segala sesuatu yang berhubungan dengan harga produk kami	0	✓
5	Produk Segala sesuatu yang berhubungan dengan produk	1	✓
3	Login Segala sesuatu yang berhubungan dengan login	12	✓

At the bottom of the interface, there is a pagination control showing "Page 1 of 1" and a status indicator "Menampilkan Kategori 1 - 4 dari 4".

Gambar 5.48 Daftar kategori knowledgebase

Administrator dapat melakukan penambahan kategori artikel baru dengan menekan tombol tambah dan mengisi tambah form kategori yang ditunjukkan pada gambar 5.49.

This screenshot is similar to the previous one but highlights the "Tambah" button with a red box. Additionally, a text box containing "Buat Kategori Baru" is positioned above the "ID" column of the table.

ID	Kategori	Jumlah Artikel
7	Baru baru baru baru	1
6	Harga Produk segala sesuatu yang berhubungan dengan harga produk kami	0
5	Produk Segala sesuatu yang berhubungan dengan produk	1
3	Login Segala sesuatu yang berhubungan dengan login	12

Gambar 5.49 Menekan tombol Tambah untuk membuat kategori artikel baru

Setelah mengisi form tambah kategori baru maka Administrator dapat melakukan penyimpanan data kategori baru dengan menekan tombol simpan (Gambar 5.50).

The screenshot shows a web application window titled "Tambah Kategori" from "WEB Customer Support System @ 2008". The main content area is labeled "Tambah Kategori Knowledgebase". It features three input fields: "Nama" containing "Toko", "Deskripsi" containing "Contoh Kategori 1 : Toko", and "Urutan / Sekuen" containing "5". Below the fields are three buttons: "Simpan", "Reset", and "Batal". A tooltip labeled "Simpan Kategori" is positioned over the "Simpan" button.

Gambar 5.50 Form tambah kategori baru

Setelah mengisi form tambah kategori artikel baru maka, kategori akan tampil pada Grid Panel daftar kategori artikel. Selain itu Administrator dapat melakukan perubahan data kategori jika terdapat kesalahan pada data kategori artikel tertentu (Gambar 5.51).

The screenshot shows a web application window titled "Edit Kategori" from "WEB Customer Support System @ 2008". The main content area is labeled "Edit Kategori Knowledgebase". It features three input fields: "Nama" containing "Toko", "Deskripsi" containing "Contoh Kategori 1 : Toko (edit)", and "Urutan / Sekuen" containing "5". Below the fields are three buttons: "Update", "Reset", and "Batal". A tooltip labeled "Update Kategori" is positioned over the "Update" button.

Gambar 5.51 Form edit dan update data kategori

Untuk proses ubah status kategori dapat dilakukan dengan memilih kategori dan menekan tombol ubah status (Gambar 5.52).

ID	Kategori	Jumlah Artikel	Aktif
7	Baru <small>Kategori Baru Baru...</small>	1	✓
8	Toko <small>Contoh Kategori 1 - Toko (edit)</small>	0	✗
6	Harga Produk <small>Segala sesuatu yang berhubungan dengan harga produk kami</small>	0	✓

Gambar 5.52 Menekan tombol Ubah status untuk mengubah status kategori artikel

Status kategori artikel dan status pada artikel dimaksudkan untuk mengubah tampilan daftar artikel yang muncul dan tersedia pada daftar kategori artikel dan artikel web *Customer Support* sisi pelanggan.

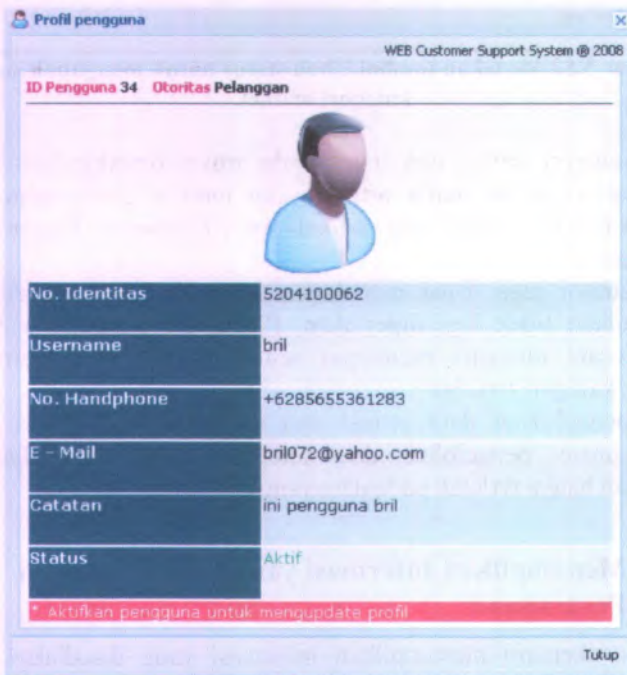
Administrator juga dapat menghapus kategori jika kategori yang dimaksudkan tidak lagi diperlukan. Proses penghapusan kategori akan secara otomatis menghapus seluruh artikel yang termasuk kedalam kategori tersebut.

Untuk pengelolaan data artikel dan data istilah dasarnya sama dengan proses pengelolaan data kategori artikel knowledgebase. Perbedaan hanya terletak pada data yang diubah.

5.4.5 Menampilkan Informasi yang Disediakan oleh Perusahaan

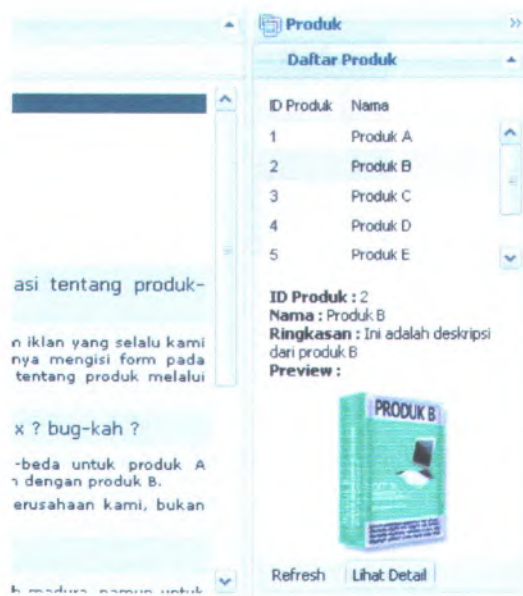
Uji coba skenario menampilkan informasi yang disediakan oleh perusahaan dibagi menjadi dua dimana informasi yang disediakan aplikasi web *Customer Support* dapat berupa tampilan halaman statis dan halaman dinamis. Untuk informasi yang bersifat umum seperti profil perusahaan, FAQ dan kontak maka digunakan halaman-halaman statis yang diload pada kontrol window atau tab Ext.

Untuk halaman yang menampilkan informasi tertentu berdasarkan pilihan informasi yang diinginkan oleh pengguna maka digunakan halaman dinamis yang di-load pada kontrol Ext. Untuk melihat informasi tentang profil pelanggan maka pengguna dapat memilih profil yang akan dilihat detailnya maka sistem akan menampilkan window dimana yang seperti telah dijelaskan pada bab 4, window Ext akan me-load halaman *Java Server Pages* dengan mem-parsing parameter-parameter tertentu.



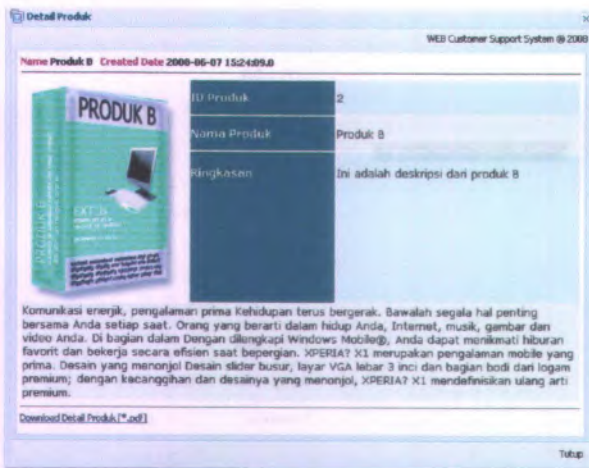
Gambar 5.53 Window detail profil pengguna

Hal yang sama dapat dilakukan pengguna terhadap informasi atau data produk yang ingin dilihat oleh pengguna.



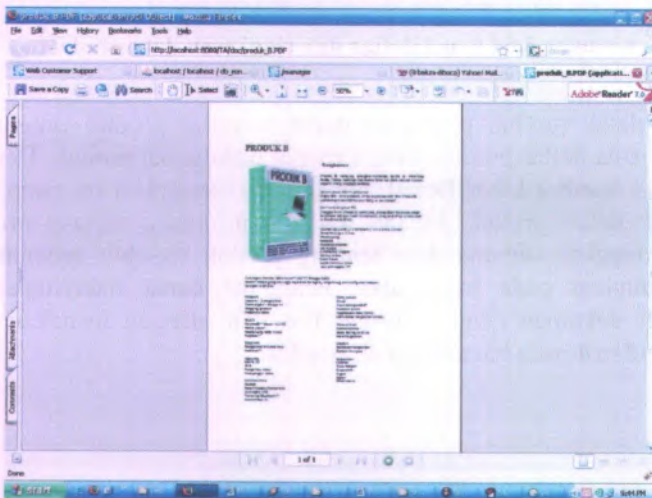
Gambar 5.54 Panel daftar dan ringkasan data produk

Untuk detail produk pengguna dapat memilih produk yang akan dilihat pada daftar produk yang terdapat pada panel produk. Dengan menekan tombol Lihat Detail, kemudian sistem akan menampilkan window detail produk dimana terdapat informasi tentang produk secara lengkap dimana data tentang produk tersebut sebelumnya telah disimpan pada basis data. Pengguna dapat menyimpannya menjadi dokumen dengan format file PDF dengan menekan link yang terdapat pada bagian bawah window.



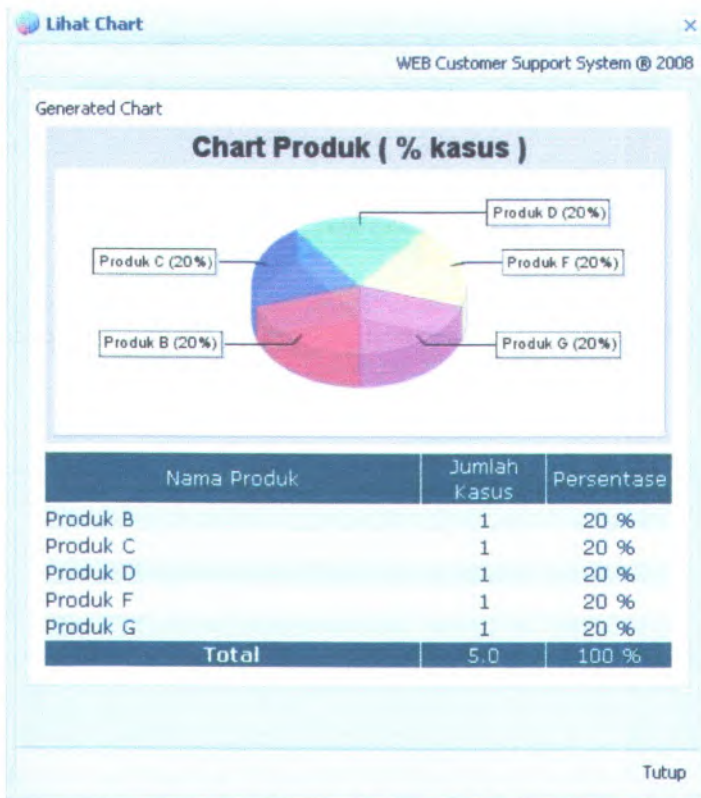
Gambar 5.55 Window Detail data produk

Pengguna dapat memilih link dibagian bawah pada window detail data produk untuk mendownload detail data produk (gambar 5.55).

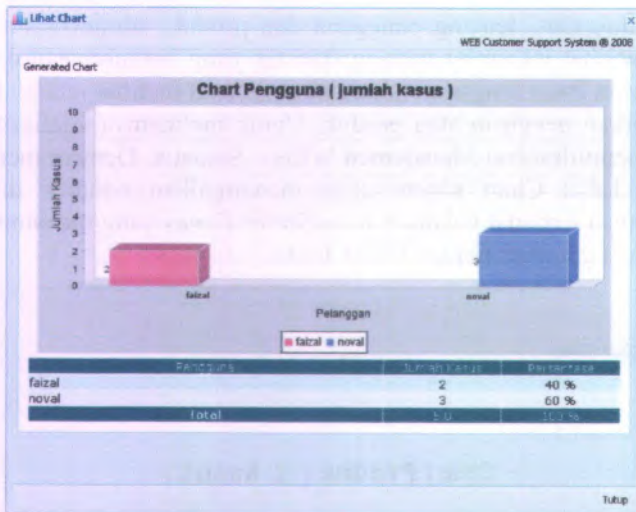


Gambar 5.56 Dokumen detail data produk

Selain informasi tentang pengguna dan produk, administrator juga dapat melihat informasi tentang statistik yang berhubungan dengan kasus web. Pada fungsi ini administrator dapat melihat jumlah kasus berdasarkan pengirim atau produk. Untuk melihatnya administrator dapat memilih menu Manajemen WEB – Statistik. Dengan menekan tombol Lihat Chart sistem akan menampilkan window dimana didalamnya terdapat halaman *Java Server Pages* yang menampilkan chart hasil generate library JFreeChart.

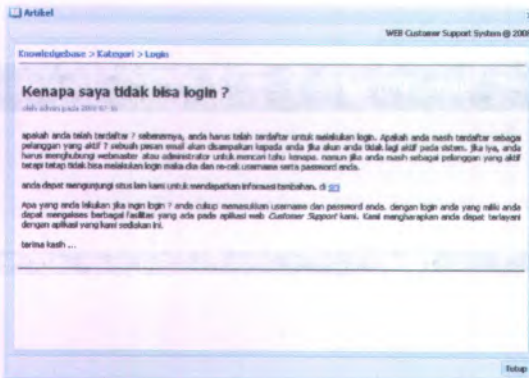


Gambar 5.57 Chart produk



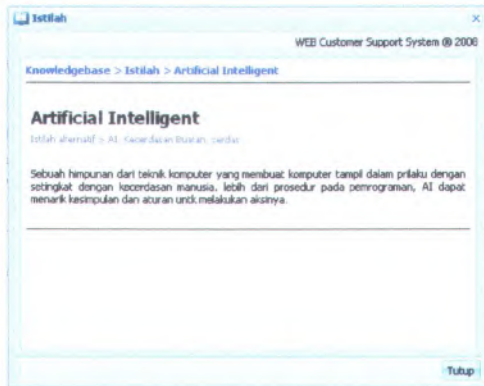
Gambar 5.58 Chart pengguna

Pengguna aplikasi web *Customer Support* juga dapat melihat artikel yang disediakan oleh manajemen perusahaan seperti pada gambar 5.59.



Gambar 5.59 Window baca artikel pada menu knowledgebase

Kemudian, daftar istilah merupakan daftar istilah yang disediakan oleh perusahaan ketika terdapat istilah yang secara umum tidak dimengerti oleh pelanggan. Gambar 5.60 merupakan tampilan window istilah.



Gambar 5.60 Window baca istilah pada menu knowledgebase

5.5 Perbandingan Proses

Tidak dapat diukur dengan jelas berapa perbedaan waktu proses antara sebelum adanya aplikasi web *Customer Support* ini dengan sesudah adanya aplikasi ini. Hal ini dikarenakan pada tugas akhir ini aplikasi dijalankan pada area lokal yang tidak terhubung dengan jaringan internet. Selain itu digunakan uji coba dilakukan pada data-data dummy atau rekayasa. Dengan adanya aplikasi web *Customer Support* ini, manajemen perusahaan dan pelanggan dapat dengan mudah mengelola data pesan permasalahan. Selain itu aplikasi dirancang diharapkan dapat memberi layanan *self-service* kepada pelanggan dimana data dari informasi yang akan ditampilkan dapat dikelola oleh pihak manajemen perusahaan secara mudah.

Kendala yang dihadapi setelah proses implementasi dilakukan adalah masih adanya pelanggan yang tidak terbiasa atau bahkan tidak bisa menggunakan sarana internet dan aplikasi WAP, sehingga

dibutuhkan sosialisasi dan pembelajaran mengenai teknologi tersebut kepada pelanggan.

BAB VI KESIMPULAN DAN SARAN

Bagian ini berisi kesimpulan dari seluruh proses pengerjaan Tugas Akhir beserta saran untuk proses pengembangan selanjutnya.

6.1 Kesimpulan

Berdasarkan hasil tugas akhir yang dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut :

- Rancangan dan desain aplikasi web *Customer Support* berhasil dibuat dengan menggunakan teknologi *Java Server Pages* dan *JavaScript library Ext* yang merupakan implementasi dari konsep RIA. Hal itu terlihat dari fungsi-fungsi yang ada dalam aplikasi web *Customer Support* dapat berjalan dengan baik. Fungsi-fungsi itu meliputi:
 - Fungsi pengelolaan data pelanggan.
 - Fungsi untuk mengelola data permasalahan dari masing-masing *touchpoint* yang digunakan.
 - Fungsi untuk mengelola dan menampilkan informasi atau data yang disediakan perusahaan.
- Masih terdapat beberapa kelemahan pada sisi fungsi atau fitur sistem aplikasi yang dikembangkan, salah satunya tidak adanya fungsi yang membuat pihak perusahaan mengendalikan permasalahan yang sedang dalam proses penyelesaian. Permasalahan yang sedang ada pada tahap penyelesaian tidak dapat diketahui oleh pengguna status atau keadaannya. Status yang dimaksud adalah seperti, sedang dikerjakan oleh siapa, bagaimana progressnya, catatan-catatan dan berbagai informasi lainnya yang berhubungan dengan keadaan permasalahan ketika sedang dalam tahap proses penyelesaian.

- Implementasi teknologi baru pada suatu aplikasi dapat menimbulkan berbagai variasi cara proses pengolahan data dalam proses bisnis yang ditangani. Dengan kemudahan yang diberikan, pemrogram atau pengembang dapat mendesain bentuk aplikasi yang bermacam-macam selama tidak keluar dari kebutuhan proses bisnis yang diinginkan.

Contohnya pada bahasa *server-side programming* yang digunakan, pengembang aplikasi dapat menggunakan berbagai bahasa pemrograman yang dikuasainya. Selain itu bentuk tampilan aplikasi, seperti menu dan konten. Pengembang dapat mendesain sesuai dengan kenyamanan pengguna secara lebih mudah tanpa mengubah proses bisnis yang harus dapat dilakukan sistem aplikasi.

6.2 Saran

Beberapa hal yang diharapkan dapat dikembangkan pada masa mendatang adalah sebagai berikut :

- Mengintegrasikan sistem aplikasi *Customer Support* secara menyeluruh dengan aplikasi CRM lain karena sistem aplikasi *Customer Support* merupakan bentuk CRM operasional yang mendukung aktifitas CRM analitik.
- Pengembangan fungsi-fungsi dan fitur-fitur pada aplikasi yang disesuaikan dengan kebutuhan perusahaan.
- Penggunaan *JavaScript library Ext* dapat memudahkan kita dalam mengembangkan aplikasi berbasis web. Namun masalah yang muncul adalah kecepatan akses yang berkurang. Hal ini dikarenakan banyak file *JavaScript* yang di-load. Oleh karena itu diharapkan dalam pengembangan selanjutnya dapat dilakukan optimasi terhadap *JavaScript library Ext* seperti penggabungan dengan *framework* tertentu sehingga diperoleh kecepatan akses yang maksimal.

DAFTAR PUSTAKA

- [1] Anonymous, 2002. **Introducing JSON**. <<http://json.org>>.
- [2] Anonymous, August 2003. **Technical Support**. <http://en.wikipedia.org/wiki/Technical_support>.
- [3] Anonymous, March 2008. **ExtJs**. <<http://www.extjs.com>>.
- [4] Anonymous, June 2007. **Short message service**. <http://en.wikipedia.org/wiki/Short_message_service>.
- [5] Anonymous, October 2005. **Wireless Application Protocol**. <http://en.wikipedia.org/wiki/Wireless_Application_Protocol>.
- [6] Boggs, Wendy dan Boggs, Michael, August 2002. **Mastering UML with Rational Rose 2002**. SYBEX Inc.
- [7] Dyche, Jill. 2002. **The CRM Handbook: A Business Guide to Customer Relationship Management**. Addison-Wesley.
- [8] Jusak. Nopember 2007. **Kreasi Situs Mobile Internet dengan xHTML MP**. Prestasi Pustaka.

Halaman ini sengaja dikosongkan.

LAMPIRAN A

Lampiran A - 1 Naratif Use Case Login

<i>Use Case name</i> : Login	<i>ID</i> : UC.01	<i>Importance level</i> : High
<i>Primary actor</i> : Pengguna	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :		
Pengguna – ingin melakukan login pada sistem sehingga dapat menggunakan fasilitas-fasilitas yang sesuai dengan hak akses masing-masing pengguna.		
<i>Brief description</i> : Use case ini digunakan untuk melakukan validasi terhadap Pengguna yang ingin menggunakan sistem aplikasi web dan WAP <i>Customer Support</i> .		
<i>Trigger</i> : Pengguna ingin mengakses fungsi-fungsi dalam aplikasi.		
<i>Type</i> : Eksternal		
<i>Relationships</i> :		
<i>Association</i> : Pengguna		
<i>Include</i> : -		
<i>Extend</i> : -		
<i>Generalization</i> : -		
<i>Normal flow of events</i> :		
1. Pengguna masuk ke halaman login.		
2. Pengguna memasukkan data pada <i>field username</i> dan <i>field password</i> .		
3. Pengguna menekan tombol <i>login</i> .		
4. Sistem melakukan validasi terhadap data.		
5. Jika data yang dimasukkan sesuai, maka <i>login accepted</i> .		
6. Sistem menampilkan halaman utama sesuai dengan otoritas yang dimiliki pengguna.		
<i>Subflows</i> :		
--		
<i>Alternate/exceptional flows</i> :		
4a Jika <i>field username</i> atau <i>field password</i> kosong, sistem akan kembali ke halaman login menampilkan pesan “username / password salah”.		
4b Jika data <i>username</i> dan <i>password</i> yang dimasukkan pengguna tidak sama dengan <i>username</i> dan <i>password</i> di database, sistem menampilkan pesan kesalahan “ username / password salah”, Sistem kembali pada halaman login.		



Lampiran A - 2 Naratif Use Case Ubah Password

<i>Use Case name</i> : Ubah Password	<i>ID</i> : UC.02	<i>Importance level</i> : High
<i>Primary actor</i> : Pengguna	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :		
Pengguna – ingin mengganti <i>password</i> untuk login yang dimiliki melalui aplikasi web <i>Customer Support</i> .		
<i>Brief description</i> : Use case ini digunakan untuk mengganti <i>password</i> pengguna melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger</i> : Terdapat <i>password</i> lama yang ingin diganti.		
<i>Type</i> : Eksternal		
<i>Relationships</i> :		
<i>Association</i> : Pengguna		
<i>Include</i> : -		
<i>Extend</i> : -		
<i>Generalization</i> : -		
<i>Normal flow of events</i> :		
<ol style="list-style-type: none"> 1. Pengguna menuju form "Ganti Password". 2. Pengguna memasukkan data <i>password</i> lama pada <i>field</i> "Password lama". 3. Pengguna memasukkan data <i>password</i> baru pada <i>field</i> "Password baru". 4. Pengguna memasukkan <i>password</i> baru pada <i>field</i> "Konfirmasi password baru" untuk tujuan konfirmasi kebenaran <i>password</i> baru. 5. Pengguna menekan tombol "Ganti". 6. Sistem mengubah data <i>password</i> lama dengan data <i>password</i> yang baru dan menampilkan pesan status bahwa proses ganti <i>password</i> telah berhasil dilakukan. 		
<i>Subflows</i> :		
--		
<i>Alternate/exceptional flows</i> :		
<ol style="list-style-type: none"> 6a Jika <i>field-field</i> masukan yang terdapat pada form "Ganti Password" tidak diisi maka sistem tidak akan melakukan proses ganti ubah password dan menampilkan pesan peringatan. 6b Jika data <i>password</i> lama pada basis data tidak sama dengan data <i>password</i> lama yang dimasukan pada form "Ganti Password", maka sistem akan menampilkan pesan kesalahan dan proses mengubah password tidak diproses. 		

Lampiran A - 3 Naratif Use Case Ubah Profil

<i>Use Case name :</i> Ubah Profil	<i>ID :</i> UC.03	<i>Importance level :</i> High
<i>Primary actor :</i> Pengguna	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Pengguna – ingin mengubah data profilnya masing-masing melalui aplikasi web <i>Customer Support</i> .		
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat mengubah data profilnya sendiri yang tidak sesuai melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat data profil yang tidak sesuai atau perlu diperbarui oleh pengguna sendiri.		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Pengguna		
<i>Include :</i> -		
<i>Extend :</i> -		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Pengguna menuju form “Edit Profil”. 2. Pengguna mengubah data profil yang dimilikinya dengan memasukkan data profil terkini yang berkenaan. 3. Pengguna menekan tombol “Simpan”. 4. Sistem melakukan penyimpanan data profil yang telah diedit atau diubah pengguna. 5. Sistem menampilkan form “Edit Profil” dengan data profil yang telah di update. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
<ol style="list-style-type: none"> 3a Jika data profil masukkan pada form tidak valid maka sistem tidak akan melakukan proses ubah profil dan menampilkan pesan peringatan. 4a Jika data profil khususnya data no. identitas yang dimasukkan merupakan no. identitas yang telah ada pada basis data maka sistem tidak akan melakukan proses ubah profil dan menampilkan pesan kesalahan. 		

Lampiran A - 4 Naratif Use Case Lihat Informasi / Berita

<i>Use Case name</i> :	Lihat Informasi / Berita	<i>ID</i> : UC.04	<i>Importance level</i> : Medium
<i>Primary actor</i> :	Pengguna	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :			
Pengguna – ingin melihat informasi / berita terbaru yang dipublikasikan perusahaan.			
<i>Brief description</i> : Use case ini digunakan untuk membuat pengguna dapat melihat informasi atau berita yang dipublikasikan perusahaan.			
<i>Trigger</i> : Terdapat informasi atau berita terbaru yang diupdate oleh perusahaan.			
<i>Type</i> : Eksternal			
<i>Relationships</i> :			
<i>Association</i> : Pengguna			
<i>Include</i> : -			
<i>Extend</i> : -			
<i>Generalization</i> : -			
<i>Normal flow of events</i> :			
<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman utama aplikasi web <i>Customer Support</i>. 2. Pengguna menekan tombol “Refresh” pada panel “Informasi / Berita”. 3. Sistem menampilkan data informasi / berita informasi yang terdapat pada basis data. 			
<i>Subflows</i> :			
--			
<i>Alternate/exceptional flows</i> :			
-- --			

Lampiran A - 5 Naratif Use Case Lihat Ringkasan Data Produk

<i>Use Case name :</i>	Lihat Ringkasan Data Produk	<i>ID :</i> UC.05	<i>Importance level :</i> Medium
<i>Primary actor :</i> Pengguna	<i>Use case Type :</i>		
<i>Stakeholders and interest :</i>			
Pengguna – ingin melihat ringkasan data produk yang dimiliki perusahaan.			
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat melihat daftar produk beserta ringkasan data produk yang dimiliki perusahaan melalui aplikasi web dan WAP <i>Customer Support</i> .			
<i>Trigger :</i> Terdapat produk tertentu yang ingin dilihat datanya secara ringkas.			
<i>Type :</i> Eksternal			
<i>Relationships :</i>			
<i>Association :</i> Pengguna			
<i>Include :</i> -			
<i>Extend :</i> -			
<i>Generalization :</i> -			
<i>Normal flow of events :</i>			
<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman utama. 2. Pengguna memilih salah satu produk pada halaman atau panel “Daftar Produk”. 3. Sistem menampilkan data produk pada halaman atau panel ringkasan data produk berdasarkan produk yang dipilih pengguna pada daftar produk. 			
<i>Subflows :</i>			
--			
<i>Alternate/exceptional flows :</i>			
-- --			

Lampiran A - 6 Naratif Use Case Lihat Detail Data Produk

<i>Use Case name :</i> Lihat Detail Data Produk	<i>ID :</i> UC.06	<i>Importance level :</i> Medium
<i>Primary actor :</i> Pengguna	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Pengguna – ingin melihat detail data produk dimana data yang ditampilkan merupakan deskripsi lengkap dari produk yang dipilih.		
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat melihat data produk secara detail melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat produk tertentu yang ingin dilihat datanya secara lengkap. .		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Pengguna		
<i>Include :</i> -		
<i>Extend :</i> UC.05		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Pengguna memilih salah satu produk pada panel “Daftar Produk”. 2. Pengguna menekan tombol “Lihat Detail”. 3. Sistem menampilkan data produk yang dipilih sebelumnya pada panel “Daftar Produk” secara lengkap dan detail pada sebuah <i>window pop-up</i>. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
-- --		

Lampiran A - 7 Naratif Use Case Download Detail Data Produk

<i>Use Case name :</i>	Download Detail Data Produk	<i>ID :</i> UC.07	<i>Importance level :</i> Medium
<i>Primary actor :</i> Pengguna	<i>Use case Type :</i>		
<i>Stakeholders and interest :</i>			
Pengguna – ingin menyimpan detail data produk tertentu kedalam bentuk dokumen.			
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat melakukan download detail data produk ke dalam bentuk dokumen melalui aplikasi web <i>Customer Support</i> .			
<i>Trigger :</i> Terdapat kebutuhan detail data produk kedalam format dokumen.			
<i>Type :</i> Eksternal			
<i>Relationships :</i>			
<i>Association :</i> Pengguna			
<i>Include :</i> -			
<i>Extend :</i> UC.06			
<i>Generalization :</i> -			
<i>Normal flow of events :</i>			
<ol style="list-style-type: none"> 1. Pengguna memilih salah satu produk pada panel “Daftar Produk”. 2. Pengguna menekannya tombol “Lihat Detail”. 3. Sistem menampilkan detail data produk pada <i>window pop-up</i>. 4. Pengguna menekan link yang terdapat pada bagian bawah <i>window pop-up</i>. 5. Sistem akan membuka tab baru pada internet browser dimana tab tersebut membuka file berisi data detail produk yang dipilih pengguna dengan format file PDF. 			
<i>Subflows :</i>			
--			
<i>Alternate/exceptional flows :</i>			
-- --			

Lampiran A - 8 Naratif Use Case Baca Istilah

<i>Use Case name :</i> Baca Istilah	<i>ID :</i> UC.08	<i>Importance level :</i> Medium
<i>Primary actor :</i> Pengguna	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Pengguna – ingin mengetahui istilah-istilah yang berhubungan dengan produk dimana istilah-istilah tersebut tidak diketahui olehnya.		
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat melihat daftar dan membaca istilah yang disediakan oleh perusahaan melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat istilah yang tidak diketahui oleh pengguna.		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Pengguna		
<i>Include :</i> -		
<i>Extend :</i> -		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Pengguna menuju ke daftar istilah pada menu “Knowledgebase”. 2. Pengguna memilih salah satu istilah yang ada pada daftar istilah. 3. Pengguna menekan tombol “Baca Istilah”. 4. Sistem menampilkan window baru dimana pada window tersebut menampilkan definisi atau deskripsi dari istilah yang dipilih sebelumnya oleh pengguna. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
3a Jika pengguna tidak memilih salah satu istilah pada daftar istilah maka, sistem akan menampilkan pesan peringatan.		

Lampiran A - 9 Naratif Use Case Baca Artikel

<i>Use Case name :</i>	Baca Artikel	<i>ID :</i> UC.09	<i>Importance level :</i> Medium
<i>Primary actor :</i> Pengguna	<i>Use case Type :</i>		
<i>Stakeholders and interest :</i>			
Pengguna – ingin membaca artikel yang disediakan oleh manajemen perusahaan guna membantu pengguna dimana artikel tersebut berhubungan dengan produk perusahaan.			
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat membaca artikel tertentu yang disediakan perusahaan melalui aplikasi web <i>Customer Support</i> .			
<i>Trigger :</i> Terdapat artikel yang perlu diketahui dan dapat membantu pengguna.			
<i>Type :</i> Eksternal			
<i>Relationships :</i>			
<i>Association :</i> Pengguna			
<i>Include :</i> -			
<i>Extend :</i> -			
<i>Generalization :</i> -			
<i>Normal flow of events :</i>			
<ol style="list-style-type: none"> 1. Pengguna menuju daftar kategori artikel pada menu “Knowledgebase”. 2. Pengguna memilih salah satu kategori artikel. 3. Pengguna menekan tombol “Buka Kategori”. 4. Sistem menampilkan daftar artikel yang termasuk pada kategori artikel yang dipilih sebelumnya. 5. Pengguna memilih salah satu artikel pada daftar artikel dan menekan tombol “Baca Artikel”. 6. Sistem menampilkan window baru dimana pada window terdapat isi dari artikel yang dipilih sebelumnya. 			
<i>Subflows :</i>			
--			
<i>Alternate/exceptional flows :</i>			
<ol style="list-style-type: none"> 3a Jika pengguna tidak memilih salah satu kategori pada daftar kategori artikel maka sistem akan menampilkan pesan peringatan. 4a Jika tidak terdapat artikel pada kategori artikel yang dipilih maka sistem akan menampilkan pesan “tidak ada artikel” pada tampilan daftar artikel. 5a Jika pengguna tidak memilih salah satu artikel pada daftar artikel maka sistem akan menampilkan pesan peringatan. 			

Lampiran A - 10 Naratif Use Case Tambah Pengguna

<i>Use Case name :</i> Tambah Pengguna	<i>ID :</i> UC.10	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Administrator / Manajemen – ingin menambah pengguna baru ke dalam sistem aplikasi <i>Customer Support</i>		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan penambahan pengguna dengan memasukkan data profil pengguna melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Belum ada pengguna tersebut pada sistem aplikasi <i>Customer Support</i> .		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Administrator / Manajemen		
<i>Include :</i> -		
<i>Extend :</i> -		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju form “Tambah Pengguna”. 2. Administrator / Manajemen memasukkan data profil pengguna baru pada form “Tambah Pengguna”. 3. Pengguna menekan tombol “Simpan”. 4. Sistem menambah data pengguna baru pada basis data berdasar data masukkan pada form “Tambah Pengguna”. 5. Sistem menampilkan pesan status proses tambah pengguna berhasil. 6. Sistem menampilkan form tambah pengguna kembali yang <i>field-field</i> masukannya telah kosong. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
<ol style="list-style-type: none"> 3a Jika data profil masukkan pada form tidak valid maka sistem tidak akan melakukan proses tambah pengguna dan menampilkan pesan peringatan. 4b Jika data profil khususnya data <i>username</i> dan atau no. identitas yang dimasukkan merupakan <i>username</i> dan atau no. identitas yang telah ada pada basis data maka sistem tidak akan melakukan proses tambah pengguna dan menampilkan pesan kesalahan. 		

Lampiran A - 11 Naratif Use Case Lihat Profil Pengguna

<i>Use Case name</i> :	Lihat Profil Pengguna	<i>ID</i> : UC.11	<i>Importance level</i> : Medium
<i>Primary actor</i> :	Administrator / Manajemen	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :			
Administrator / Manajemen – ingin melihat data profil pengguna, dalam hal ini data profil pelanggan.			
<i>Brief description</i> : Use case ini digunakan untuk melihat detail data profil pengguna yaitu pelanggan yang terdaftar pada sistem aplikasi melalui aplikasi web <i>Customer Support</i> .			
<i>Trigger</i> : Terdapat pengguna atau pelanggan yang data profilnya perlu dilihat.			
<i>Type</i> : Eksternal			
<i>Relationships</i> :			
<i>Association</i> : Administrator / Manajemen			
<i>Include</i> : -			
<i>Extend</i> : -			
<i>Generalization</i> : -			
<i>Normal flow of events</i> :			
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju form pencarian pengguna. 2. Administrator / Manajemen menekan tombol "Cari". 3. Sistem menampilkan hasil pencarian daftar pengguna pada panel "Hasil Pencarian Pengguna". 4. Administrator / Manajemen memilih salah satu pengguna pada panel "Hasil Pencarian Pengguna". 5. Administrator / Manajemen menekan tombol "Lihat Profil". 6. Sistem menampilkan data profil pengguna yang dipilih pada suatu <i>window pop-up</i>. 			
<i>Subflows</i> :			
--			
<i>Alternate/exceptional flows</i> :			
5a Jika Administrator / Manajemen tidak memilih salah satu pengguna pada panel "Hasil Pencarian Pengguna" maka sistem akan menampilkan pesan peringatan.			

Lampiran A - 12 Naratif Use Case Ubah Profil Pengguna

<i>Use Case name :</i> Ubah Profil Pengguna	<i>ID :</i> UC.12	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Administrator / Manajemen – ingin mengubah data pengguna yang terdaftar pada sistem aplikasi.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan data profil pengguna aplikasi <i>Customer Support</i> dalam hal ini adalah data profil pelanggan.		
<i>Trigger :</i> Terdapat data profil pengguna yang dianggap oleh Administrator / Manajemen tidak sesuai.		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Administrator / Manajemen		
<i>Include :</i> -		
<i>Extend :</i> -		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Pengguna”. 2. Administrator / Manajemen memilih salah satu pengguna pada panel “Daftar Pengguna”. 3. Administrator / Manajemen menekan tombol “Edit Pengguna”. 4. Sistem menampilkan form “Edit Pengguna” pada <i>window pop-up</i> dimana form berisi data profil pengguna yang ada pada basis data. 5. Administrator / Manajemen mengubah data profil dan menekan tombol “Update”. 6. Sistem menyimpan data profil pengguna yang telah diubah dan menampilkan status bahwa proses ubah profil pengguna berhasil. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
<ol style="list-style-type: none"> 3a Jika Administrator / Manajemen tidak memilih salah satu pengguna pada panel “Daftar Pengguna” maka sistem akan menampilkan pesan peringatan. 5a Jika form “Edit Pengguna” tidak valid maka sistem tidak akan melakukan proses ubah profil dan menampilkan pesan peringatan. 6b Jika data profil pengguna telah ada khususnya data <i>username</i> dan atau <i>no. identitas</i> maka sistem tidak akan melakukan proses ubah profil dan menampilkan pesan kesalahan. 		

Lampiran A - 13 Naratif Use Case Ubah Status Pengguna

<i>Use Case name :</i> Ubah Status Pengguna	<i>ID :</i> UC.13	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin mengubah status pengguna menjadi aktif atau tidak aktif.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan status pengguna menjadi aktif atau tidak aktif melalui aplikasi web <i>Customer Support</i> . Pengguna yang mempunyai status tidak aktif maka pengguna tersebut tidak dapat mengakses sistem.		
<i>Trigger :</i> Terdapat pengguna yang akunnnya tidak lagi aktif atau sebaliknya, perlu diaktifkan. <i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Pengguna”. 2. Administrator / Manajemen memilih salah satu pengguna pada panel “Daftar Pengguna”. 3. Administrator / Manajemen menekan tombol “Non-Aktifkan Pengguna”. 4. Sistem melakukan perubahan data status pengguna menjadi tidak aktif. 5. Sistem menampilkan pesan status proses perubahan status pengguna berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> <ol style="list-style-type: none"> 3a Jika Administrator / Manajemen tidak memilih salah satu pengguna pada panel “Daftar Pengguna” maka sistem akan menampilkan pesan peringatan. 4a Untuk mengubah status pengguna yang aktif tidak aktif menjadi aktif maka Administrator / Manajemen memilih pengguna pada form cari pelanggan kemudian memilih pelanggan dan menekan tombol aktifkan pengguna. 		

Lampiran A - 14 Naratif Use Case Tambah Kategori Artikel

<i>Use Case name</i> :	Tambah Kategori Artikel	<i>ID</i> : UC.14	<i>Importance level</i> : High
<i>Primary actor</i> :	Administrator / Manajemen	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :			
Administrator / Manajemen – ingin melakukan penambahan kategori artikel baru.			
<i>Brief description</i> : Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan penambahan kategori artikel baru melalui aplikasi web <i>Customer Support</i> .			
<i>Trigger</i> : Terdapat kategori artikel baru yang belum terdapat sebelumnya pada basis data.			
<i>Type</i> : Eksternal			
<i>Relationships</i> :			
<i>Association</i> : Administrator / Manajemen			
<i>Include</i> : -			
<i>Extend</i> : -			
<i>Generalization</i> : -			
<i>Normal flow of events</i> :			
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel "Daftar Kategori Knowledgebase". 2. Administrator / Manajemen menekan tombol "Tambah". 3. Sistem menampilkan form "Tambah Kategori" pada suatu <i>window pop-up</i>. 4. Administrator / Manajemen mengisi form "Tambah Kategori" dengan data kategori baru yang diinginkan. Data kategori antara lain : Nama, Deskripsi, Urutan/Sekuen. 5. Administrator / Manajemen menekan tombol "Simpan". 6. Sistem melakukan penambahan data kategori artikel baru pada basis data dan menampilkan pesan status penambahan kategori artikel baru berhasil dilakukan. 			
<i>Subflows</i> :			
--			
<i>Alternate/exceptional flows</i> :			
5a Jika form "Tambah Kategori" tidak valid maka sistem tidak akan melakukan proses tambah kategori artikel dan menampilkan pesan peringatan.			

Lampiran A - 15 Naratif Use Case Edit Kategori Artikel

<i>Use Case name</i> :	Edit Kategori Artikel Kb	<i>ID</i> : UC.15	<i>Importance level</i> : High
<i>Primary actor</i> :	Administrator / Manajemen	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :			
Administrator / Manajemen – ingin melakukan perubahan data kategori artikel yang sudah ada.			
<i>Brief description</i> : Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan data kategori artikel yang sudah ada melalui aplikasi web <i>Customer Support</i> .			
<i>Trigger</i> : terdapat data kategori artikel yang tidak sesuai atau tidak benar.			
<i>Type</i> : Eksternal			
<i>Relationships</i> :			
<i>Association</i> : Administrator / Manajemen			
<i>Include</i> : -			
<i>Extend</i> : -			
<i>Generalization</i> : -			
<i>Normal flow of events</i> :			
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Kategori Knowledgebase”. 2. Administrator / Manajemen memilih salah satu kategori artikel yang ada pada panel “Daftar Kategori Knowledgebase”. 3. Administrator / Manajemen menekan tombol “Edit”. 4. Sistem menampilkan form “Edit Kategori” pada suatu <i>window pop-up</i> dimana form berisi data kategori artikel yang ada pada basis data. 5. Administrator / Manajemen mengubah data kategori yang lama dengan data kategori baru yang diinginkan. Data kategori yang dapat diubah antara lain : Nama, Deskripsi, Urutan/Sekuen. 6. Administrator / Manajemen menekan tombol “Update”. 7. Sistem melakukan penyimpanan data kategori artikel yang telah diupdate pada basis data dan menampilkan pesan status proses ubah kategori berhasil dilakukan. 			
<i>Subflows</i> :			
--			
<i>Alternate/exceptional flows</i> :			
<ol style="list-style-type: none"> 3a Jika Administrator / Manajemen tidak memilih salah satu kategori artikel yang ada pada panel “Daftar Kategori Knowledgebase” maka sistem akan menampilkan pesan peringatan. 6a Jika form “Edit Kategori” tidak valid maka sistem tidak akan melakukan proses ubah kategori dan menampilkan pesan peringatan. 			

Lampiran A - 16 Naratif Use Case Ubah Status Kategori Artikel

<i>Use Case name :</i> Ubah Status Kategori Artikel	<i>ID :</i> UC.16	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melakukan perubahan status kategori artikel.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan status kategori artikel melalui aplikasi web <i>Customer Support</i> . Kategori yang mempunyai status tidak aktif maka kategori tersebut tidak akan tampil pada daftar kategori aplikasi web <i>Customer Support</i> sisi pelanggan.		
<i>Trigger :</i> Terdapat kategori yang statusnya perlu diaktifkan atau di non-aktifkan.		
<i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Kategori Knowledgebase”. 2. Administrator / Manajemen memilih salah satu kategori artikel yang ada pada panel “Daftar Kategori Knowledgebase”. 3. Administrator / Manajemen menekan tombol “Ubah Status”. 4. Sistem melakukan perubahan data status kategori artikel. 5. Sistem menampilkan pesan status proses perubahan status kategori artikel berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> <ol style="list-style-type: none"> 3a Jika Administrator / Manajemen tidak memilih salah satu kategori artikel yang ada pada panel “Daftar Kategori Knowledgebase” maka sistem akan menampilkan pesan peringatan. 4a Jika status kategori aktif maka, sistem akan mengubah status kategori artikel menjadi tidak aktif dan sebaliknya jika kategori artikel tidak aktif maka sistem akan mengubah status kategori artikel menjadi aktif. 		

Lampiran A - 17 Naratif Use Case Hapus Kategori Artikel

<i>Use Case name :</i> Hapus Kategori Artikel	<i>ID :</i> UC.17	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melakukan penghapusan data kategori artikel.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan penghapusan data kategori artikel melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat kategori artikel beserta artikel yang termasuk didalamnya yang tidak sesuai dan perlu dihapus. <i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel "Daftar Kategori Knowledgebase". 2. Administrator / Manajemen memilih salah satu kategori artikel yang ada pada panel "Daftar Kategori Knowledgebase". 3. Administrator / Manajemen menekan tombol "Hapus". 4. Sistem melakukan penghapusan data kategori artikel beserta data artikel yang termasuk dalam kategori artikel yang dihapus tersebut. 5. Sistem menampilkan pesan status proses hapus kategori artikel berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> 3a Jika Administrator / Manajemen tidak memilih salah satu kategori artikel yang ada pada panel "Daftar Kategori Knowledgebase" maka sistem akan menampilkan pesan peringatan.		

Lampiran A - 18 Naratif Use Case Tambah Artikel

<i>Use Case name :</i> Tambah Artikel	<i>ID :</i> UC.18	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Administrator / Manajemen – ingin melakukan penambahan artikel baru pada kategori tertentu.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan penambahan artikel baru pada kategori tertentu melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat artikel baru yang belum terdapat sebelumnya pada basis data.		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Administrator / Manajemen		
<i>Include :</i> -		
<i>Extend :</i> -		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Artikel Knowledgebase”. 2. Administrator / Manajemen menekan tombol “Tambah”. 3. Sistem menampilkan form “Tambah Artikel” pada suatu <i>window pop-up</i>. 4. Administrator / Manajemen mengisi form “Tambah Artikel” dengan data artikel baru yang diinginkan. Data artikel antara lain : Kategori, Judul, Isi Artikel. 5. Administrator / Manajemen menekan tombol “Simpan”. 6. Sistem melakukan penambahan data artikel baru pada basis data dan menampilkan pesan status penambahan artikel baru berhasil dilakukan. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
5a Jika form “Tambah Artikel” tidak valid maka sistem tidak akan melakukan proses tambah artikel dan menampilkan pesan peringatan.		

Lampiran A - 19 Naratif Use Case Edit Artikel

<i>Use Case name :</i> Edit Artikel	<i>ID :</i> UC.19	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melakukan perubahan data artikel yang sudah ada.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan data artikel yang sudah ada melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> terdapat data artikel yang tidak sesuai atau tidak benar.		
<i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Artikel Knowledgebase”. 2. Administrator / Manajemen memilih salah satu artikel yang ada pada panel “Daftar Artikel Knowledgebase”. 3. Administrator / Manajemen menekan tombol “Edit”. 4. Sistem menampilkan form “Edit Artikel” pada suatu <i>window pop-up</i> dimana form berisi data artikel yang ada pada basis data. 5. Administrator / Manajemen mengubah data artikel yang lama dengan data artikel baru yang diinginkan. 6. Administrator / Manajemen menekan tombol “Update”. 7. Sistem melakukan penyimpanan data artikel yang telah diupdate pada basis data dan menampilkan pesan status proses ubah artikel berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> <ol style="list-style-type: none"> 3a Jika Administrator / Manajemen tidak memilih salah satu artikel yang ada pada panel “Daftar Artikel Knowledgebase” maka sistem akan menampilkan pesan peringatan. 6a Jika form “Edit Artikel” tidak valid maka sistem tidak akan melakukan proses ubah artikel dan menampilkan pesan peringatan. 		

Lampiran A - 20 Naratif Use Case Ubah Status Artikel

<i>Use Case name :</i> Ubah Status Artikel	<i>ID :</i> UC.20	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melakukan perubahan status artikel.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan status artikel melalui aplikasi web <i>Customer Support</i> . Artikel yang mempunyai status tidak aktif maka artikel tersebut tidak akan tampil pada daftar artikel aplikasi web <i>Customer Support</i> sisi pelanggan.		
<i>Trigger :</i> Terdapat artikel yang statusnya perlu diaktifkan atau di non-aktifkan.		
<i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Artikel Knowledgebase”. 2. Administrator / Manajemen memilih salah satu artikel yang ada pada panel “Daftar Artikel Knowledgebase”. 3. Administrator / Manajemen menekan tombol “Ubah Status”. 4. Sistem melakukan perubahan data status artikel. 5. Sistem menampilkan pesan status proses perubahan status artikel berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> <ol style="list-style-type: none"> 3a Jika Administrator / Manajemen tidak memilih salah satu artikel yang ada pada panel “Daftar Artikel Knowledgebase” maka sistem akan menampilkan pesan peringatan. 4a Jika status artikel aktif maka, sistem akan mengubah status artikel menjadi tidak aktif dan sebaliknya jika artikel tidak aktif maka sistem akan mengubah status artikel menjadi aktif. 		

Lampiran A - 21 Naratif Use Case Hapus Artikel

<i>Use Case name :</i> Hapus Artikel	<i>ID :</i> UC.21	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melakukan penghapusan data artikel.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan penghapusan data artikel melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat artikel yang tidak sesuai dan perlu dihapus.		
<i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar Artikel Knowledgebase”. 2. Administrator / Manajemen memilih salah satu artikel yang ada pada panel “Daftar Artikel Knowledgebase”. 3. Administrator / Manajemen menekan tombol “Hapus”. 4. Sistem melakukan penghapusan data artikel. 5. Sistem menampilkan pesan status proses hapus artikel berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> 3a Jika Administrator / Manajemen tidak memilih salah satu artikel yang ada pada panel “Daftar Artikel Knowledgebase” maka sistem akan menampilkan pesan peringatan.		

Lampiran A - 22 Naratif Use Case Tambah Istilah

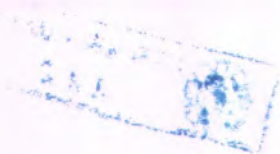
<i>Use Case name :</i> Tambah Istilah	<i>ID :</i> UC.22	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melakukan penambahan istilah baru..		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan penambahan data istilah baru melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat istilah baru yang belum terdapat sebelumnya pada basis data.		
<i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar istilah-istilah”. 2. Administrator / Manajemen menekan tombol “Tambah”. 3. Sistem menampilkan form “Tambah Istilah Baru” pada suatu <i>window pop-up</i>. 4. Administrator / Manajemen mengisi form “Tambah Istilah Baru” dengan data istilah baru yang diinginkan. Data artikel antara lain : Istilah, Alternatif 1, Alternatif 2, Alternatif 3 dan Definisi. 5. Administrator / Manajemen menekan tombol “Simpan”. 6. Sistem melakukan penambahan data istilah baru pada basis data dan menampilkan pesan status penambahan istilah baru berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> 5a Jika form “Tambah Istilah Baru” tidak valid maka sistem tidak akan melakukan proses tambah istilah dan menampilkan pesan peringatan.		

Lampiran A - 23 Naratif Use Case Edit Istilah

<i>Use Case name</i> : Edit Istilah	<i>ID</i> : UC.23	<i>Importance level</i> : High
<i>Primary actor</i> : Administrator / Manajemen		<i>Use case Type</i> :
<i>Stakeholders and interest</i> :		
Administrator / Manajemen – ingin melakukan perubahan data artikel yang sudah ada.		
<i>Brief description</i> : Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan data istilah yang sudah ada melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger</i> : terdapat data istilah yang tidak sesuai atau tidak benar.		
<i>Type</i> : Eksternal		
<i>Relationships</i> :		
<i>Association</i> : Administrator / Manajemen		
<i>Include</i> : -		
<i>Extend</i> : -		
<i>Generalization</i> : -		
<i>Normal flow of events</i> :		
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar istilah-istilah”. 2. Administrator / Manajemen memilih salah satu istilah yang ada pada panel “Daftar istilah-istilah”. 3. Administrator / Manajemen menekan tombol “Edit”. 4. Sistem menampilkan form “Edit Istilah” pada suatu <i>window pop-up</i> dimana form berisi data istilah yang ada pada basis data. 5. Administrator / Manajemen mengubah data istilah yang lama dengan data istilah baru yang diinginkan. Data istilah yang dapat diubah antara lain : Istilah, Alternatif 1, Alternatif 2, Alternatif 3 dan Definisi 6. Administrator / Manajemen menekan tombol “Update”. 7. Sistem melakukan penyimpanan data istilah yang telah diupdate pada basis data dan menampilkan pesan status proses ubah istilah berhasil dilakukan. 		
<i>Subflows</i> :		
--		
<i>Alternate/exceptional flows</i> :		
3a Jika Administrator / Manajemen tidak memilih salah satu istilah yang ada pada panel “Daftar istilah-istilah” maka sistem akan menampilkan pesan peringatan.		
6a Jika form “Edit Istilah” tidak valid maka sistem tidak akan melakukan proses ubah istilah dan menampilkan pesan peringatan.		

Lampiran A - 24 Naratif Use Case Hapus Istilah

<i>Use Case name :</i> Hapus Istilah	<i>ID :</i> UC.24	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melakukan penghapusan data istilah.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan penghapusan data istilah melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat istilah yang tidak sesuai dan perlu dihapus.		
<i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “Daftar istilah-istilah”. 2. Administrator / Manajemen memilih salah satu artikel yang ada pada panel “Daftar istilah-istilah”. 3. Administrator / Manajemen menekan tombol “Hapus”. 4. Sistem melakukan penghapusan data istilah. 5. Sistem menampilkan pesan status proses hapus istilah berhasil dilakukan. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> 3a Jika Administrator / Manajemen tidak memilih salah satu artikel yang ada pada panel “Daftar Artikel Knowledgebase” maka sistem akan menampilkan pesan peringatan.		



Lampiran A - 25 Naratif Usecase Ubah Data Informasi / Berita

<i>Use Case name :</i>	Ubah Data Informasi / Berita	<i>ID :</i> UC.25	<i>Importance level :</i> High
<i>Primary actor :</i>	Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>			
Administrator / Manajemen – ingin melakukan perubahan informasi / berita.			
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melakukan perubahan data informasi / berita melalui aplikasi web <i>Customer Support</i> .			
<i>Trigger :</i> terdapat informasi / berita terbaru yang ingin dipublikasikan perusahaan.			
<i>Type :</i> Eksternal			
<i>Relationships :</i>			
<i>Association :</i> Administrator / Manajemen			
<i>Include :</i> -			
<i>Extend :</i> -			
<i>Generalization :</i> -			
<i>Normal flow of events :</i>			
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju form “Update informasi / berita”. 2. Administrator / Manajemen mengubah data informasi / berita yang lama dengan data informasi / berita baru yang diinginkan. Data informasi / berita yang dapat diubah antara lain : Subjek Informasi dan Deskripsi 3. Administrator / Manajemen menekan tombol “Update”. 4. Sistem melakukan penyimpanan data informasi / berita yang telah diupdate pada basis data dan menampilkan pesan status proses ubah informasi / berita berhasil dilakukan. 5. Sistem menampilkan data informasi / berita baru pada panel “Informasi / Berita”. 			
<i>Subflows :</i>			
--			
<i>Alternate/exceptional flows :</i>			
4a Jika form “Update informasi / berita” tidak valid maka sistem tidak akan melakukan proses ubah data informasi / berita dan menampilkan pesan peringatan.			



Lampiran A - 26 Naratif Use Case Baca Pesan SMS

<i>Use Case name</i> : Baca Pesan SMS	<i>ID</i> : UC.26	<i>Importance level</i> : High
<i>Primary actor</i> : Administrator / Manajemen		<i>Use case Type</i> :
<i>Stakeholders and interest</i> :		
Administrator / Manajemen – ingin melihat atau membaca pesan permasalahan pelanggan yang dikirimkan melalui SMS.		
<i>Brief description</i> : Use case ini digunakan untuk membuat Administrator / Manajemen dapat membaca pesan permasalahan yang dikirimkan pelanggan melalui SMS pada aplikasi web <i>Customer Support</i> .		
<i>Trigger</i> : Terdapat permasalahan yang disampaikan pelanggan dalam bentuk pesan SMS.		
<i>Type</i> : Eksternal		
<i>Relationships</i> :		
<i>Association</i> : Administrator / Manajemen		
<i>Include</i> : -		
<i>Extend</i> : -		
<i>Generalization</i> : -		
<i>Normal flow of events</i> :		
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel "SMS – Kotak Masuk". 2. Administrator / Manajemen memilih salah satu pesan SMS pada panel "SMS – Kotak Masuk". 3. Administrator / Manajemen menekan tombol "Baca Pesan". 4. Sistem menampilkan isi pesan SMS pada suatu <i>window pop-up</i>. 		
<i>Subflows</i> :		
--		
<i>Alternate/exceptional flows</i> :		
3a Jika Administrator / Manajemen tidak memilih salah satu pesan yang ada pada panel "SMS – Kotak Masuk" maka sistem akan menampilkan pesan peringatan.		

Lampiran A - 27 Naratif Usecase Lihat Kasus WAP

<i>Use Case name :</i>	Lihat Kasus WAP	<i>ID :</i> UC.27	<i>Importance level :</i> High
<i>Primary actor :</i>	Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>			
Administrator / Manajemen – ingin melihat daftar kasus permasalahan yang dikirimkan pelanggan melalui aplikasi WAP <i>Customer Support</i> .			
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melihat daftar kasus permasalahan yang dikirimkan pelanggan melalui aplikasi WAP <i>Customer Support</i> pada aplikasi web <i>Customer Support</i> .			
<i>Trigger :</i> Terdapat kasus permasalahan WAP yang akan diproses atau ditindaklanjuti.			
<i>Type :</i> Eksternal			
<i>Relationships :</i>			
<i>Association :</i> Administrator / Manajemen			
<i>Include :</i> -			
<i>Extend :</i> -			
<i>Generalization :</i> -			
<i>Normal flow of events :</i>			
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “WAP – Kasus”. 2. Sistem menampilkan daftar kasus WAP. 			
<i>Subflows :</i>			
--			
<i>Alternate/exceptional flows :</i>			
<ol style="list-style-type: none"> 1a Administrator / Manajemen dapat melihat kasus berdasarkan status kasus. Untuk kasus aktif Administrator / Manajemen dapat memilih menu kasus aktif dan sebaliknya. 2a Jika tidak terdapat kasus WAP pada basis data maka, sistem akan menampilkan tidak ada kasus pada daftar kasus WAP. 			

Lampiran A - 28 Naratif Use Case Baca Pesan WAP

<i>Use Case name</i> : Baca Pesan WAP	<i>ID</i> : UC.28	<i>Importance level</i> : High
<i>Primary actor</i> : Administrator / Manajemen	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :		
Administrator / Manajemen – ingin melihat pesan WAP yang termasuk pada kasus tertentu.		
<i>Brief description</i> : Use case ini digunakan untuk membuat Administrator / Manajemen dapat melihat isi pesan permasalahan yang dikirimkan pelanggan melalui aplikasi WAP <i>Customer Support</i> pada aplikasi web <i>Customer Support</i> .		
<i>Trigger</i> : Terdapat pesan yang termasuk pada kasus permasalahan WAP tertentu yang akan diproses atau ditindaklanjuti.		
<i>Type</i> : Eksternal		
<i>Relationships</i> :		
<i>Association</i> : Administrator / Manajemen		
<i>Include</i> : -		
<i>Extend</i> : UC.27		
<i>Generalization</i> : -		
<i>Normal flow of events</i> :		
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel “WAP – Kasus”. 2. Administrator / Manajemen menekan tombol “Buka Kasus”. 3. Sistem menampilkan panel “WAP – Pesan” dimana berisi daftar pesan yang termasuk dalam kasus WAP yang dipilih sebelumnya. 4. Administrator / Manajemen memilih salah satu pesan WAP. 5. Administrator / Manajemen menekan tombol “Baca Pesan”. 6. Sistem menampilkan isi pesan WAP pada suatu <i>window pop-up</i>. 		
<i>Subflows</i> :		
--		
<i>Alternate/exceptional flows</i> :		
3a Jika Administrator / Manajemen tidak memilih salah satu pesan yang ada pada panel “WAP – Pesan” maka sistem akan menampilkan pesan peringatan.		

Lampiran A - 29 Naratif Use Case Cari Kasus Web

<i>Use Case name :</i> Cari Kasus Web	<i>ID :</i> UC.29	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Administrator / Manajemen – ingin melakukan pencarian kasus dan pesan permasalahan web tertentu.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator/ Manajemen dapat melakukan pencarian data kasus permasalahan web yang dikirimkan melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat kasus web yang perlu dicari.		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Administrator / Manajemen		
<i>Include :</i> -		
<i>Extend :</i> -		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Administrator / Manajemen menuju form cari kasus web. 2. Administrator / Manajemen mengisi form “Filter” pencarian data kasus web. Filter yang dipakai antara lain : Subjek, Produk, Pengirim, Urutan berdasarkan. 3. Administrator / Manajemen menekan tombol cari. 4. Sistem melakukan pencarian data kasus pada basis data sesuai dengan parameter filter yang diinginkan. 5. Sistem menampilkan hasil pencarian data kasus web pada panel “Hasil Pencarian Kasus”. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
<ol style="list-style-type: none"> 2a Jika form “Filter” dikosongkan proses pencarian tetap bisa dilakukan. 5a Jika data kasus yang dicari tidak ditemukan maka, sistem akan menampilkan pesan tidak ada hasil pencarian pada panel “Hasil Pencarian Kasus”. 		

Lampiran A - 30 Naratif Use Case Lihat Statistik Kasus Web

<i>Use Case name :</i> Lihat Statistik Kasus Web	<i>ID :</i> UC.30	<i>Importance level :</i> High
<i>Primary actor :</i> Administrator / Manajemen	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Administrator / Manajemen – ingin melihat statistik data kasus web berdasarkan tolak ukur tertentu.		
<i>Brief description :</i> Use case ini digunakan untuk membuat Administrator / Manajemen dapat melihat statistik data kasus berdasarkan jumlah parameter tertentu yang direpresentasikan kedalam bentuk chart melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat kebutuhan analisa dari kasus permasalahan berdasarkan parameter tertentu yang diinginkan untuk ditampilkan. <i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Administrator / Manajemen <i>Include :</i> - <i>Extend :</i> - <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Administrator / Manajemen menuju panel "Menu Statistik". 2. Administrator / Manajemen menekan tombol "Lihat chart". 3. Sistem melakukan kalkulasi data kasus berdasarkan jenis menu chart yang dipilih. 4. Sistem menampilkan hasil perhitungan dalam bentuk chart. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> <ol style="list-style-type: none"> 3a Jika chart yang diinginkan berdasarkan parameter jumlah pengguna, maka chart yang ditampilkan adalah jumlah kasus yang dikirimkan oleh tiap pelanggan. 3b Jika chart yang diinginkan berdasarkan parameter jumlah produk maka, chart yang ditampilkan adalah jumlah kasus yang dikelompokkan berdasarkan produk yang dipermasalahkan. 		

Lampiran A - 31 Naratif Use Case Lihat Kasus Web

<i>Use Case name</i> :	Lihat Kasus Web	<i>ID</i> : UC.31	<i>Importance level</i> : High
<i>Primary actor</i> :	Pengguna	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :			
Pengguna – ingin melihat daftar kasus web sesuai dengan otoritasnya masing-masing.			
<i>Brief description</i> : Use case ini digunakan untuk membuat Pengguna dapat melihat daftar data kasus yang disampaikan melalui aplikasi web <i>Customer Support</i> sesuai dengan otoritasnya masing-masing. Otoritas bagi pelanggan adalah kasus web yang dibuat oleh dirinya sendiri.			
<i>Trigger</i> : Terdapat kasus yang dimiliki oleh pengguna sesuai dengan otoritasnya.			
<i>Type</i> : Eksternal			
<i>Relationships</i> :			
<i>Association</i> : Pengguna			
<i>Include</i> : -			
<i>Extend</i> : -			
<i>Generalization</i> : -			
<i>Normal flow of events</i> :			
<ol style="list-style-type: none"> 1. Pengguna menuju panel “WEB – Kasus” 2. Sistem menampilkan daftar kasus web sesuai dengan otoritas yang dimiliki pengguna. 			
<i>Subflows</i> :			
--			
<i>Alternate/exceptional flows</i> :			
<ol style="list-style-type: none"> 1a Untuk administrator dapat memilih menu aktif atau tidak aktif untuk melihat daftar pesan sesuai dengan statusnya. 2a Jika tidak terdapat kasus web maka, sistem akan menampilkan pesan bahwa tidak ada kasus pada panel “WEB – Kasus”. 			

Lampiran A - 32 Naratif Use Case Lihat Pesan Web

<i>Use Case name :</i> Lihat Pesan Web	<i>ID :</i> UC.32	<i>Importance level :</i> High
<i>Primary actor :</i> Pengguna	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i> Pengguna – ingin melihat isi pesan web yang termasuk pada kasus tertentu.		
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat melihat isi data pesan web yang termasuk pada kasus tertentu melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat pesan yang termasuk pada kasus web tertentu yang ingin dilihat atau dibaca pengguna. <i>Type :</i> Eksternal		
<i>Relationships :</i> <i>Association :</i> Pengguna <i>Include :</i> - <i>Extend :</i> UC.31 <i>Generalization :</i> -		
<i>Normal flow of events :</i> <ol style="list-style-type: none"> 1. Pengguna menuju panel “WEB – Kasus”. 2. Pengguna memilih salah satu kasus web. 3. Pengguna menekan tombol “Buka Kasus”. 4. Sistem menampilkan panel “WEB – Pesan” dimana berisi daftar pesan web yang termasuk pada kasus web yang dipilih sebelumnya. 5. Pengguna memilih salah satu pesan web. 6. Pengguna menekan tombol “Baca Pesan”. 7. Sistem menampilkan isi data pesan web yang dipilih pada suatu <i>window pop-up</i>. 		
<i>Subflows :</i> --		
<i>Alternate/exceptional flows :</i> <ol style="list-style-type: none"> 3a Jika Pengguna tidak memilih salah satu kasus yang ada pada panel “WEB – Kasus” maka sistem akan menampilkan pesan peringatan. 6a Jika Pengguna tidak memilih salah satu pesan yang ada pada panel “WEB – Pesan” maka sistem akan menampilkan pesan peringatan. 		

Lampiran A - 33 Naratif Use Case Buat Pesan Web

<i>Use Case name</i> : Buat Pesan Web	<i>ID</i> : UC.33	<i>Importance level</i> : High
<i>Primary actor</i> : Pengguna	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :		
Pengguna – ingin membuat pesan pada kasus web permasalahan tertentu pada aplikasi web.		
<i>Brief description</i> : Use case ini digunakan untuk membuat Pengguna dapat membuat pesan pada kasus web tertentu yang dikimkan melalui aplikasi web <i>Customer Support</i> .		
<i>Trigger</i> : Terdapat pesan yang ingin disampaikan pengguna pada kasus tertentu.		
<i>Type</i> : Eksternal		
<i>Relationships</i> :		
<i>Association</i> : Pengguna		
<i>Include</i> : UC.34		
<i>Extend</i> : UC.32		
<i>Generalization</i> : -		
<i>Normal flow of events</i> :		
<ol style="list-style-type: none"> 1. Pengguna menuju panel "WEB – Kasus". 2. Pengguna memilih salah satu kasus web. 3. Pengguna menekan tombol "Buka Kasus". 4. Sistem menampilkan panel "WEB – Pesan" dimana berisi daftar pesan web yang termasuk pada kasus web yang dipilih sebelumnya. 5. Pengguna memilih salah satu pesan web. 6. Pengguna menekan tombol "Tulis Pesan". 7. Sistem menampilkan form "Tulis pesan baru". Form berisi inputan : Pengirim, ID Produk, Subjek dan Isi pesan. 8. Pengguna mengisi form "tulis pesan baru" dan menekan tombol "Kirim". 9. Sistem menyimpan data pesan yang dibuat pengguna pada basis data. 		
<i>Subflows</i> :		
--		
<i>Alternate/exceptional flows</i> :		
<ol style="list-style-type: none"> 1a Jika pesan dibuat tanpa membuat kasus web maka, pengguna dapat memilih tombol buat pesan baru pada daftar pesan kasus tertentu. 8a Jika form "Tulis pesan baru" tidak valid maka sistem tidak akan melakukan proses buat pesan web dan menampilkan pesan peringatan. 		

Lampiran A - 34 Naratif Use Case Buat Kasus Web

<i>Use Case name</i> :	Buat Kasus Web	<i>ID</i> : UC.34	<i>Importance level</i> : High
<i>Primary actor</i> :	Pelanggan	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :			
Pelanggan – ingin membuat kasus baru menggunakan aplikasi web <i>Customer Support</i> .			
<i>Brief description</i> : Use case ini digunakan untuk membuat Pelanggan dapat membuat kasus tertentu melalui aplikasi web <i>Customer Support</i> .			
<i>Trigger</i> : Pelanggan memiliki permasalahan dengan subjek atau topik permasalahan yang baru dan ingin didiskusikan dengan perusahaan.			
<i>Type</i> : Eksternal			
<i>Relationships</i> :			
<i>Association</i> : Pelanggan			
<i>Include</i> : UC.33			
<i>Extend</i> : -			
<i>Generalization</i> : -			
<i>Normal flow of events</i> :			
<ol style="list-style-type: none"> 1. Pengguna menuju panel “WEB – Kasus”. 2. Pelanggan menekan tombol “Buat Kasus”. 3. Sistem menampilkan form “Buat Kasus Baru” pada suatu <i>window pop-up</i>. 4. Pelanggan mengisi form “Buat Kasus Baru” dengan data kasus baru yang diinginkan. Data kasus antara lain : Subjek, Produk dan Isi Pesan. 5. Pelanggan menekan tombol “Kirim”. 6. Sistem melakukan penyimpanan data kasus baru beserta pesan pertama pada kasus tersebut pada basis data dan menampilkan pesan status proses buat kasus web berhasil dilakukan. 			
<i>Subflows</i> :			
--			
<i>Alternate/exceptional flows</i> :			
5a Jika form “Buat Kasus Baru” tidak valid maka sistem tidak akan melakukan proses buat kasus web dan menampilkan pesan peringatan.			

Lampiran A - 35 Naratif Use Case Buat Kasus WAP (WAP)

<i>Use Case name</i> :	Buat Kasus WAP (WAP)	<i>ID</i> : UC.35	<i>Importance level</i> : High
<i>Primary actor</i> :	Pelanggan	<i>Use case Type</i> :	
<i>Stakeholders and interest</i> :			
Pelanggan – ingin membuat kasus baru menggunakan aplikasi WAP <i>Customer Support</i> .			
<i>Brief description</i> : Use case ini digunakan untuk membuat Pelanggan dapat membuat kasus tertentu melalui aplikasi WAP <i>Customer Support</i> .			
<i>Trigger</i> : Pelanggan memiliki permasalahan dengan subjek atau topik permasalahan yang baru dan ingin didiskusikan dengan perusahaan.			
<i>Type</i> : Eksternal			
<i>Relationships</i> :			
<i>Association</i> : Pelanggan			
<i>Include</i> : -			
<i>Extend</i> : -			
<i>Generalization</i> : -			
<i>Normal flow of events</i> :			
<ol style="list-style-type: none"> 1. Pelanggan menuju "Form Kasus" pada halaman "Kirim kasus baru" 2. Pelanggan mengisi form "Form Kasus" dengan data kasus baru yang diinginkan. Data kasus antara lain : Subjek, Produk dan Isi Pesan. 3. Pelanggan menekan tombol "Send". 4. Sistem melakukan penyimpanan data kasus baru beserta pesan pertama pada kasus tersebut pada basis data. 5. Sistem menampilkan pesan status proses buat kasus WAP berhasil dilakukan. 			
<i>Subflows</i> :			
--			
<i>Alternate/exceptional flows</i> :			
4a Jika form "Buat Kasus Baru" tidak valid maka sistem tidak akan melakukan proses buat kasus web dan menampilkan pesan kesalahan.			

Lampiran A - 36 Naratif Use Case Buat Pesan WAP (WAP)

<i>Use Case name :</i> Buat Pesan WAP (WAP)	<i>ID :</i> UC.36	<i>Importance level :</i> High
<i>Primary actor :</i> Pelanggan	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Pelanggan – ingin membuat pesan pada kasus tertentu melalui aplikasi WAP <i>Customer Support</i> .		
<i>Brief description :</i> Use case ini digunakan untuk membuat Pelanggan dapat membuat pesan pada kasus tertentu melalui aplikasi WAP <i>Customer Support</i> .		
<i>Trigger :</i> Terdapat permasalahan yang dimiliki pelanggan dan ingin disampaikan dalam bentuk pesan melalui aplikasi WAP <i>Customer Support</i> .		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Pelanggan		
<i>Include :</i> UC.35		
<i>Extend :</i> UC.38		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Pelanggan menuju halaman “Baca Pesan”. 2. Pelanggan memilih salah satu kasus WAP pada daftar kasus. 3. Sistem menampilkan daftar pesan pada kasus WAP yang dipilih sebelumnya. 4. Pelanggan memilih salah satu pesan WAP pada daftar pesan. 5. Sistem menampilkan isi pesan yang dipilih pada halaman “Pesan”. 6. Pelanggan memilih link untuk membalas pesan dan membuat pesan WAP baru. 7. Sistem menampilkan form “Balas pesan”. 8. Pelanggan mengisi form “Balas pesan” dan menekan tombol “Send”. 9. Sistem menyimpan data pesan yang dibuat pengguna pada basis data. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
<ol style="list-style-type: none"> 1a Jika pesan dibuat tanpa membuat kasus WAP maka, pelanggan dapat memilih tombol buat pesan baru pada daftar pesan kasus tertentu. 9a Jika form “Balas pesan” tidak valid maka sistem tidak akan melakukan proses buat pesan WAP dan menampilkan pesan kesalahan. 		

Lampiran A - 37 Naratif Use Case Lihat Kasus WAP (WAP)

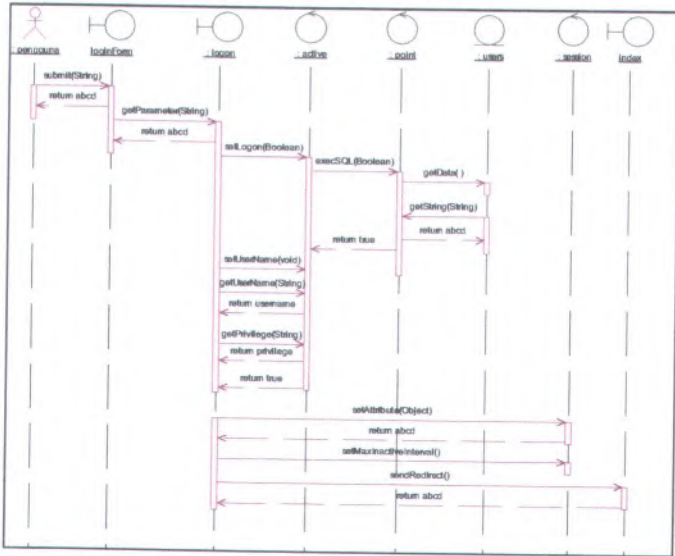
<i>Use Case name :</i>	Lihat Kasus WAP (WAP)	<i>ID :</i> UC.37	<i>Importance level :</i> High
<i>Primary actor :</i> Pelanggan	<i>Use case Type :</i>		
<i>Stakeholders and interest :</i>			
Pelanggan – ingin melihat daftar kasus WAP aktif yang dimilikinya melalui aplikasi WAP <i>Customer Support</i> sesuai dengan otoritasnya masing-masing.			
<i>Brief description :</i> Use case ini digunakan untuk membuat Pelanggan dapat melihat daftar kasus WAP yang dimilikinya melalui aplikasi WAP <i>Customer Support</i> sesuai dengan otoritasnya masing-masing. Otoritas yang dimaksudkan disini adalah kasus yang dibuatnya sendiri.			
<i>Trigger :</i> Terdapat kasus yang belum terselesaikan atau dengan status aktif dan ingin dilihat oleh pelanggan.			
<i>Type :</i> Eksternal			
<i>Relationships :</i>			
<i>Association :</i> Pelanggan			
<i>Include :</i> -			
<i>Extend :</i> -			
<i>Generalization :</i> -			
<i>Normal flow of events :</i>			
<ol style="list-style-type: none"> 1. Pelanggan menuju halaman “Baca pesan” aplikasi WAP <i>Customer Support</i>. 2. Sistem menampilkan data kasus yang aktif yang dimilikinya pada daftar kasus WAP. 			
<i>Subflows :</i>			
--			
<i>Alternate/exceptional flows :</i>			
<ol style="list-style-type: none"> 2a Jika tidak terdapat kasus atau tidak ada kasus yang aktif yang dimiliki oleh pelanggan tersebut maka, sistem akan menampilkan pesan bahwa tidak ada kasus pada daftar kasus yang dimiliki pelanggan tersebut. 			

Lampiran A - 38 Naratif Use Case Lihat Pesan WAP (WAP)

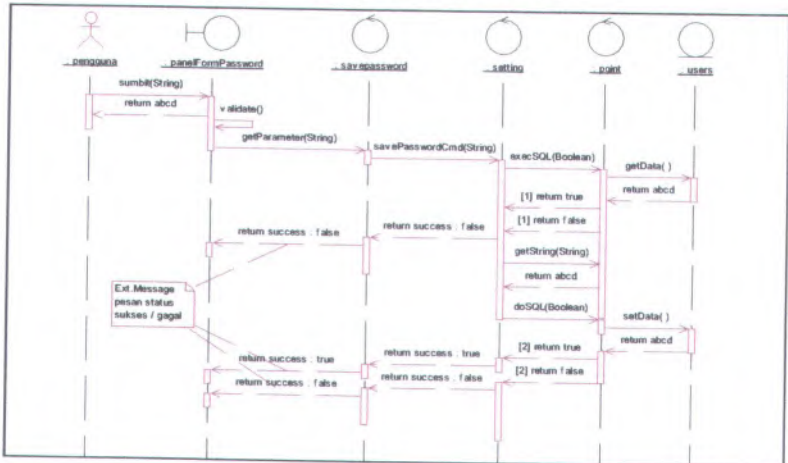
<i>Use Case name :</i> Lihat Pesan WAP (WAP)	<i>ID :</i> UC.38	<i>Importance level :</i> High
<i>Primary actor :</i> Pelanggan	<i>Use case Type :</i>	
<i>Stakeholders and interest :</i>		
Pelanggan – ingin melihat isi pesan pada kasus WAP tertentu melalui aplikasi WAP <i>Customer Support</i> sesuai dengan otoritasnya masing-masing..		
<i>Brief description :</i> Use case ini digunakan untuk membuat pengguna dapat melihat isi pesan melalui aplikasi WAP <i>Customer Support</i>		
<i>Trigger :</i> Terdapat pesan yang ingin dilihat oleh pelanggan melalui aplikasi WAP <i>Customer Support</i> .		
<i>Type :</i> Eksternal		
<i>Relationships :</i>		
<i>Association :</i> Pelanggan		
<i>Include :</i> -		
<i>Extend :</i> -		
<i>Generalization :</i> -		
<i>Normal flow of events :</i>		
<ol style="list-style-type: none"> 1. Pelanggan menuju halaman "Baca pesan" aplikasi WAP <i>Customer Support</i>. 2. Sistem menampilkan daftar kasus WAP aktif yang dimilikinya. 3. Pelanggan memilih salah satu kasus pada daftar kasus yang ada. 4. Sistem menampilkan daftar pesan yang termasuk ke dalam kasus yang dipilih sebelumnya. 5. Pelanggan memilih salah satu pesan. 6. Sistem menampilkan isi pesan yang dipilih pelanggan pada halaman baca pesan WAP. 		
<i>Subflows :</i>		
--		
<i>Alternate/exceptional flows :</i>		
-- --		

LAMPIRAN B

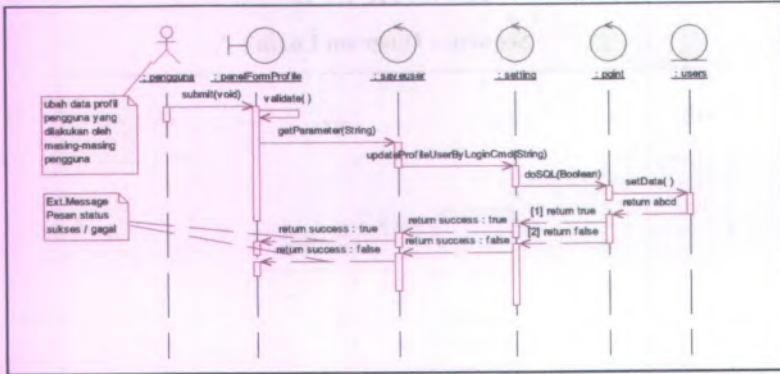
Sequence Diagram Login



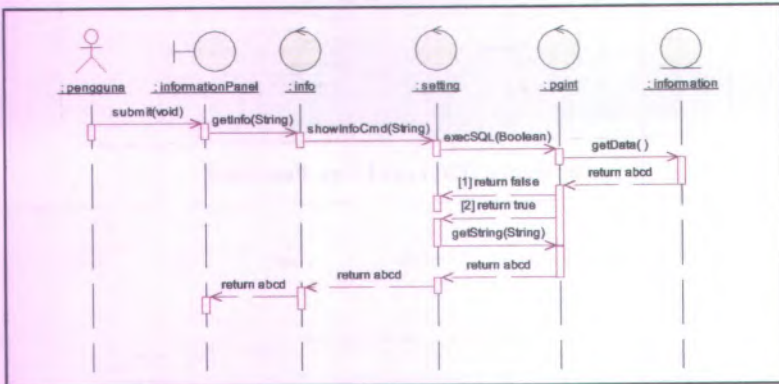
Sequence Diagram Ubah Password



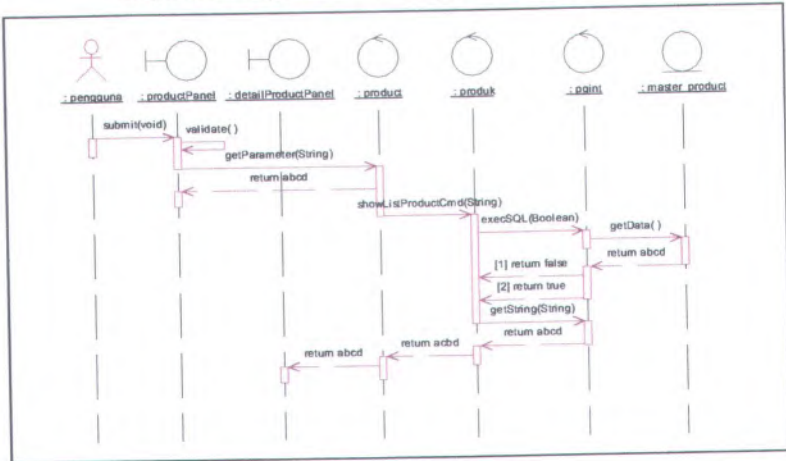
Sequence Diagram Ubah Profil



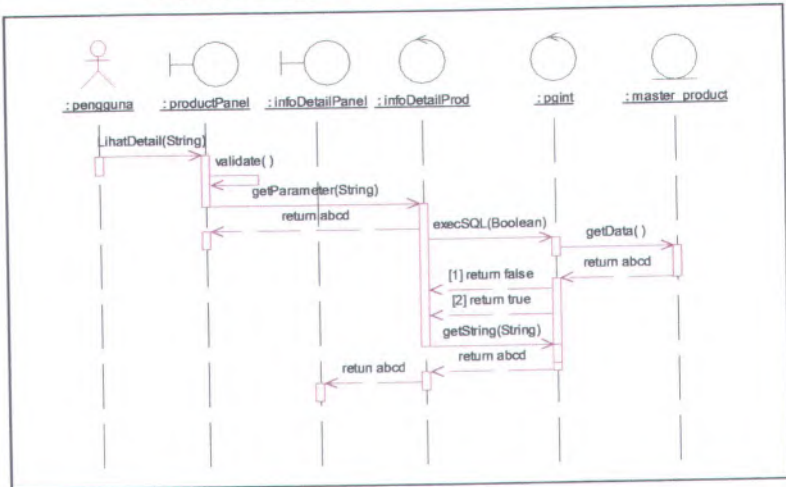
Sequence Diagram Lihat informasi / berita



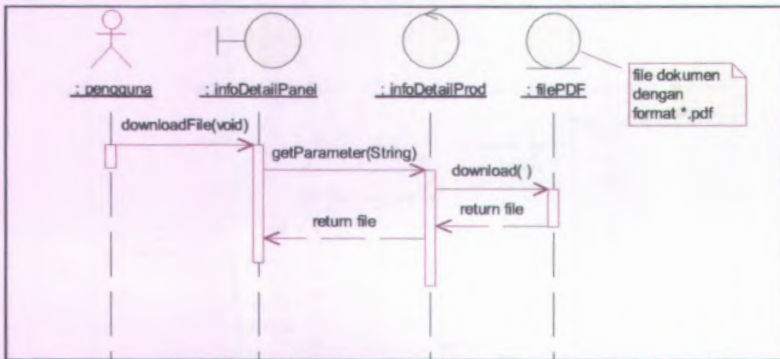
Sequence Diagram Lihat Ringkasan Data Produk



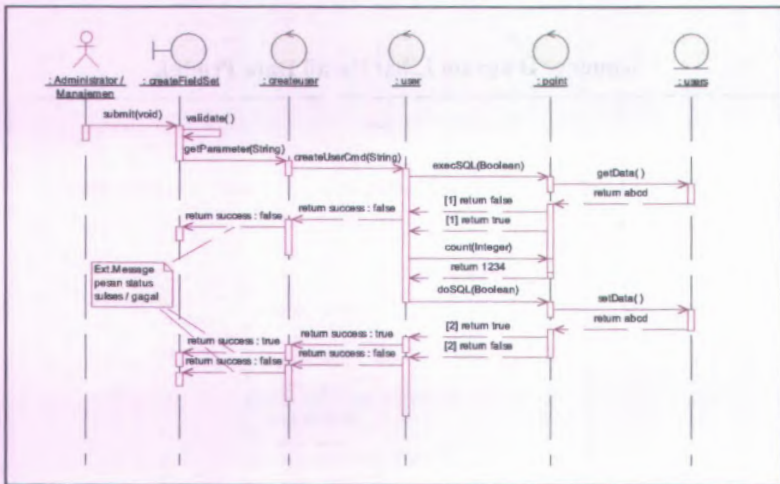
Sequence Diagram Lihat Detail Data Produk



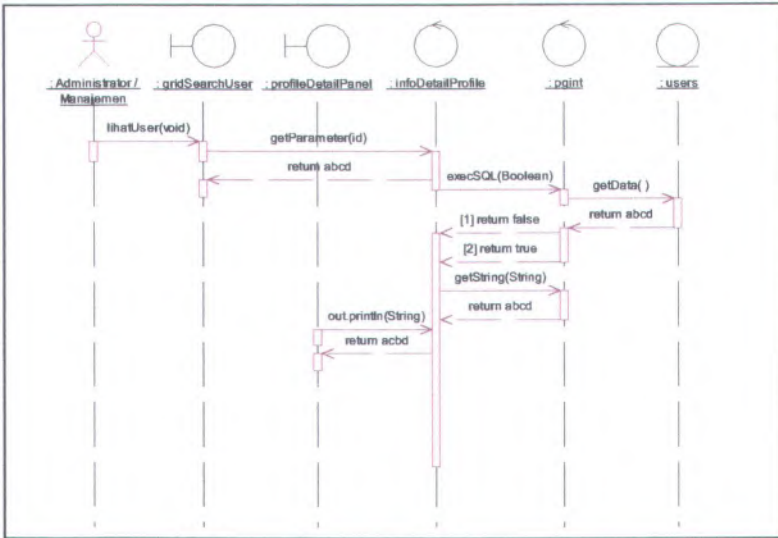
Sequence Diagram Download Dokumen Detail Data Produk



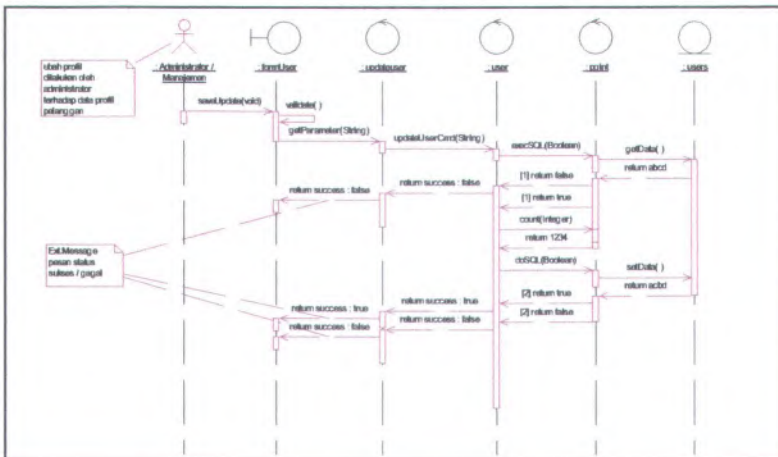
Sequence Diagram Tambah Pengguna



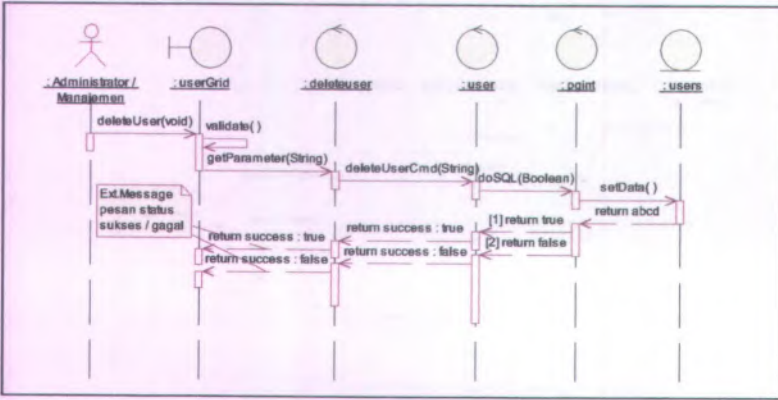
Sequence Diagram Lihat Profil Pengguna



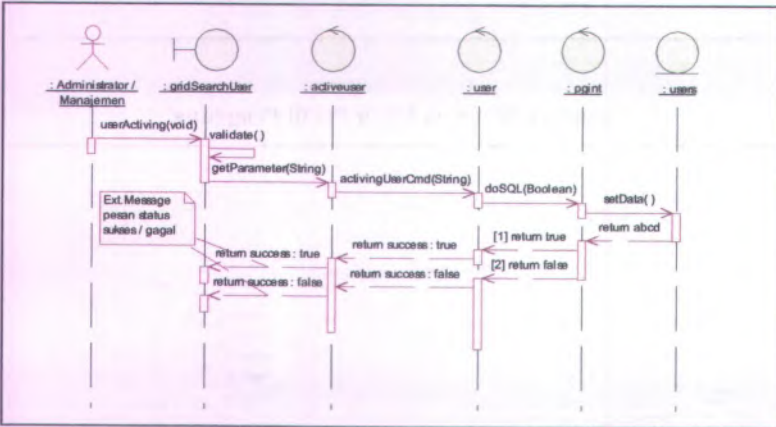
Sequence Diagram Ubah Profil Pengguna



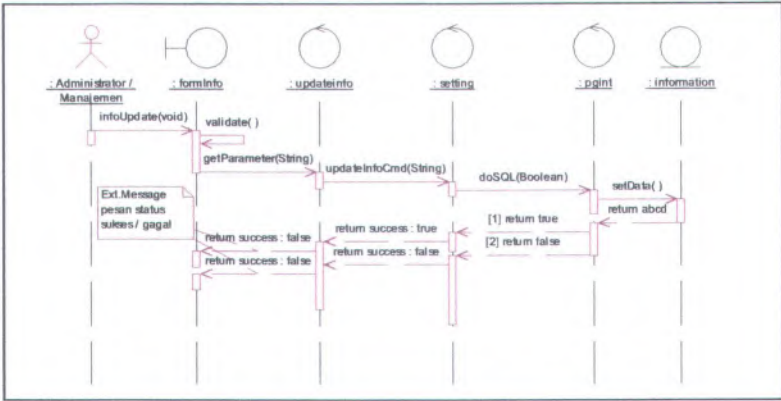
Sequence Diagram Non-aktifkan Pengguna



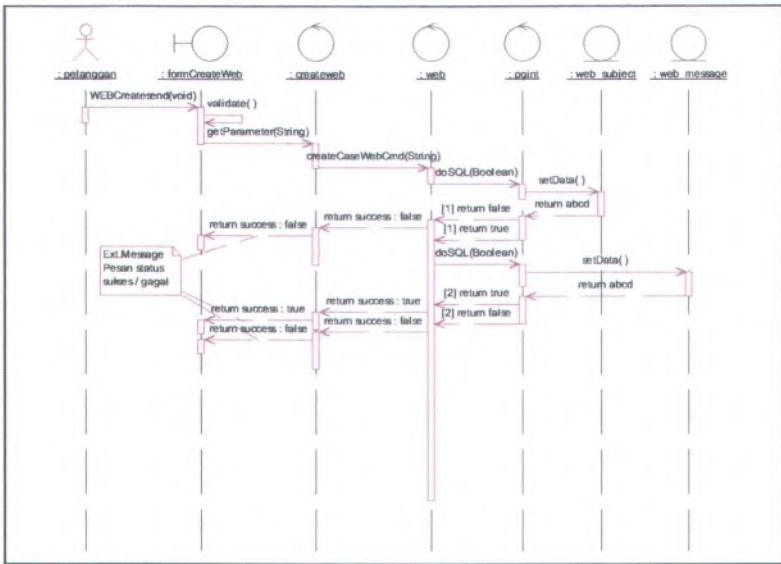
Sequence Diagram Aktifkan Pengguna



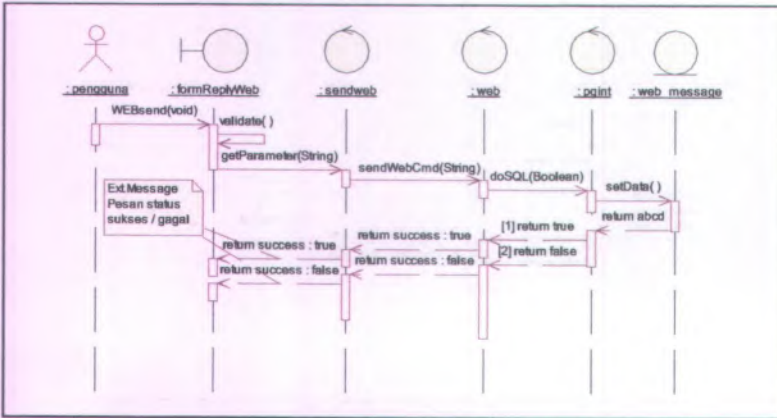
Sequence Diagram Ubah Data Informasi / Berita



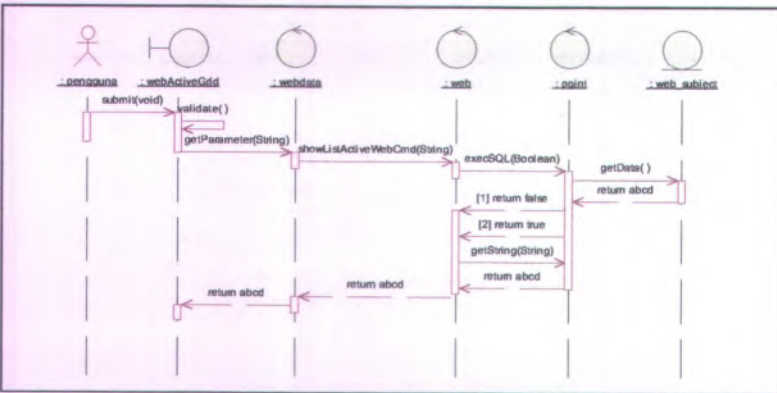
Sequence Diagram Buat Kasus Web



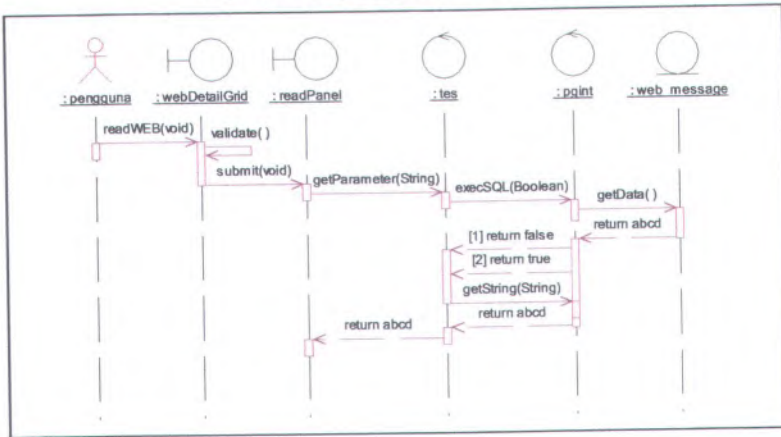
Sequence Diagram Buat Pesan Web



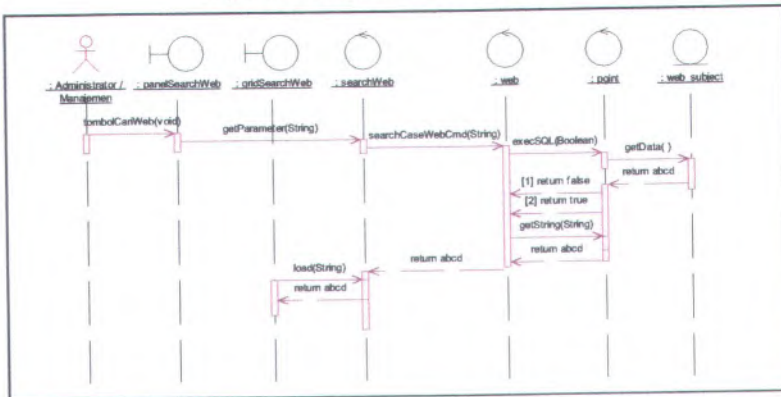
Sequence Diagram Lihat Kasus Web



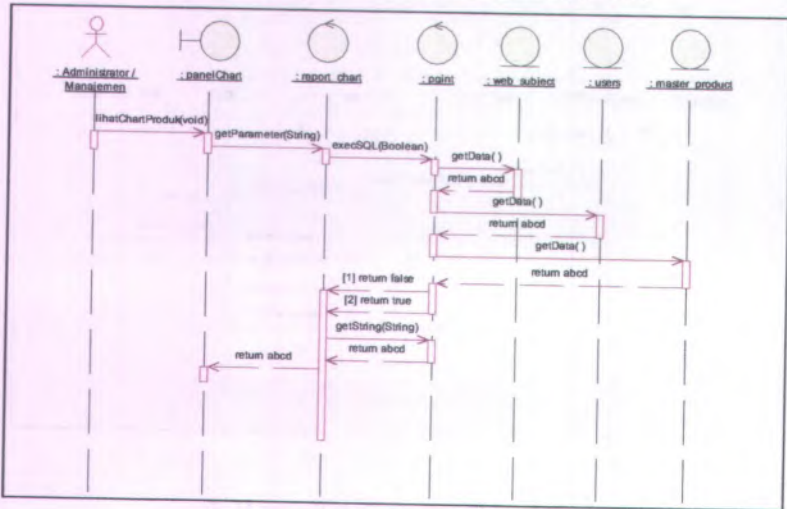
Sequence Diagram Lihat Pesan Web



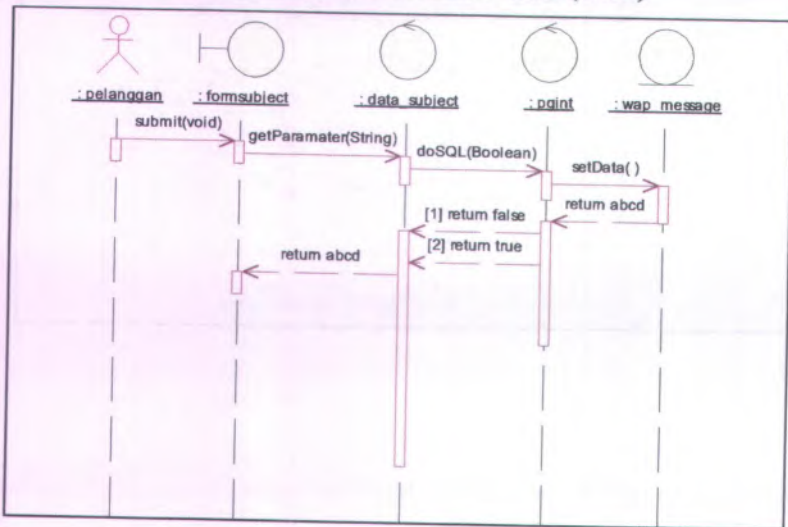
Sequence Diagram Cari Kasus Web



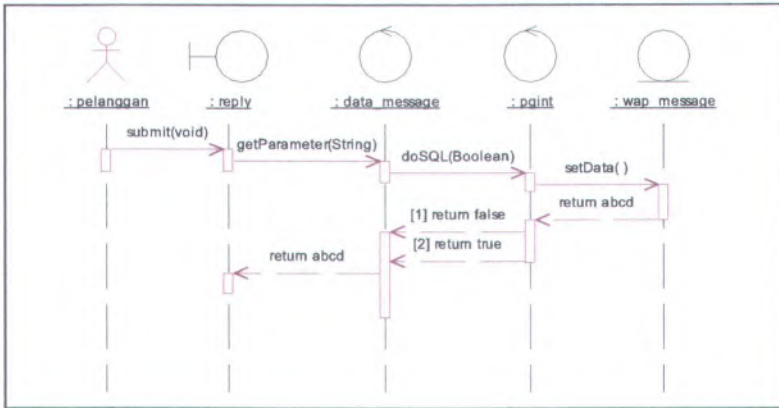
Sequence Diagram Lihat Statistik Kasus Web



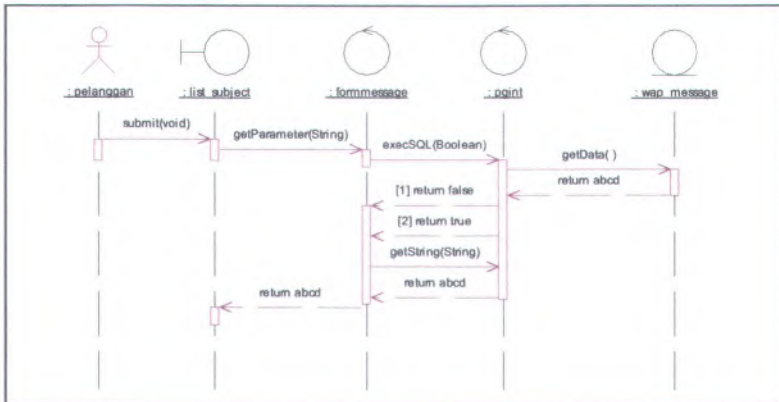
Sequence Diagram Buat Kasus WAP (WAP)



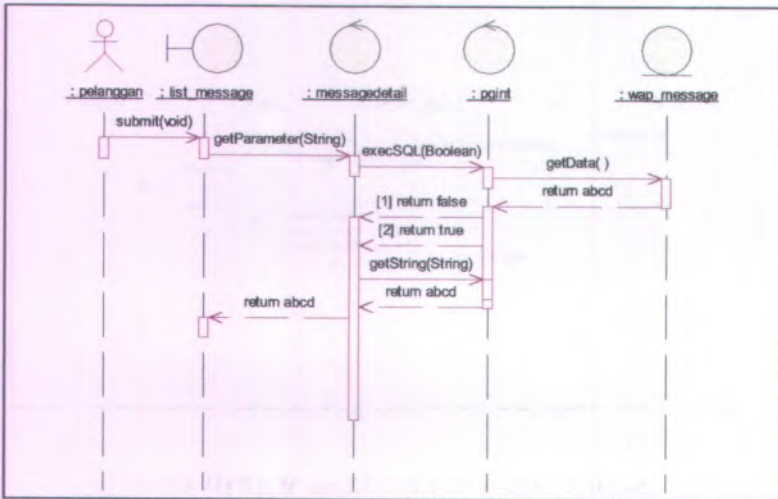
Sequence Diagram Buat Pesan WAP (WAP)



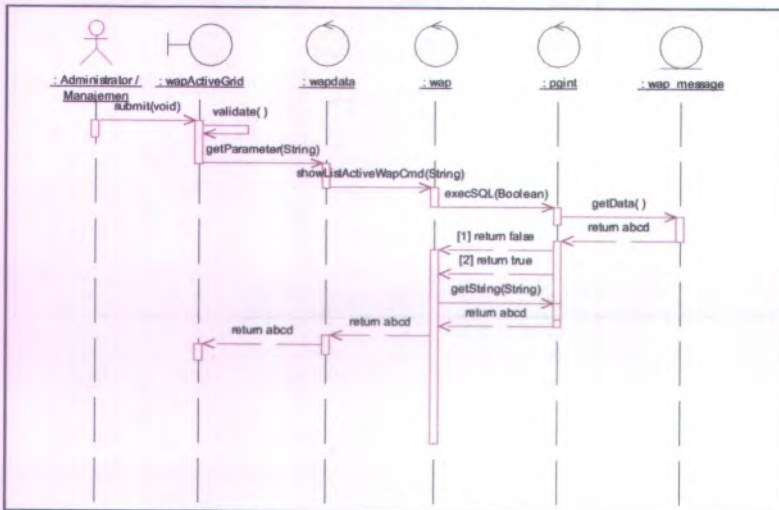
Sequence Diagram Lihat Kasus WAP (WAP)



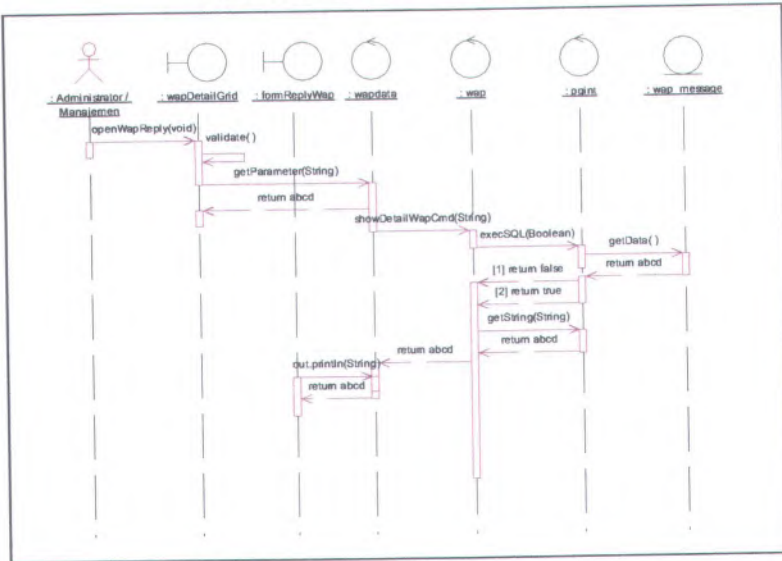
Sequence Diagram Lihat Pesan WAP (WAP)



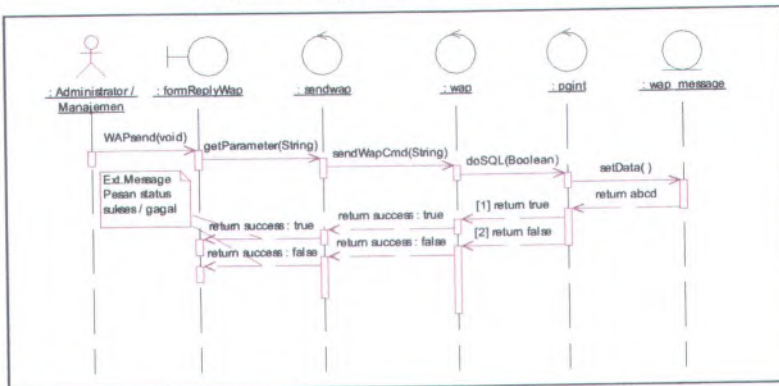
Sequence Diagram Lihat Kasus WAP



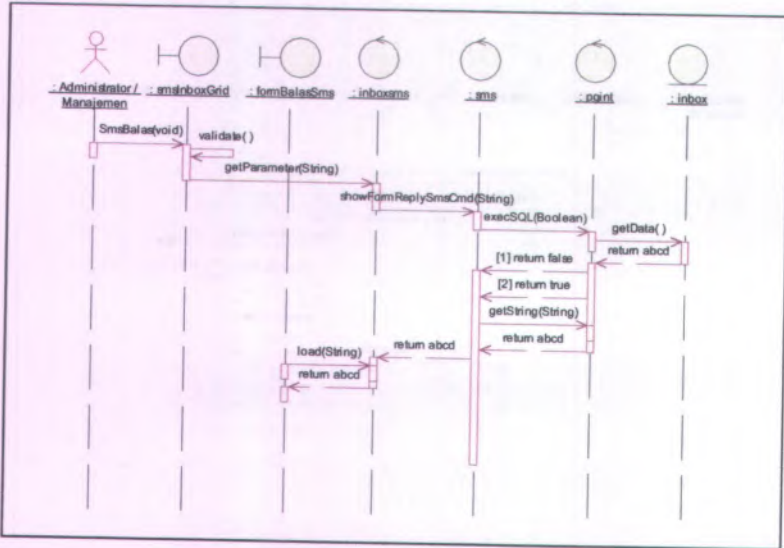
Sequence Diagram Lihat Pesan WAP



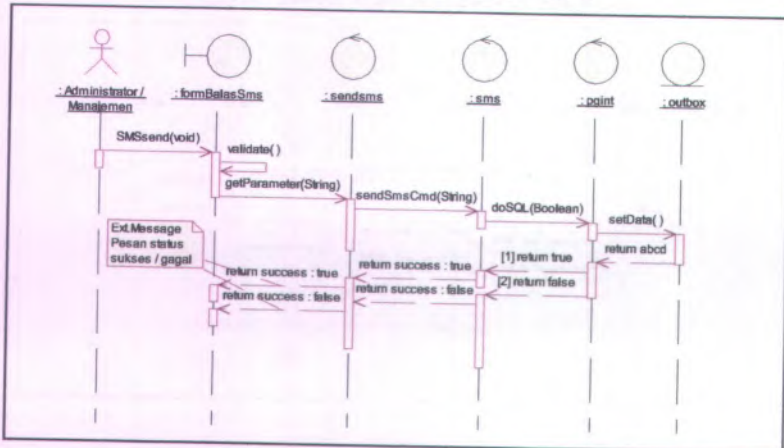
Sequence Diagram Buat Pesan WAP



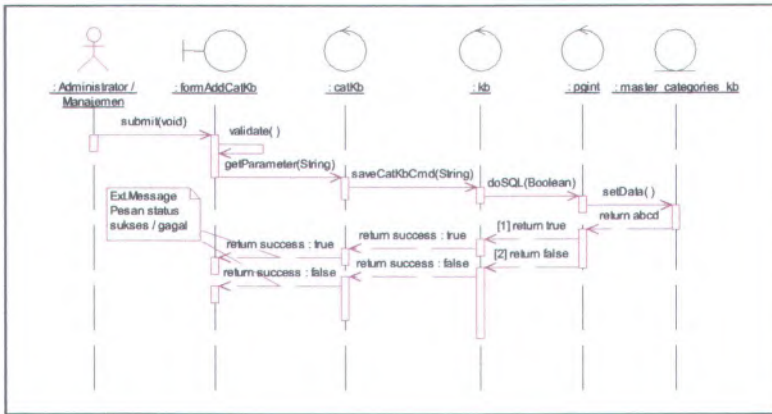
Sequence Diagram Lihat Pesan SMS



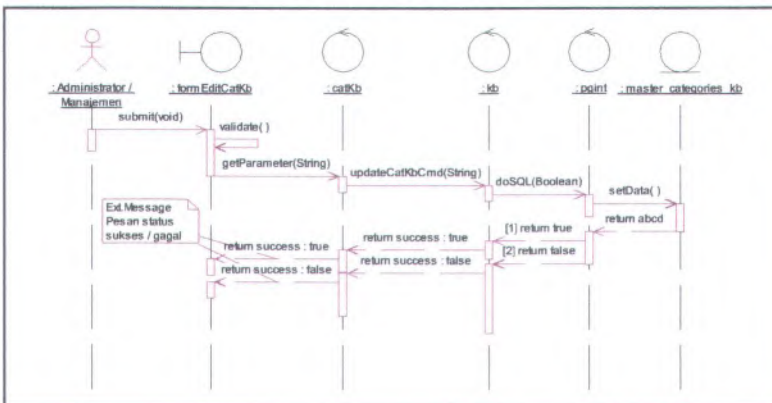
Sequence Diagram Buat Pesan SMS



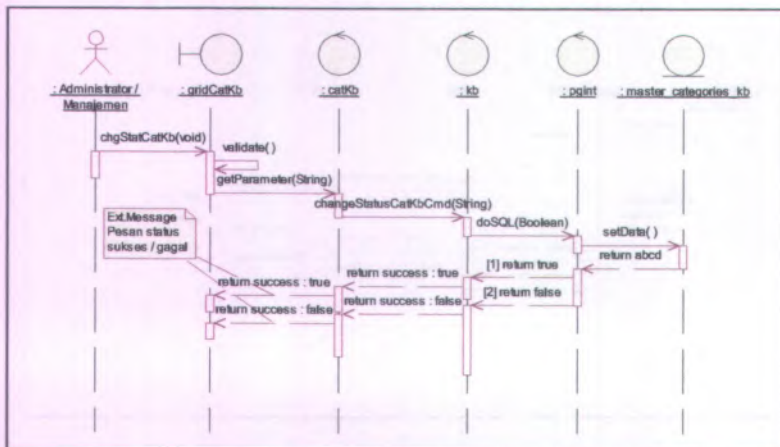
Sequence Diagram Tambah Kategori Artikel Kb



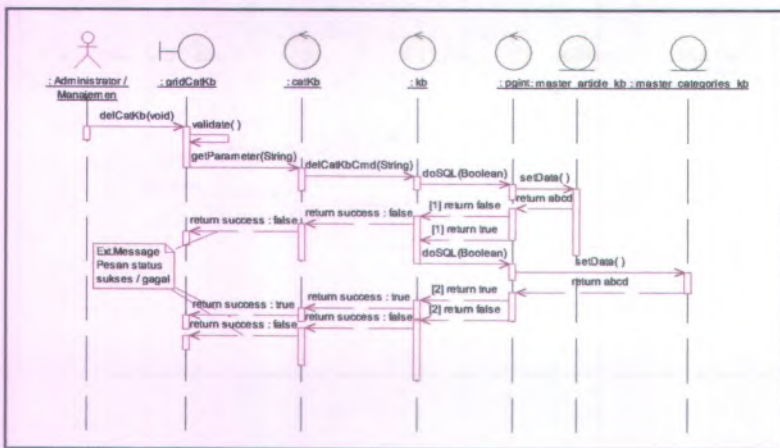
Sequence Diagram Edit Kategori Artikel Kb



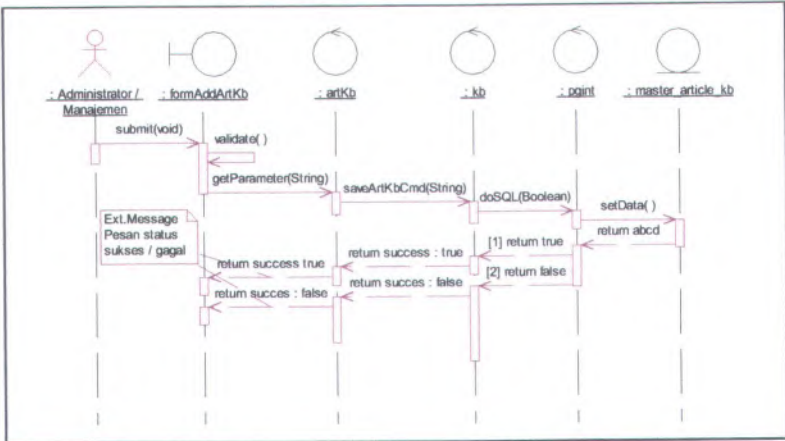
Sequence Diagram Ubah Status Kategori Artikel Kb



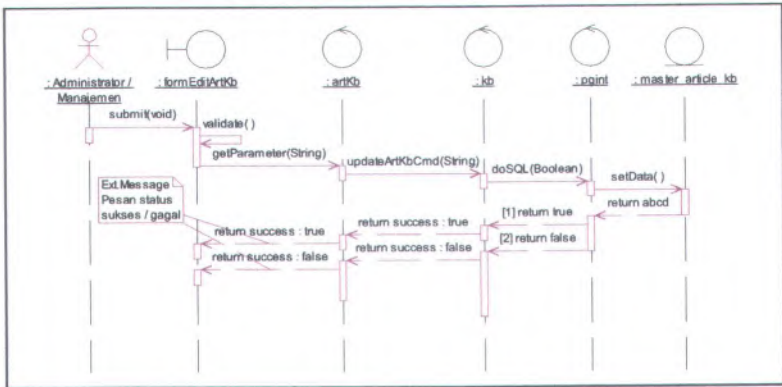
Sequence Diagram Hapus Kategori Artikel Kb



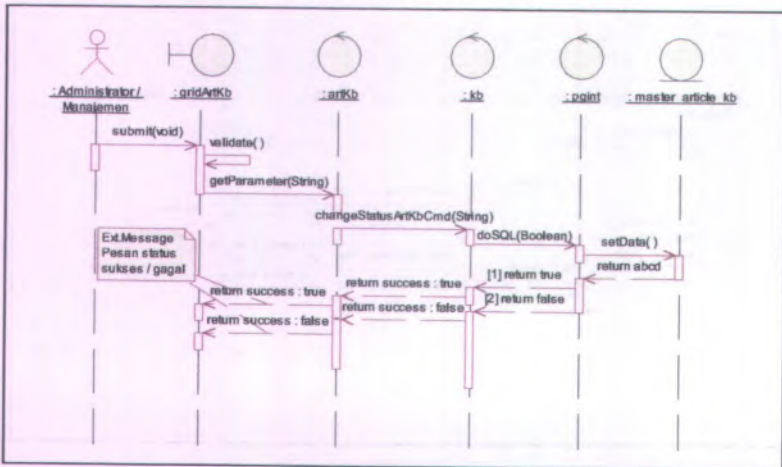
Sequence Diagram Tambah Artikel Kb



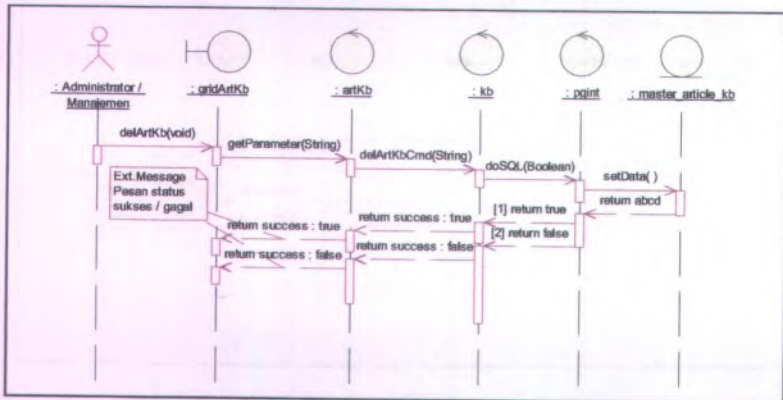
Sequence Diagram Edit Artikel Kb



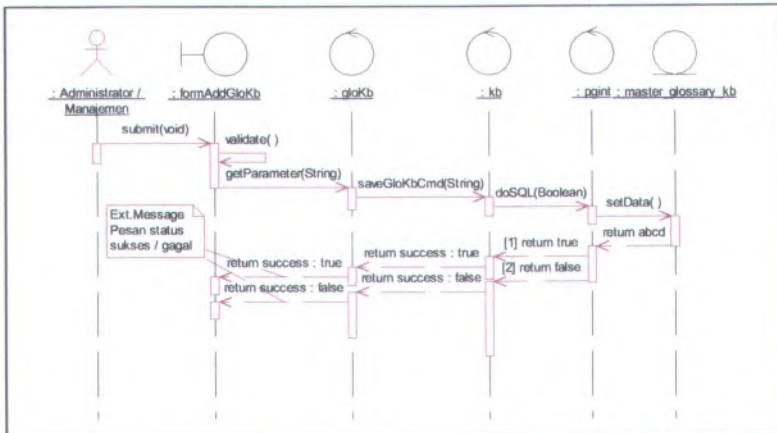
Sequence Diagram Ubah Status Artikel Kb



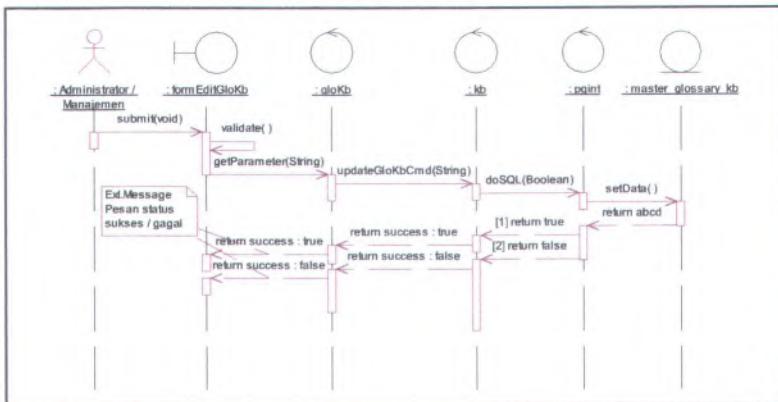
Sequence Diagram Hapus Artikel Kb



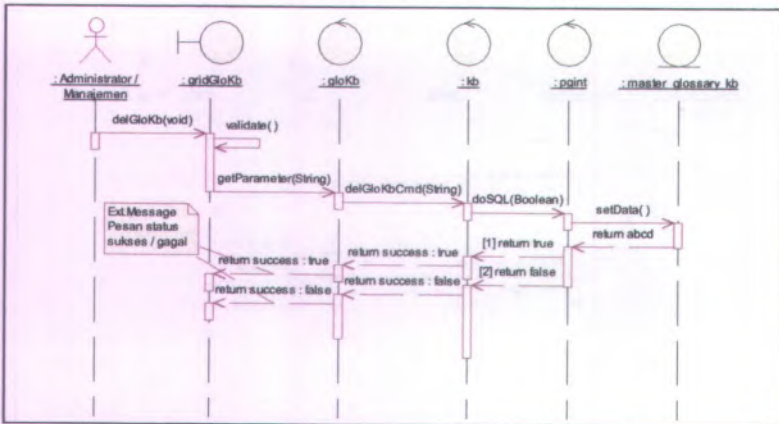
Sequence Diagram Tambah Istilah Kb



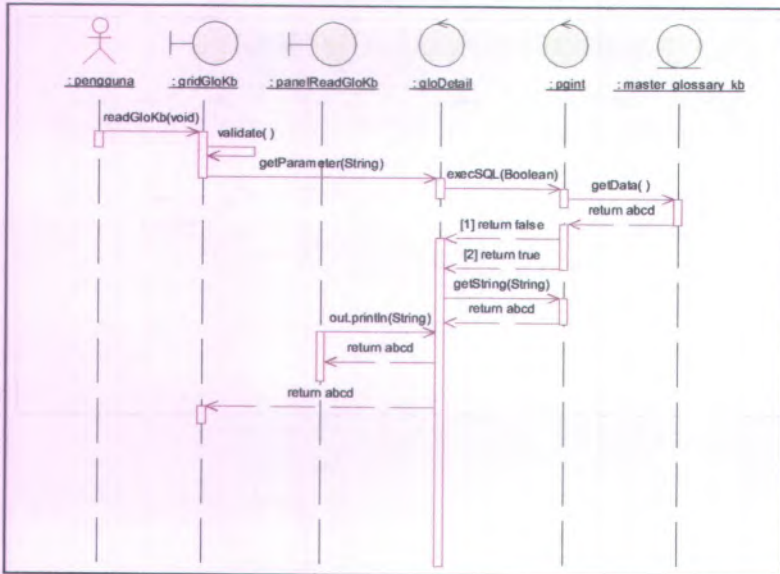
Sequence Diagram Edit Istilah Kb



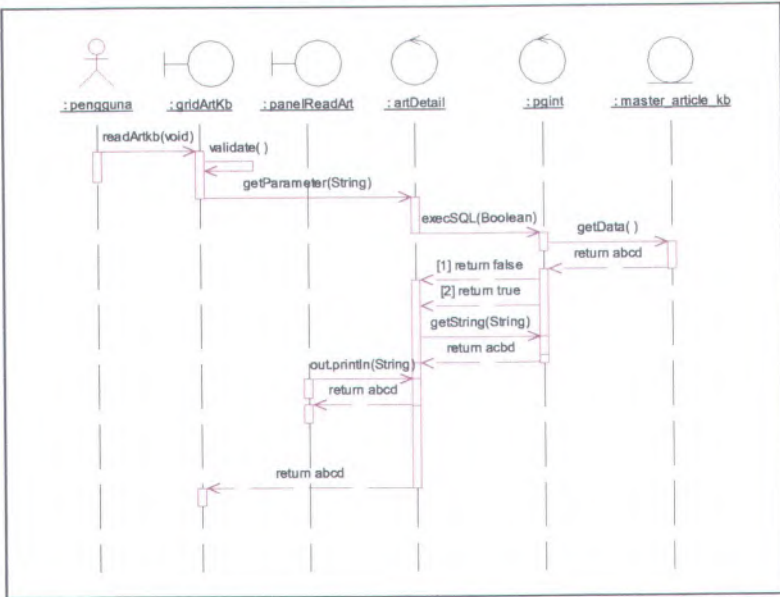
Sequence Diagram Hapus Istilah Kb



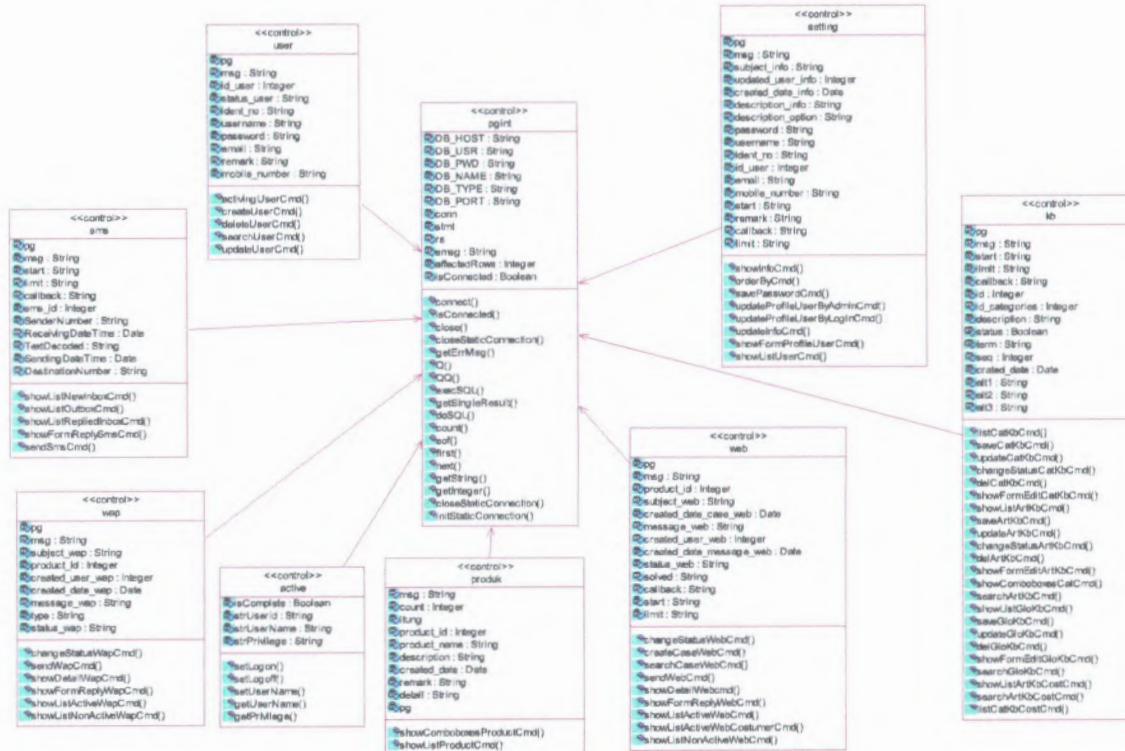
Sequence Diagram Baca Istilah Kb



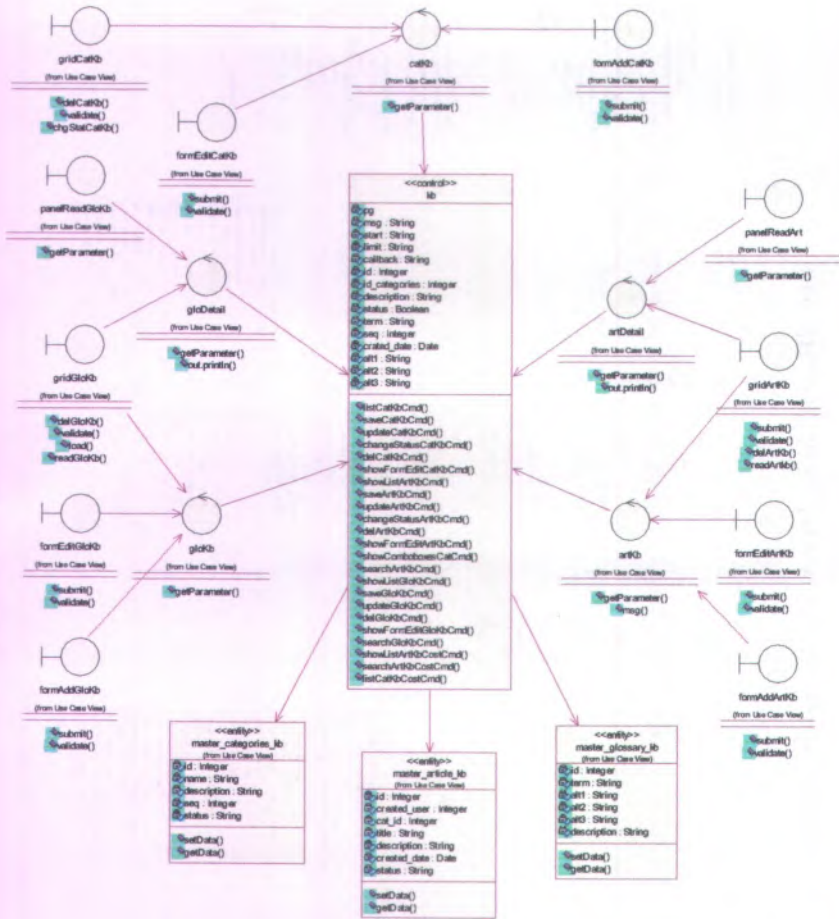
Sequence Diagram Baca Artikel Kb



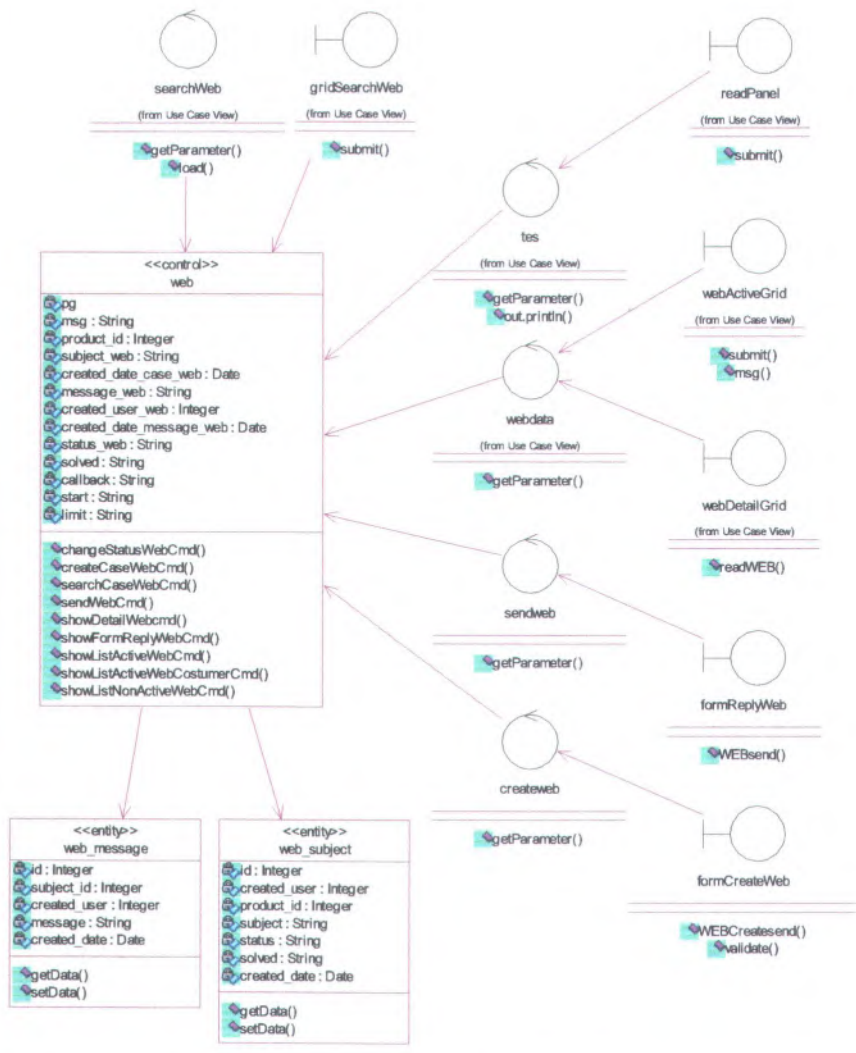
LAMPIRAN C



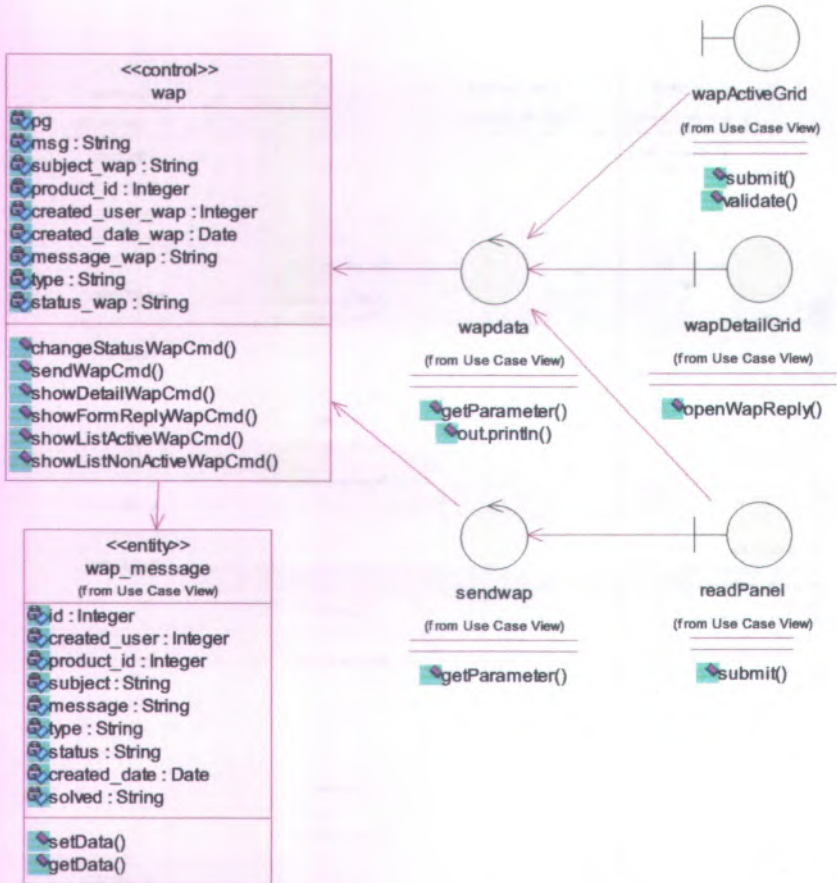
Class Diagram Kelas Java Aplikasi Customer Support



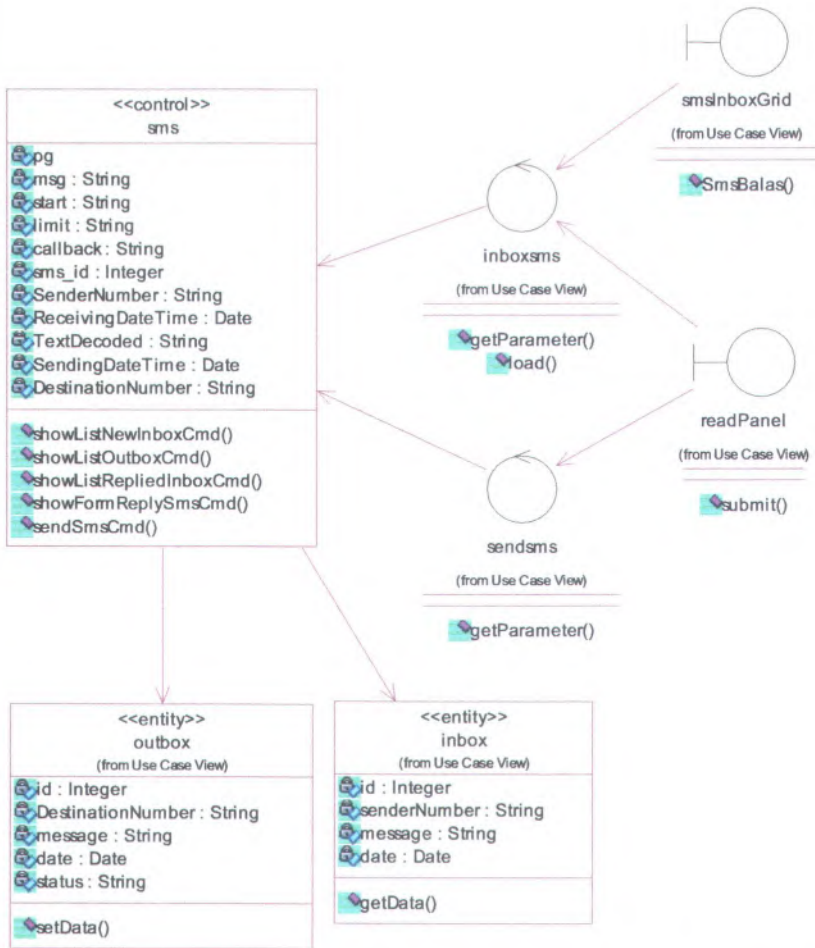
Class Diagram Kelas Java, Controller, Boundary dan Entity Knowledgebase



Class Diagram Kelas Java, Controller, Boundary dan Entity pesan web



Class Diagram Kelas Java, Controller, Boundary dan Entity pesan WAP



Class Diagram Kelas Java, Controller, Boundary dan Entity pesan SMS



Penulis dilahirkan di Jakarta, 8 Desember 1984, Putra dari Bapak Nasril dan Ibu Isdalina. Penulis telah menempuh pendidikan formal yaitu di TK Mekarsari Surabaya, SD Negeri gayungan 423 Surabaya, SLTP Negeri 32 Surabaya, dan SMA Negeri 16 Surabaya.

Pada tahun 2004 Penulis mengikuti SPMB dan diterima di Jurusan Sistem Informasi FTIf-ITS dan terdaftar dengan NRP 5204100053.

Di Jurusan Sistem Informasi ini Penulis mengambil bidang minat *E-Bussiness*. Pada tahun 2007 penulis sudah mulai bekerja di sebuah perusahaan konsultan IT dan software developer dimana penulis mendapatkan banyak ilmu tentang web programming pada perusahaan tersebut.