

**APLIKASI PENDAFTARAN PASIEN KLINIK  
BERBASIS SMS DENGAN INTEGRASI SIM  
(Studi Kasus : MMC)**

**TUGAS AKHIR**

Oleh :  
**IYUT NORTON. T 33105056**

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN APLIKASI PERANGKAT LUNAK  
POLITEKNIK BATAM  
BATAM  
2008**

# LEMBAR PENGESAHAN

Batam, Juni 2008

**Pembimbing I,**

**Riwinoto, ST**  
**NIK. 103025**

**Pembimbing II,**

**Selly Novita, S.Kom**  
**NIK. 107053**

**ABSTRAKSI**  
**APLIKASI PENDAFTARAN PASIEN KLINIK**  
**BERBASIS SMS DENGAN INTEGRASI SIM**  
**(Studi Kasus : MMC)**

Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrai SIM (Studi Kasus : MMC) merupakan suatu terobosan terbaru dalam sistim pendaftaran berobat pasien klinik saat ini. Pendaftaran berobat yang dilakukan di Indonesia khususnya di Kepulauan Riau masih menerapkan sistim manual. Dimana untuk mendaftar, setiap pasien harus datang langsung ke tempat pendaftaran berobat yang disediakan klinik. Pendaftaran secara manual ini membutuhkan waktu yang cukup lama sehingga membuat para pasien jenuh. Serta untuk mendapatkan informasi klinik mengenai nama dokter, jadwal dokter, jumlah pasien dan omzet klinik yang di butuhkan oleh user (pasien, dokter dan direktur) masih secara manual.

Dengan adanya Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrai SIM (Studi Kasus : MMC), para pasien tidak perlu lagi menunggu antrian yang panjang untuk melakukan pendaftaranberobat, serta dapat mengetahui informasi klinik secara langsung setiap saat tanpa harus datang ke klinik.

Keyword : pendaftaran, sms dan Mysql

## KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Kuasa atas berkat, rahmat dan hidayah-Nya sehingga saya dapat Menyelesaikan Tugas Akhir yang berjudul “Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM (Studi Kasus : MMC)”. Dalam Menyelesaikan Tugas Akhir ini saya banyak mendapat bantuan, bimbingan dan dorongan dari berbagai pihak. Oleh karena itu, pada kesempatan ini saya ingin menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya, kepada:

1. Yang terhormat Bapak Dr. Priyono Eko Sanyoto, selaku Direktur Politeknik Batam.
2. Yang terhormat Bapak Riwinoto, ST selaku Dosen Pembimbing 1 Tugas Akhir,
3. Yang terhormat Ibu Selly Novita, S.Kom selaku Dosen Pembimbing 2 Tugas Akhir,
4. Yang terhormat Ibu Evaluata Sembiring, S.Kom selaku Dosen Tugas Akhir 1,
5. Yang terhormat Ibu Metta Santiputri, M,Sc selaku Dosen Tugas Akhir 2,
6. Yang terhormat Bapak dan Ibu dosen selaku dosen penguji Tugas Akhir 1 dan 2,
7. Yang tercinta Keluarga saya yang telah banyak memberikan dukungannya,
8. Teman-teman APL angkatan 2005 atas dukungan dan kerjasamanya.
9. Serta semua pihak yang turut membantu dalam penyelesaian Tugas Akhir.

Serta penulis memohon maaf jika terdapat kesalahan-kesalahan dalam pembuatan aplikasi serta kesilapan dalam penulisan laporan ini. Akhir kata, saya sebagai penulis mengucapkan terima kasih. Dengan harapan semoga aplikasi yang telah saya buat ini dapat bermanfaat serta dapat diterapkan pada klinik yang ada di Kepulauan Riau khususnya kota Batam.

Batam. Januari 2008

Penulis

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Kesehatan adalah salah satu hal yang sangat penting dalam kehidupan sehingga kehadiran klinik sebagai salah satu sarana bagi kesehatan juga tidak kalah pentingnya. Tentunya akan sangat baik jika informasi klinik juga diiringi dengan perkembangan teknologi yang digunakannya. Khususnya untuk proses yang sampai saat ini kebanyakan masih dilakukan secara manual sehingga kurang memberikan keefektifan dan keefisienan bagi pasien dan pihak klinik dalam mengelola management klinik. Kekurangan dalam pengelolaan management yang masih manual tersebut adalah :

1. Proses pendaftaran berobat di klinik masih dilakukan secara manual dan membutuhkan waktu yang cukup lama, sehingga membuat para pasien jenuh.
2. Semua informasi data pasien yang terdaftar sebagai member klinik masih disimpan dalam bentuk kertas yang disimpan di dalam map yang terpisah-pisah berarti dalam hal ini masih dilakukan pendataan secara tertulis.
3. Informasi klinik mengenai nama dokter, jadwal dokter, jumlah pasien dan omzet klinik yang dibutuhkan oleh user (pasien, dokter dan direktur) masih secara manual.

Oleh karena itu seiring dengan perkembangan ilmu teknologi saat ini yang semakin pesat, diperlukan suatu aplikasi yang dapat mempercepat proses dan transaksi yang mungkin terjadi pada suatu klinik, namun kemungkinan human error-nya sangat sedikit. Dengan menggunakan Sistem Manajemen Data Pasien yang telah ada (dibuat oleh M. Zulkurnaen), teknologi SMS(Short Message Service), HP(Handphone) dan IT(Information Technology) diharapkan dapat menghasilkan suatu aplikasi yang dapat mengatasi permasalahan yang timbul seperti diatas.

Untuk mengatasi hal tersebut, maka perlu dikembangkan Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM pada klinik yang dapat membantu pasien dalam hal pendaftaran berobat di klinik. Aplikasi ini juga diharapkan dapat membantu pasien, dokter dan direktur untuk mendapatkan informasi nama dokter yang bertugas, jadwal dokter dan jumlah pasien yang melakukan pendaftaran berobat melalui SMS di klinik. Dan dengan adanya aplikasi ini diharapkan dapat membantu direktur untuk mendapatkan informasi jumlah omzet klinik perhari.

## 1.2 Tujuan

Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM (Studi Kasus: MMC) ini memiliki tujuan-tujuan sebagai berikut :

1. Mempermudah user biasa (pasien) dalam proses pendaftaran berobat ke klinik dengan proses yang lebih cepat, mudah dan efisien tanpa harus meninggalkan aktivitas mereka melalui SMS.
2. Mempermudah user (pasien, dokter dan direktur) mengetahui nama dokter yang bertugas, jadwal dokter dan jumlah pasien yang mendaftar berobat melalui SMS.
3. Mempermudah direktur dalam mengecek omzet perhari klinik melalui SMS.

### 1.3 Batasan Masalah

Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM (Studi Kasus: MMC) ini mempunyai batasan masalah sebagai berikut :

1. Aplikasi ini hanya berjalan di dalam jaringan lokal (LAN).
2. Aplikasi ini hanya menangani pendaftaran berobat untuk pasien member (pasien yang merupakan karyawan dari perusahaan yang terdaftar sebagai member klinik) melalui SMS.
3. Aplikasi ini hanya menangani pendaftaran berobat untuk satu klinik melalui SMS.
4. Aplikasi ini hanya menyediakan informasi Nama Dokter, Jadwal Dokter, Jumlah Pasien yang mendaftar berobat dan Jumlah Omzet perhari klinik melalui SMS.
5. Aplikasi ini tidak menangani pengelolaan obat-obatan dan gaji karyawan.
6. Aplikasi ini tidak menangani pembuatan report gaji karyawan (Ket : Direktur, Dokter, Perawat, Administrasi dan Customer Service) klinik, laporan keuangan dan pendapatan klinik.

### 1.4 Ikhtisar Buku

Sistematika penulisan laporan ini adalah sebagai berikut :

- BAB 1 : Pendahuluan  
berisi penjelasan mengenai latar belakang, tujuan, batasan masalah, dan ikhtisar buku.
- BAB 2 : Deskripsi Umum Aplikasi  
berisi tentang deskripsi umum sistem, karakteristik pengguna dan lingkungan operasional yang terdiri atas perangkat keras dan perangkat lunak.
- BAB 3 : Analisis  
berisi tentang deskripsi perangkat keras serta hubungan antara perangkat lunak dan perangkat keras. Pembuatan Context diagram dan DFD level yang menggambarkan alur data dalam suatu aplikasi.
- BAB 4 : Deskripsi Perancangan  
berisi tentang penamaan dan penomoran, deskripsi data, deskripsi fungsional yang di dalamnya terdiri dari struktur menu dan spesifikasi fungsi, dalam bab ini ada juga rancangan implementasi fisik yang terbagi atas spesifikasi ketergantungan antara modul dan juga struktur direktori dan deskripsi file.
- BAB 5 : Implementasi dan Pengujian  
berisi tentang penjelasan singkat implementasi yang telah dibuat dan penulisan source code juga pengujian terhadap setiap fungsi menu yang telah diterapkan dan penerapannya sehingga menjadi suatu program aplikasi yang telah siap pakai.
- BAB 6 : Kesimpulan dan Saran  
berisi tentang kesimpulan atas pembuatan Aplikasi Pilkada Berbasis Sms dan saran untuk pengembangan selanjutnya.
- Lampiran A : Perancangan Rinci Fungsional  
Berisi tentang spesifikasi fungsi/proses dari dari sistem yang terdiri dari Spesifikasi layar utama, spesifikasi objek pada layar, spesifikasi layar pesan, spesifikasi algoritma, spesifikasi report..
- Lampiran B : Uraian Rinci Library  
berisi tentang rincian librari yang dipakai dalam aplikasi.
- Lampiran C : Daftar Rinci File dan Data  
berisi tentang struktur Direktori pengembangan Apliaksi.
- Lampiran D : Dokumen Rinci dan testing  
berisi tentang Tabel pengujian fungsi-fungsi pada aplikasi.



## 2.2 Karakteristik Pengguna

**Tabel 2-1 Kategori Pengguna Aplikasi**

Kategori Pengguna	Tugas	Hak Akses ke aplikasi	Jabatan
Admin	<ul style="list-style-type: none"> <li>Menambah, mengedit dan menghapus data pasien member, data dokter dan direktur.</li> <li>Menjaga keamanan sistem.</li> </ul>	<ul style="list-style-type: none"> <li>Menambah data</li> <li>Mengedit data pasien, dokter dan direktur</li> <li>Menghapus data</li> </ul>	Admin
Dokter	<ul style="list-style-type: none"> <li>Mencatat data mengenai diagnosa pasien.</li> <li>Mencatat tindakan-tindakan medis yang dilakukan terhadap pasien.</li> <li>Mencatat daftar resep obat.</li> </ul>	<ul style="list-style-type: none"> <li>Mengecek jumlah pasien yang berobat melalui SMS</li> <li>Mengecek jadwal dokter melalui SMS</li> <li>Mengecek nama dokter melalui SMS</li> </ul>	Staff
Direktur	<ul style="list-style-type: none"> <li>Mengurus management klinik.</li> <li>Mengecek omzet klinik perhari.</li> </ul>	<ul style="list-style-type: none"> <li>Mengatur, memantau dan mengecek omzet klinik perhari</li> <li>Mengetahui nama dokter melalui SMS</li> <li>Mengetahui jumlah pasien melalui SMS</li> <li>Mengetahui jadwal dokter</li> </ul>	Direktur
Pasien	-	<ul style="list-style-type: none"> <li>Melakukan Pendaftaran</li> <li>Mengetahui nama dokter melalui SMS</li> <li>Mengetahui jadwal dokter melalui SMS</li> <li>Mengetahui jumlah pasien melalui SMS</li> </ul>	User biasa

## 2.3 Batasan Sistem

Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM (Studi Kasus: MMC) ini memiliki batasan sebagai berikut :

- Aplikasi ini dikembangkan dari aplikasi sebelumnya yaitu Sistem Management Data Pasien yang di buat oleh M. Zulkurnaen (mahasiswa Politeknik Batam jurusan Teknik Informatika angkatan 2004).
- Aplikasi ini mempunyai ketergantungan dengan aplikasi Sistem Management Data Pasien
- Aplikasi ini dikembangkan dengan pemrograman C#.
- Aplikasi ini menggunakan Database Mysql.
- Aplikasi ini menggunakan library gammu for windows.

## 2.4 Lingkungan Operasional

Aplikasi ini mempunyai dua lingkungan operasional yaitu perangkat keras dan perangkat lunak.

### 2.4.1 Perangkat Keras

- a. Prosesor : Pentium IV 2.4 GHz
- b. Memori : 256 MB
- c. Handphone
  - merek : Sony Ericson
  - type : K610 i
- d. USB kabel data Sony Ericson

### 2.4.2 Perangkat Lunak

- a. Sistem Operasi : Microsoft Windows XP
- b. Bahasa Pemrograman : Microsoft Visual Studio 2005
- c. DBMS : Mysql server version 5.0
- d. Program/utilities lain : MyODBC-3.51.06, xampp-win32-1.6.0a
- e. Library : Gammu for windows

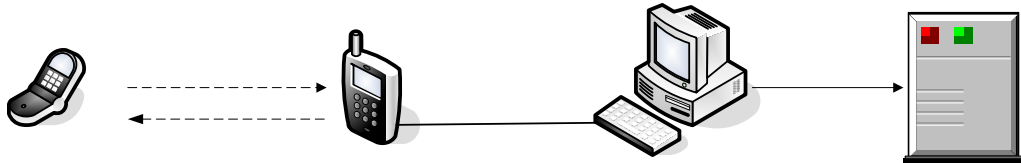
## 2.5 Aturan Penamaan dan Penomoran

Spesifikasi aturan penomoran laporan Tugas Akhir ini adalah sebagai berikut:

1. Aturan penamaan bab menggunakan angka, dilanjutkan dengan penulisan judul bab.  
Contoh: BAB 1 Pendahuluan
2. Aturan penamaan sub bab menggunakan angka, dilanjutkan dengan penulisan judul sub bab.  
Contoh: 1.1. Latar Belakang.
3. Aturan penamaan sub sub bab, menggunakan angka. Terdiri dari nomor bab dan nomor sub bab dan sub sub bab.  
Contoh: 2.4.1 Lingkungan Operasional
4. Aturan penamaan untuk gambar menggunakan angka, dilanjutkan dengan penulisan nama gambar.  
Contoh: Gambar 2.1 deskripsi umum aplikasi
5. Aturan Penamaan lampiran menggunakan huruf besar di awalnya.  
Contoh: Lampiran A
6. Aturan penamaan sub lampiran menggunakan huruf besar dan angka.  
Contoh: A.1 Spesifikasi
7. Aturan penamaan tabel program menggunakan angka, dilanjutkan dengan penulisan nama tabel  
Contoh: *Tabel 2.1 Karakteristik Pengguna*

## BAB 3 ANALISIS

### 3.1 Deskripsi Perangkat Keras



Gambar 3.1 deskripsi perangkat keras

Perangkat keras yang digunakan dalam Aplikasi Pendaftaran Pasien Klinik Berbasis SMS (Studi Kasus: MMC) Dengan Integrasi SIM yaitu :

1. Handphone yang berfungsi untuk mengirimkan SMS .
2. Handphone sebagai Sms Gateway yang berfungsi sebagai penerima sms.
3. Kabel konektor yang berfungsi sebagai penghubung antara handphone (sms gateway) dengan sebuah PC.
4. Komputer yang berfungsi sebagai pusat pengolahan data.
5. Server yang berfungsi sebagai tempat penyimpanan data.

SMS Gateway

### 3.2 Hubungan Antara Perangkat Keras dan Perangkat Lunak

Aplikasi ini menggunakan 2 ponsel yaitu ponsel pengirim dan ponsel penerima yang bertipe *Sony Ericsson K610 i* juga kabel konektor yang mempunyai seri RPM 131 12/1 R1A 06W50 untuk menghubungkan ponsel penerima dengan komputer.

### 3.3 Deskripsi Fungsional Handphone User

Deskripsi fungsional dari aplikasi ini adalah sebagai berikut:

#### 3.3.1 Context Diagram



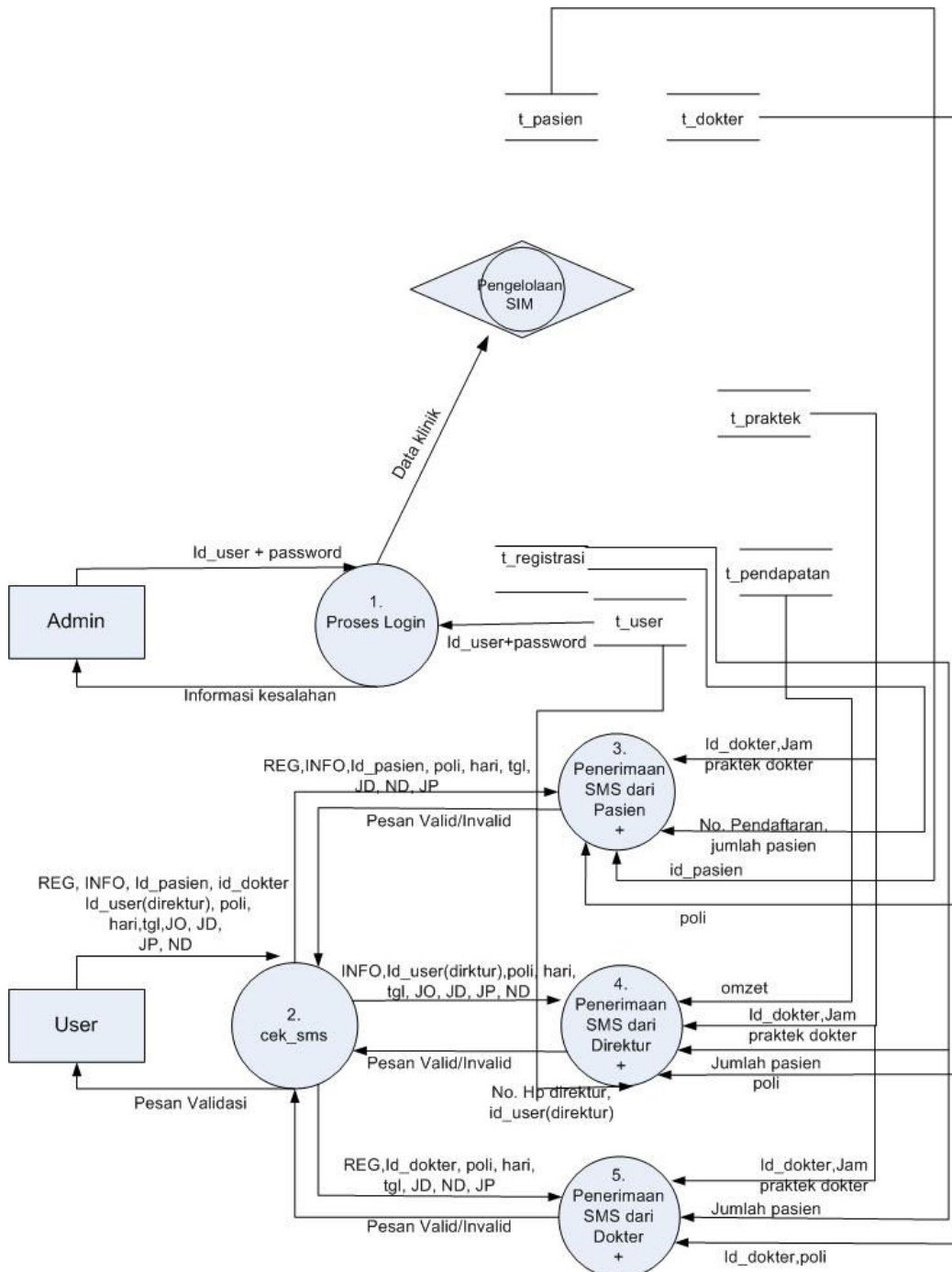
Gambar 3.2 Context Diagram

Penjelasan gambar :

User yang menjalankan aplikasi ini adalah admin, sedangkan user yang menggunakan fasilitas yang di sediakan aplikasi adalah dokter, direktur dan pasien klinik melalui SMS. Admin memasukkan id user dan password ke aplikasi, apabila berhasil akan muncul menu utama aplikasi, sedangkan jika terjadi kesalahan dalam memasukkan id user atau password akan ada informasi pesan kesalahan. Dokter, direktur dan pasien dapat mengetahui beberapa informasi

tentang klinik yang dapat diketahui melalui SMS yang dikirim ke aplikasi sesuai dengan format yang telah ditentukan.

### 3.3.1.1 DFD Level 1

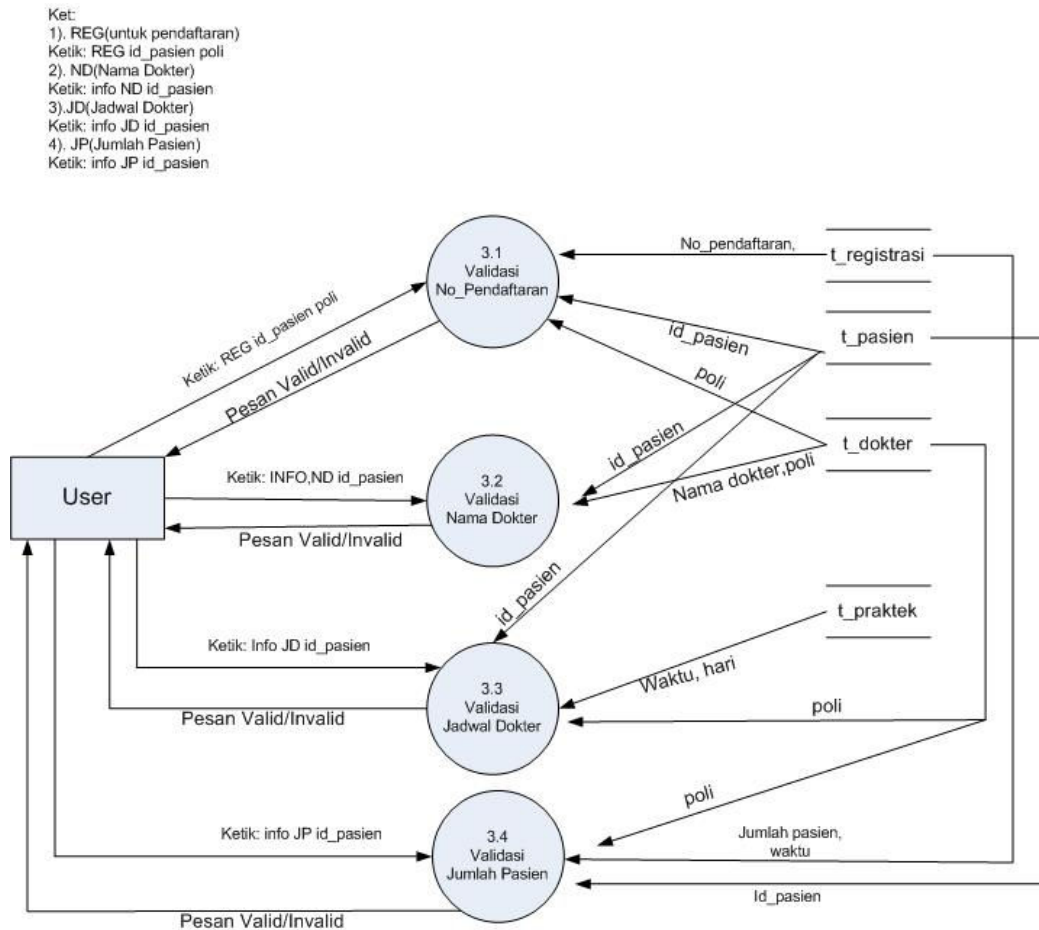


Gambar 3.3 DFD Level 1

Pada DFD level 1 terdapat 6 proses. Proses yang pertama yaitu Login. Pada proses ini yang mempunyai hak akses adalah Admin. Admin harus login terlebih dahulu sebelum masuk ke dalam sistem. Pada proses ini terjadi aliran data dari tabel t\_user ke proses 1 dan terjadi pengecekan data dari data yang di input oleh Admin dan data dari tabel t\_user. Proses yang kedua yaitu proses Pengelolaan SIM, pada

proses ini yang terjadi yaitu pemanggilan aplikasi Pengelolaan SIM. Proses ketiga yaitu proses Penerimaan SMS dari Pasien. Proses ini menerima inputan data dari pasien melalui SMS, berupa id\_pasien, poli, hari, tgl, JD, ND, JP setelah di proses didalam proses cek\_sms. Kemudian inputan data tersebut akan diproses dan menghasilkan output berupa pesan valid atau invalid. Pada proses ini terjadi aliran data dari tabel t\_pasien, t\_praktek, t\_dokter dan t\_registrasi berupa id pasien, poli, nama dokter, jadwal dokter, no\_pendaftaran, dan jumlah pasien. Proses keempat yaitu Penerimaan SMS dari Direktur. Proses ini menerima inputan data dari direktur melalui SMS, berupa id\_user(direktur), poli, hari, tgl, JD, ND, JP dan JO setelah di proses didalam proses cek\_sms. Kemudian inputan data tersebut akan diproses dan menghasilkan output berupa pesan valid atau invalid. Pada proses ini terjadi aliran data dari tabel t\_user, t\_pendapatan, t\_praktek dan t\_registrasi berupa id\_user(direktur), no\_hp\_user, poli, nama dokter, jadwal dokter, omzet dan jumlah pasien. Proses kelima yaitu proses Penerimaan SMS dari Dokter. Proses ini menerima inputan data dari dokter melalui SMS, berupa id\_dokter, poli, hari, tgl, JD, ND, JP setelah di proses didalam proses cek\_sms. Kemudian inputan data tersebut akan diproses dan menghasilkan output berupa pesan valid atau invalid. Pada proses ini terjadi aliran data dari tabel t\_dokter, t\_praktek dan t\_registrasi berupa id\_dokter, jam praktek dokter, poli dan jumlah pasien. Proses keenam yaitu proses cek\_sms. Pada proses ini terjadi pengecekan format sms yang diinput oleh user.

### 3.3.1.2 DFD Level 2 Proses Penerimaan SMS dari Pasien

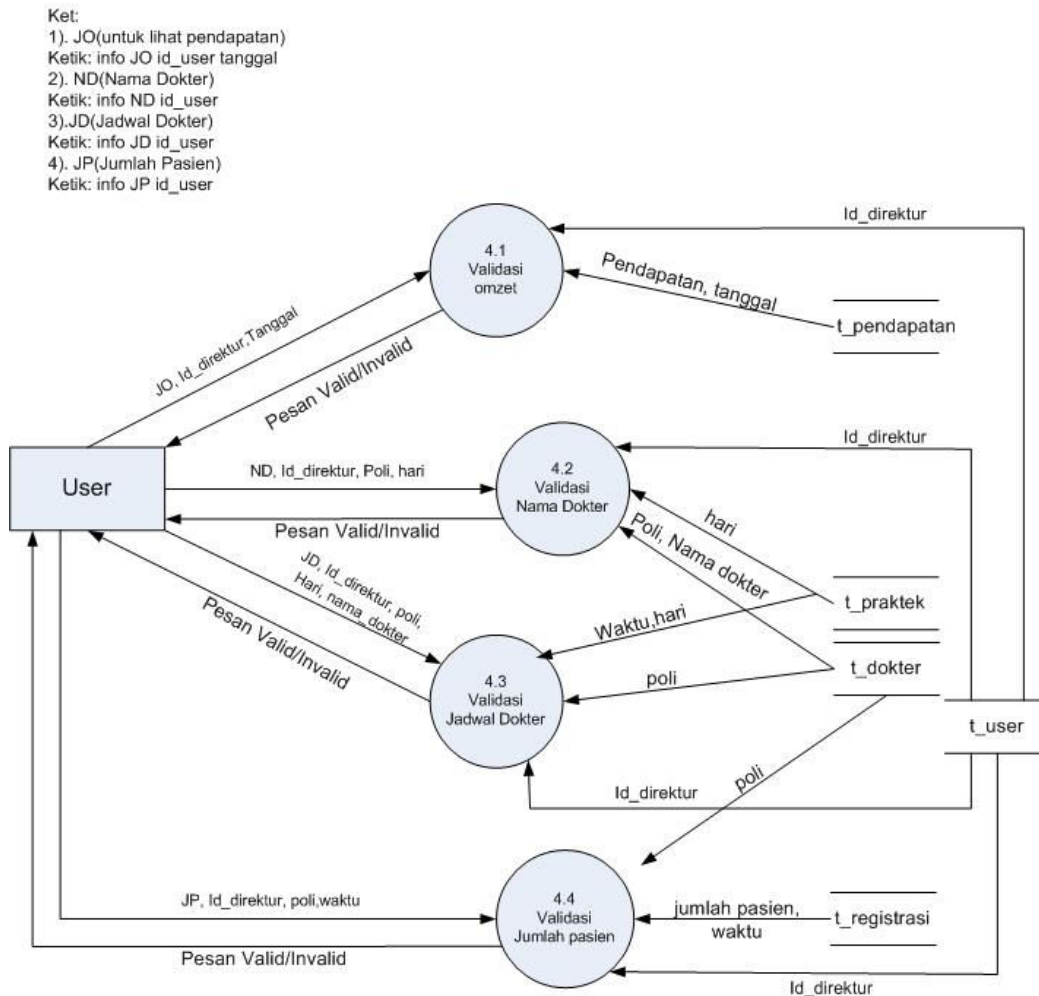


Gambar 3.4 DFD Level 2 proses Penerimaan SMS dari Pasien

Proses yang terjadi pada DFD level 2 Proses Penerimaan SMS dari Pasien ada 4 proses.

Proses Validasi no\_pendaftaran menerima input data berupa id\_pasien dan poli dari User(Pasien) dan data id\_pasien, poli dari tabel t\_pasien dan t\_dokter yang selanjutnya diproses dan menghasilkan pesan validasi berupa no\_pendaftaran yang akan dikirim oleh aplikasi ke User dan data pasien disimpan ke t\_registrasi. Pada proses Validasi Nama Dokter menerima input data berupa JD(Jadwal Dokter), id\_pasien, poli dan hari dari User(Pasien) dan data Nama Dokter, poli dari t\_dokter yang selanjutnya diproses dan menghasilkan pesan validasi berupa Nama Dokter yang bertugas yang akan dikirim oleh aplikasi ke User. Pada proses Validasi Jadwal Dokter menerima input data berupa JD(Jadwal Dokter), id\_pasien, poli, hari dan nama dokter dari User(Pasien). data Jam Praktek Dokter dari t\_praktek yang selanjutnya akan menghasilkan pesan validasi berupa jadwal berobat yang bertugas yang akan dikirim oleh aplikasi ke User. Pada proses Validasi Jumlah Pasien menerima input data berupa id\_pasien dan JP dari User(Pasien) dan data Jumlah Pasien dari t\_registrasi yang selanjutnya akan menghasilkan pesan validasi berupa data jumlah pasien yang mendaftar berobat yang akan dikirim oleh aplikasi ke User.

### 3.3.1.3 DFD Level 2 Proses Penerimaan SMS dari Direktur

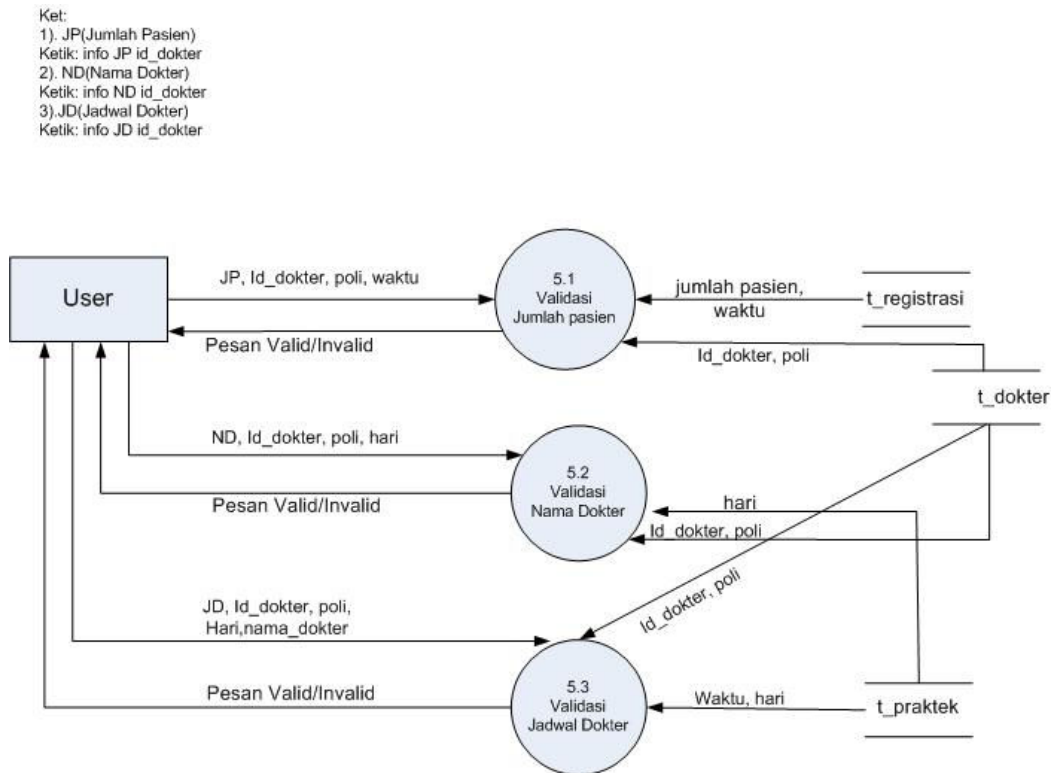


Gambar 3.5 DFD Level 2 proses Penerimaan SMS dari Direktur

Proses yang terjadi pada DFD level 2 Proses Penerimaan SMS dari Direktur ada 4 proses. Pada proses Validasi Omzet menerima input data berupa id\_direktur dan omzet dari User(Direktur) dan data pendapatan(omzet) dari t\_pendapatan yang selanjutnya akan menghasilkan pesan validasi berupa jumlah pendapatan perhari yang akan dikirim oleh aplikasi ke User. Pada proses Validasi Nama Dokter menerima input data berupa id\_direktur dan ND dari User(Direktur) dan data Nama Dokter dari t\_praktek yang selanjutnya akan menghasilkan pesan validasi berupa Nama Dokter yang bertugas yang

akan dikirim oleh aplikasi ke User. Pada proses Validasi Jadwal Dokter menerima input data berupa id\_direktur dan JD dari User(Direktur) dan data Jam Praktek Dokter dari t\_praktek yang selanjutnya akan menghasilkan pesan validasi berupa Jadwal Berobat yang bertugas yang akan dikirim oleh aplikasi ke User. Pada proses Validasi Jumlah Pasien menerima input data berupa id\_direktur dan JP dari User(Direktur) data Jumlah Pasien dari t\_registrasi yang selanjutnya akan menghasilkan pesan validasi berupa data jumlah pasien yang mendaftar berobat yang akan dikirim oleh aplikasi ke User.

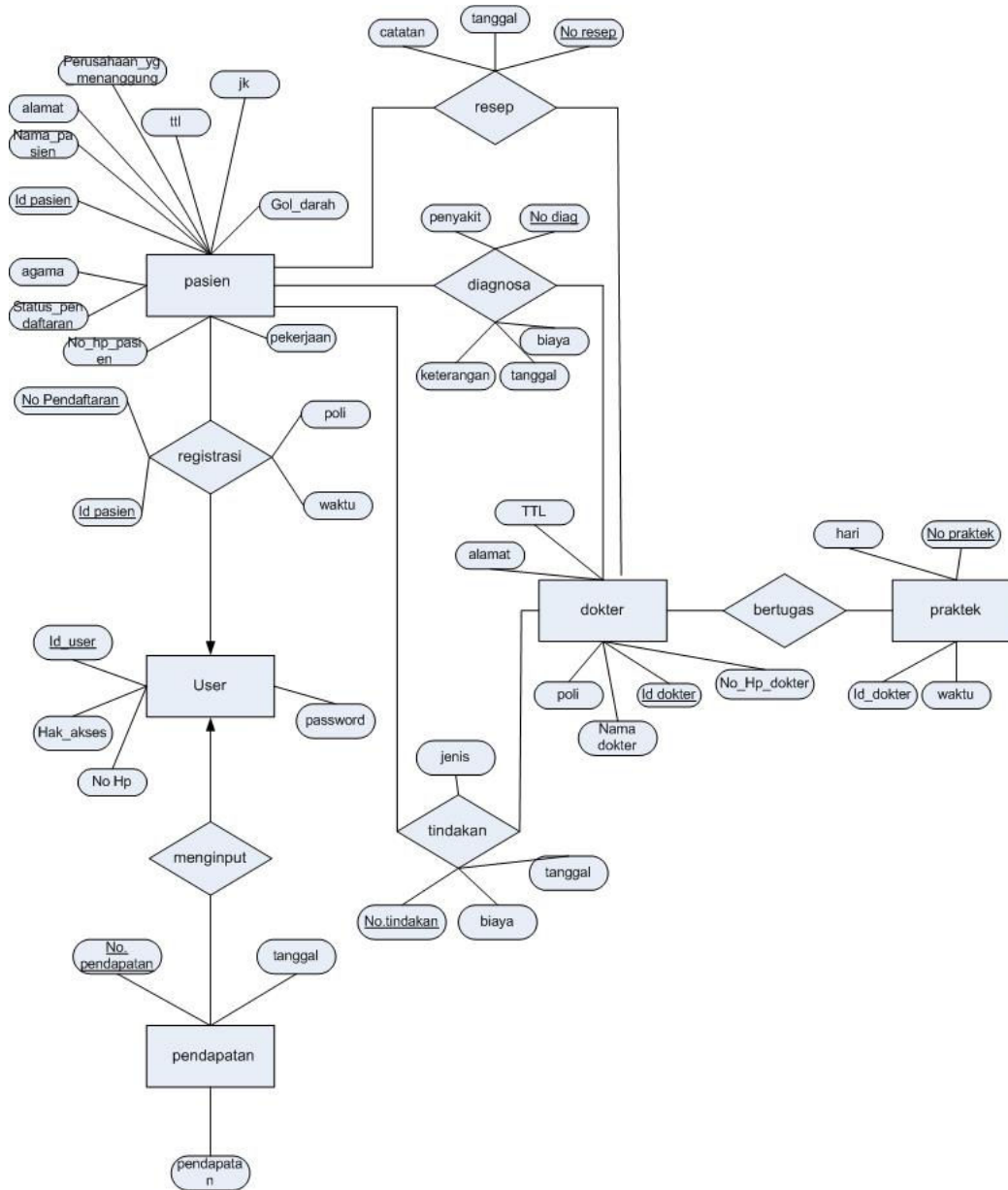
### 3.3.1.4 DFD Level 2 Proses Penerimaan SMS dari Dokter



Gambar 3.6 DFD Level 2 proses Penerimaan SMS dari Dokter

Proses yang terjadi pada DFD level 2 Proses Penerimaan SMS dari Pasien ada 3 proses. Pada proses Validasi Jumlah Pasien menerima input data berupa JP, id\_dokter, poli, waktu dan JP dari User(Dokter) dan data Jumlah Pasien dari t\_registrasi yang selanjutnya akan menghasilkan pesan validasi berupa data jumlah pasien yang mendaftar berobat yang akan dikirim oleh aplikasi ke User. Pada proses Validasi Nama Dokter menerima input data berupa id\_dokter dan ND dari User(Dokter) melalui proses Baca SMS Dokter dan data Nama Dokter dari t\_praktek yang selanjutnya akan menghasilkan pesan validasi berupa Nama Dokter yang bertugas yang akan dikirim oleh aplikasi ke User.

ERD



Gambar 3.7 ERD

Penjelasan ER diagram pada Aplikasi Pendaftaran Pasien Klinik Berbasis SMS ini adalah sebagai berikut:

1. Terdapat 5 entitas yaitu user(admin) (atribut: id\_user, no\_hp\_user, password, hak akses), t\_pasien (atribut: id\_pasien, nama, alamat, ttl, jk, status\_pendaftaran, perusahaan yg menanggung dan no\_hp\_pasien), t\_pendapatan (atribut: no\_pendapatan, tanggal, pendapatan), t\_dokter (atribut: id\_dokter, nama dokter, alamat, specialist, ttl, jk, no\_hp\_dokter ) dan t\_praktek (atribut: no\_praktek, id\_dokter, hari, waktu).
2. Memiliki 6 relationship yaitu registrasi, menginput, diagnosa, tindakan, resep dan bertugas.

## BAB 4 DESKRIPSI PERANCANGAN

### 4.1 Deskripsi Data

Tabel berikut menjelaskan deskripsi data yang digunakan dalam Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM (Studi Kasus: MMC).

Tabel 4.1 Deskripsi Data Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM

No	Nama Data	Tipe Data	Keterangan
1	No handphone User	String	No handphone User( Pasien, Dokter dan Direktur) harus terdaftar di Database klinik
2	Pesan SMS	String	<p>Format sms untuk melakukan pendaftaran :</p> <ul style="list-style-type: none"> <li>• REG#Id_pasien#poli</li> </ul> <p>Contoh : REG# P0001# gigi</p> <p>Format sms untuk melihat jadwal dokter :</p> <ul style="list-style-type: none"> <li>• INFO#ND#Id_pasien</li> </ul> <p>Contoh : INFO#ND#P0001</p> <ul style="list-style-type: none"> <li>• INFO#ND#id_Direktur</li> </ul> <p>Contoh : INFO#ND#U</p> <ul style="list-style-type: none"> <li>• INFO#ND#id_Dokter</li> </ul> <p>Contoh : INFO#ND#D001</p> <p>Format sms untuk melihat jadwal dokter :</p> <ul style="list-style-type: none"> <li>• INFO#JD# Id_pasien</li> </ul> <p>Contoh : INFO#JD#P0001</p> <ul style="list-style-type: none"> <li>• INFO#JD#id_Direktur</li> </ul> <p>Contoh : INFO#JD#Dir</p> <ul style="list-style-type: none"> <li>• INFO#JD#id_Dokter</li> </ul> <p>Contoh : INFO#JD#D001</p> <p>Format sms untuk melihat jumlah pasien :</p> <ul style="list-style-type: none"> <li>• INFO#JP#Id_pasien</li> </ul> <p>Contoh : INFO#JP#P0001</p> <ul style="list-style-type: none"> <li>• INFO#JP#id_Direktur</li> </ul> <p>Contoh : INFO#JP#Dir</p> <ul style="list-style-type: none"> <li>• INFO#JP#id_Dokter</li> </ul> <p>Contoh : INFO#JP#D001</p> <p>Format sms untuk melihat omzet :</p> <ul style="list-style-type: none"> <li>• INFO#JO#id_direktur</li> </ul> <p>Contoh : INFO#JO#Dir</p>

3	Pesan Validasi	String	<p>Format pesan yang dikirim ke user :</p> <ul style="list-style-type: none"> <li>• Jika proses pendaftaran valid : Pesan: No pendaftaran anda adalah :....</li> <li>• Jika proses pendaftaran tidak valid : Pesan: Maaf id anda tidak terdaftar Atau : Maaf format pesan anda salah</li> <li>• Jika proses melihat nama dokter valid : Pesan: dr.leo poli:gigi dr.andi poli:mata dr.nova poli:gigi dr.fendi poli:mata</li> <li>• Jika proses melihat nama dokter tidak valid : Pesan: Maaf id anda tidak terdaftar. Atau : Maaf format pesan anda salah</li> <li>• Jika proses melihat jadwal dokter valid : Pesan : Dr.leo poli : gigi praktek senin 09.00 - 15.00 Dr.andi poli : mata praktek senin 16.00 - 21.00 Dr.nova poli : gigi praktek senin 16.00 - 21.00 Dr.fendi poli : mata praktek senin 09.00 - 15.00</li> <li>• Jika proses melihat jadwal dokter tidak valid: Pesan: Maaf id anda tidak terdaftar. Atau : Maaf format pesan anda salah</li> <li>• Jika proses melihat jumlah pasien valid : Pesan : jumlah pasien yang telah melakukan registrasi pada poli (nama poli) adalah(jmlh_pendaftaran).</li> <li>• Jika proses melihat jumlah pasien tidak valid : Pesan: Maaf id anda tidak terdaftar. Atau : Maaf format pesan anda salah</li> <li>• Jika proses melihat omzet valid : Pesan : Pendapatan Klinik untuk tanggal 2008-05-05 adalah 500000</li> <li>• Jika proses melihat omzet tidak valid : Pesan: Maaf id anda tidak terdaftar. Atau : Maaf format pesan anda salah</li> </ul>
---	----------------	--------	--

## 4.2 Dekomposisi Fungsional Modul

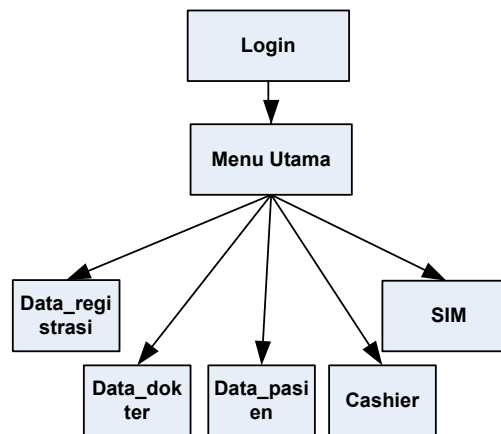
Tabel. 4.2 Input-Proses-Output Aplikasi Pendaftaran Pasien Klinik Berbasis Sms Dengan Integrasi SIM (Studi Kasus: MMC)

No	No. Fungsi	Fungsi/Proses	Tabel Input	Data Input	Tabel Output	Data output	Keterangan
1	F1	Proses Login	t_user	Username, password	-	pesan	Proses masuk ke menu utama
2	F2	cek_sms	-	-	-	-	Proses pengecekan SMS
3	F3.1	Validasi No_Pendaftaran	t_pasien, t_registrasi	id_pasien, poli	t_registrasi	No_Pendaftaran	Melakukan verifikasi format SMS untuk Mendapatkan No.pendaftaran
4	F3.2	Validasi Nama Dokter (ket: untuk pasien )	t_pasien, t_praktek	id_pasien,ND	-	Jadwal Dokter	Melakukan verifikasi format SMS untuk Mendapatkan Nama Dokter
5	F3.3	Validasi Jam Berobat (ket: untuk pasien)	t_pasien, t_praktek	id_pasien,JD	-	Jadwal Berobat	Melakukan verifikasi format SMS untuk Mendapatkan Jadwal Beobat
6	F3.4	Validasi Jumlah Pasien (ket: untuk pasien)	t_pasien, t_registrasi	id_pasien, JP	-	Jumlah Pasien	Melakukan verifikasi format SMS untuk Mendapatkan Jumlah Pasien
7	F4.1	Validasi Omzet (ket: untuk direktur)	t_user, t_pendaftaran	id_direktur, JO	-	omzet	Melakukan verifikasi format SMS untuk Mendapatkan omzet
8	F4.2	Validasi Nama Dokter (ket: untuk direktur)	t_user, t_praktek	id_direktur, ND	-	Jadwal Dokter	Melakukan verifikasi format SMS untuk Mendapatkan Nama Dokter
9	F4.3	Validasi Jam Berobat (ket: untuk direktur)	t_user, t_praktek	id_direktur, JD	-	Jadwal Berobat	Melakukan verifikasi format SMS untuk Mendapatkan Jadwal Beobat
10	F4.4	Validasi Jumlah Pasien (ket: untuk direktur)	t_user, t_registrasi	id_direktur, JP	-	Jumlah Pasien	Melakukan verifikasi format SMS untuk Mendapatkan Jumlah Pasien
11	F5.1	Validasi Jumlah Pasien (ket:	t_dokter, t_registrasi	id_dokter, JP	-	Jumlah Pasien	Melakukan verifikasi format

		untuk dokter)	asi				SMS untuk Mendapatkan Jumlah Pasien
12	F5.2	Validasi Nama Dokter (ket: untuk dokter)	t_dokter, t_praktek	id_dokter, ND	-	Jadwal Dokter	Melakukan verifikasi format SMS untuk Mendapatkan Nama Dokter
13	F5.3	Validasi Jam Berobat (ket: untuk dokter)	t_dokter, t_praktek	id_dokter, JD	-	Jadwal Berobat	Melakukan verifikasi format SMS untuk Mendapatkan Jadwal Beobat

Keterangan fungsional rinci akan dituliskan dilampiran A.

### 4.3 Spesifikasi Kebergantungan Antar Layar



Gambar 4.1 Spesifikasi Kebergantungan Antar Layar

### 4.4 Struktur Menu

Struktur menu aplikasi Pendaftaran Pasien Klinik Berbasis SMS dengan Integrasi SIM (Studi Kasus: MMC) ini adalah sebagai berikut :

Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM (Studi Kasus: MMC)

- Login
- Menu Utama
  - Data\_registrasi
  - Data\_dokter
  - Data\_pasien
  - Cashier
  - SIM

## BAB 5 IMPLEMENTASI DAN PENGUJIAN

### 5.1 Library

Aplikasi Pendaftaran Pasien Klinik Berbasis SMS dengan Integrasi SIM menggunakan library Gammu for Windows yang berfungsi sebagai sms gateway. Gammu berfungsi untuk membaca setiap isi sms yang masuk ke dalam inbox handphone dan menyimpannya secara otomatis ke dalam database Mysql. Setiap sms yang masuk akan di Update setiap saat dan disimpan pada tabel inbox.

File Gammu harus ekstrak pada direktori C:\Win32. Sebelum menjalankan service pada gammu, lakukan konfigurasi pada file *gammurc* berupa nomor port dan koneksi yang digunakan. Untuk mengetahui port mana yang digunakan oleh Hp anda dapat dilihat melalui device manager (catatan : port yang digunakan selalu berubah disetiap komputer yang berbeda), serta koneksi yang digunakan yaitu *at19200*. Lakukan konfigurasi pada file *smsdrc*, Ubah nama file database yang digunakan pada konfigurasi database smsd.

Untuk menjalankannya harus membuka command prompt, kemudian masuk ke `c:\win32>` lalu ketik `gammu -identify`, jika pada layar command prompt menampilkan informasi handphone beserta No IMEI-nya berarti handphone sudah dikenali dengan baik oleh Gammu dan dapat dijalankan. Setelah itu aktifkan sms service dengan mengetikkan `gammu --smsd MYSQL smsdrc`. Perintah ini berfungsi untuk menjalankan tool pada Gammu dan layar command prompt harus tetap dibuka agar fungsinya terus berjalan. Dalam penggunaan tool Gammu ini aplikasi hanya membaca isi sms pada tabel inbox dalam database klinik dan mengelola setiap isi sms yang masuk.

### 5.2 Spesifikasi Kebergantungan Antar Modul

Tabel 5.1 Tabel daftar Modul

No	Nama modul	Fungsi
1	Login	Modul untuk proses login admin
2	Menu_utama	Menampilkan semua informasi hasil pendaftaran pasien melalui SMS dan SIM data pasien
3	Data_registrasi	Modul untuk menampilkan informasi pendaftaran pasien melalui SMS
4	Data_dokter	Modul untuk menampilkan informasi dokter
5	Cashier	Modul untuk menampilkan informasi pendapatan klinik
6	Pengelolaan_SIM	Modul untuk menampilkan Sistem Informasi Manajemen data pasien

### 5.3 Struktur Direktori dan Deskripsi File

Tabel 5.1 Daftar Direktori dan file Aplikasi Pendaftaran Pasien Klinik Berbasis SMS dengan Integrasi SIM (Studi Kasus: MMC)

Nama Direktori	Nama File	Nama Modul	No Fungsi	Keterangan
Form	Login.cs	-	F1	File untuk proses login user
	Menu_utama.cs	-	-	Form untuk menampilkan form Data pasien, Data registrasi, Data_dokter, cashier dan form Pengelolaan_SIM
	Data_registrasi.cs	-	F2, F3.1, F3.2, F3.3, F3.4, F4.1, F4.2, F4.3, F4.4, F5.1, F5.2, F5.3	Menampilkan informasi pendaftaran pasien melalui SMS
	Data_dokter.cs	-	-	Menampilkan form informasi dokter
	Cashier.cs	-	-	Menampilkan form cashier(input pendapatan)
	SIM.cs	-	-	Menampilkan Form aplikasi SISTEM MANAGAMENT DATA PASIEN

Daftar rinci direktori dan file aplikasi ada pada Lampiran C.

### 5.4 Pengujian dan Hasilnya

Implementasi yang dilakukan sesuai pada deskripsi perancangan, untuk rincian pelaksanaan dan hasil pengujian dapat dilihat pada lampiran D.

## **BAB 6 KESIMPULAN DAN SARAN**

### **6.1 Kesimpulan**

Dari proses aplikasi *Pendaftaran Pasien Klinik Berbasis SMS dengan Integrasi SIM (Studi Kasus: MMC)* yang telah dilakukan, maka dapat ditarik kesimpulan bahwa proses pendaftaran berobat dapat dilakukan dengan cara mengirimkan SMS ke Aplikasi yang telah disediakan klinik. User juga dapat mengetahui informasi seputar klinik, seperti : informasi nama dokter klinik, jadwal dokter klinik, jumlah pasien yang berobat dan juga informasi jumlah omzet perhari (khusus direktur klinik). SMS yang dikirim akan diproses oleh aplikasi dan hasil proses akan dikirim langsung dari aplikasi kepada user, baik berupa pesan invalid yaitu proses gagal dilakukan atau pesan validasi, yaitu pesan bahwa proses berhasil dilakukan.

### **6.2 Saran**

Setelah saya menyelesaikan aplikasi *Pendaftaran Pasien Klinik Berbasis SMS dengan Integrasi SIM (Studi Kasus: MMC)*, saya sarankan kepada siapapun yang ingin melakukan pengembangan aplikasi ini agar dapat membuat aplikasi yang dapat menangani proses lain yang ada di klinik seperti : proses yang menangani pengelolaan obat-obatan, report gaji karyawan, menangani pendaftaran pasien dan perusahaan baru untuk didaftar menjadi member klinik melalui SMS dan juga pengembangan system yang dapat menjaga keamanan data pasien dan proses validasi yang lebih terjamin.

## LAMPIRAN A PERANCANGAN RINCI FUNGSIONAL

### A.1 Spesifikasi Fungsi/Proses F1

Identifikasi>Nama : Login  
Deskripsi Isi : id user, password  
Jenis : Form

#### A.1.1 Spesifikasi Layar Utama



The screenshot shows a window titled "Form1" with a gray background. At the top center, the text "Aplikasi Pendaftaran Klinik" is displayed in a large, bold, black font. Below this, there are two text input fields. The first is labeled "ID User" and the second is labeled "Password". Below the input fields, there are two buttons: "login" and "Cancel".

#### A.1.2 Spesifikasi Objek-Objek pada layar

<b>Id_Objek</b>	<b>Jenis</b>	<b>Keterangan</b>
User Name	TextBox	Untuk memasukkan ID USER
Password	TextBox	Untuk memasukkan Password user
Submit	Button	Jika diklik akan masuk ke menu utama jika username dan password valid
Cancel	Button	Jika diklik akan menutup layar login

#### A.1.3 Spesifikasi layar pesan

<b>No</b>	<b>Kasus</b>	<b>Pesan</b>
1.	Belum memasukkan username	Anda belum memasukkan username
2.	Belum Password	Anda belum memasukkan password
3.	Jika username atau password salah	Anda salah memasukkan username dan password
4	Jika Proses valid	<i>Selamat datang di Aplikasi Pendaftaran PasienKlinik Berbasis SMS</i>

## A.1.4 Spesifikasi proses/algorithm

**A.1.4.1. F1.1** : Login  
**Objek terkait** : -  
**Event** : klik

<b>Initial State (IS):- memasukkan id user dan password</b>
<b>Final State (FS) :-Tampil Menu Utama</b> -Tampilkan pesan login sukses atau login gagal
<b>Spesifikasi Proses/algorithm:</b>  <pre>Koneksi open; Input username; Input password; IF(username==true \$\$ password== true){ Tampilkan Menu Utama Tampil messageBox('Selamat Datang di Aplikasi Pendaftaran Pasien Klinik Berbasis SMS'); } }else{ Tampil messageBox(errorMessage); } Koneksi close</pre>

## A.1.5 Spesifikasi Report

*Selamat Datang di Aplikasi Pendaftaran Pasien Klinik Berbasis SMS Dengan Integrasi SIM*

## A.2 Spesifikasi Fungsi/Proses F2

Identifikasi>Nama : cek\_sms  
Deskripsi Isi : pengecekan Format SMS  
Jenis : Proses tanpa layar

### A.2.1 Spesifikasi Layar Utama

*Tidak ada*

### A.2.2 Spesifikasi Objek-Objek pada layar

*Tidak ada*

### A.2.3 Spesifikasi layar pesan

*Tidak Ada*

## A.2.4 Spesifikasi proses/algorithm

**A.2.4.1. F2** : cek\_sms  
**Objek terkait** : -  
**Event** : Mengecek format SMS masuk

Initial State (IS): -Format SMS masuk belum diketahui

Final State (FS) :- Format SMS masuk diketahui

Spesifikasi Proses/algortima:

```
private void cek_sms()
{
    string no_hp, pesan;
    int no_daftar;
    qry = "SELECT SenderNumber,TextDecoded,ID from inbox order by ReceivingDateTime desc limit
1";
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
    //baca data
    System.Data.Odbc.OdbcDataReader reader = OdbcCom.ExecuteReader();
    if (reader.Read())
    {
        no_hp = reader.GetString(0);
        pesan = reader.GetString(1);

        string sep = "#";
        char[] separator = sep.ToCharArray();

        //cek dulu jenis pesanny
        //REG - daftar
        //ND - cek nama dokter
        //JD - cek jam praktek
        //JP - cek jumlah pasien per hari yg telah registrasi
        //JO - jumlah omset per tanggal
        if (pesan.StartsWith("REG"))
        {
            string[] isiPesan = pesan.Split(separator, 3);
            sql = "select count(id_pasien) from t_pasien where id_pasien = '" + isiPesan[1] + "'";
            OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
            //baca data
            reader = OdbcCom.ExecuteReader();
            if (reader.Read())
            {
                = "select count(poli) from t_dokter where poli like '%" + isiPesan[2] + "%'";
                OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
                //baca data
                reader = OdbcCom.ExecuteReader();
                if (reader.Read())
                {
                    if (reader.GetInt32(0) >= 1)
                    {
                        SqlQry = "insert into t_registrasi(id_pasien,poli) value('" + isiPesan[1] + "','" +
isiPesan[2] + "')";
                        //ambil no_pendaftaran

                        //balas pesan
                        message = "No Pendaftaran Anda adalah " + no_daftar + " ";
                    }
                }
            }
            else
            {
                SqlQry = "select count(*) from outbox where DestinationNumber = '" + no_hp + "' and
TextDecoded = '" + message + "' and waktu = now()";
                OdbcCom = new System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
                //baca data
                reader = OdbcCom.ExecuteReader();
            }
        }
    }
}
```

```

        if (reader.Read())
        {
            if (reader.GetInt32(0) == 0)
            {
                sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf poli tidak tersedia di Klinik kami!!!!','true')";
                OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
            }
        }
    }
}
else
{
    SqlQry = "select count(*) from outbox where DestinationNumber = '" + no_hp + "' and
TextDecoded = '" + message + "' and waktu = now()";
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
    //baca data
    reader = OdbcCom.ExecuteReader();
    if (reader.Read())
    {
        if (reader.GetInt32(0) == 0)
        {
            sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf ID anda tidak terdaftar!!!','true')";
            OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
        }
    }
}
}
else if (pesan.StartsWith("ND"))
{
    string[] isiPesan = pesan.Split(separator, 4);
    if (isiPesan[1].StartsWith("P"))
    {
        sql = "select count(id_pasien) from t_pasien where id_pasien = '" + isiPesan[1] + "'";
    }
    else if (isiPesan[1].StartsWith("D"))
    {
        sql = "select count(id_dokter) from t_dokter where id_dokter = '" + isiPesan[1] + "'";
    }
    else if (isiPesan[1].StartsWith("U"))
    {
        sql = "select count(id_user) from t_user where id_user = '" + isiPesan[1] + "' and no_hp_user
= '" + no_hp + "'";
    }
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
    //baca data
    reader = OdbcCom.ExecuteReader();
    if (reader.Read())
    {
        if (reader.GetInt32(0) == 1)
        {
            qry = "select count(poli) from t_dokter where poli like '%" + isiPesan[2] + "%'";
            OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
            //baca data
            reader = OdbcCom.ExecuteReader();
            if (reader.Read())

```

```

        {
            if (reader.GetInt32(0) >= 1)
            {
                {
                    SqlQry = "select t_dokter.nama_dokter from t_dokter,t_praktek where t_praktek.hari
= '" + isiPesan[3] + "' and t_dokter.poli = '" + isiPesan[2] + "' and t_dokter.id_dokter =
t_praktek.id_dokter";
                    OdbcCom = new System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
                    OdbcCom.ExecuteNonQuery();
                    //baca data
                    reader = OdbcCom.ExecuteReader();
                    nama_dokter = "";
                    while (reader.Read())
                    {
                        nama_dokter = "- " + nama_dokter + "\n" + reader.GetString(0);
                    }
                    sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Nama Dokter yang praktek adalah ' +
nama_dokter + "','true)";
                    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                    OdbcCom.ExecuteNonQuery();
                }
            }
            else
            {
                sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf poli tidak tersedia di Klinik kami!!!','true)";
                OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
            }
        }
        else
        {
            sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf ID anda tidak terdaftar!!!','true)";
            OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
        }
    }
}
else if (pesan.StartsWith("JD"))
{
    string[] isiPesan = pesan.Split(separator, 5);
    if (isiPesan[1].StartsWith("P"))
    {
        sql = "select count(id_pasien) from t_pasien where id_pasien = '" + isiPesan[1] + "'";
    }
    else if (isiPesan[1].StartsWith("D"))
    {
        sql = "select count(id_dokter) from t_dokter where id_dokter = '" + isiPesan[1] + "'";
    }
    else if (isiPesan[1].StartsWith("U"))
    {
        sql = "select count(id_user) from t_user where id_user = '" + isiPesan[1] + "' and no_hp_user
= '" + no_hp + "'";
    }
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
    //baca data
    reader = OdbcCom.ExecuteReader();
    if (reader.Read())
    {
        if (reader.GetInt32(0) == 1)
        {
            qry = "select count(poli) from t_dokter where poli like '%" + isiPesan[2] + "%'";
            OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);

```

```

OdbcCom.ExecuteNonQuery();
//baca data
reader = OdbcCom.ExecuteReader();
if (reader.Read())
{
    if (reader.GetInt32(0) >= 1)
    {
        sql = "select count(hari) from t_praktek where hari = '" + isiPesan[3] + "'";
        OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
        //baca data
        reader = OdbcCom.ExecuteReader();
        if (reader.Read())
        {
            if (reader.GetInt32(0) == 1)
            {
                SqlQry = "select t_praktek.waktu from t_praktek,t_dokter where t_dokter.poli =
'" + isiPesan[2] + "' and t_praktek.hari = '" + isiPesan[3] + "' and t_dokter.nama_dokter = '" + isiPesan[4]
+ "' and t_dokter.id_dokter = t_praktek.id_dokter";
                OdbcCom = new System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
                //baca data
                reader = OdbcCom.ExecuteReader();
                while (reader.Read())
                {
                    jam_praktek = reader.GetString(0);
                }
                sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Dokter " + isiPesan[4] + " spesialis " + isiPesan[2]
+ " praktek " + jam_praktek + "','true)";
                OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
            }
            else
            {
                sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf Dokter " + isiPesan[4] + " tidak praktek
pada hari " + isiPesan[3] + "','true)";
                OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
            }
        }
    }
}
else
{
    sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf poli tidak tersedia di Klinik kami!!!','true)";
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
}
}
else
{
    sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf ID anda tidak terdaftar!!!','true)";
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
}
}
}
else if (pesan.StartsWith("JP"))
{
    string[] isiPesan = pesan.Split(separator, 5);
    if (isiPesan[1].StartsWith("P"))

```

```

    {
        sql = "select count(id_pasien) from t_pasien where id_pasien = '" + isiPesan[1] + "'";
    }
    else if (isiPesan[1].StartsWith("D"))
    {
        sql = "select count(id_dokter) from t_dokter where id_dokter = '" + isiPesan[1] + "'";
    }
    else if (isiPesan[1].StartsWith("U"))
    {
        sql = "select count(id_user) from t_user where id_user = '" + isiPesan[1] + "' and no_hp_user
= '" + no_hp + "'";
    }
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
    //baca data
    reader = OdbcCom.ExecuteReader();
    if (reader.Read())
    {
        if (reader.GetInt32(0) == 1)
        {
            qry = "select count(poli) from t_dokter where poli like '%" + isiPesan[2] + "%'";
            OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
            //baca data
            reader = OdbcCom.ExecuteReader();
            if (reader.Read())
            {
                if (reader.GetInt32(0) >= 1)
                {
                    SqlQry = "select count(*) from t_registrasi where poli like '%" + isiPesan[2] + "%' and
waktu like '%" + isiPesan[3] + "%'";
                    OdbcCom = new System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
                    OdbcCom.ExecuteNonQuery();
                    //baca data
                    reader = OdbcCom.ExecuteReader();
                    while (reader.Read())
                    {
                        jumlah = reader.GetInt32(0);
                    }
                    sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','jumlah pasien yang telah melakukan pendaftaran
pada poli '" + isiPesan[2] + "' adalah '" + jumlah + "','true)";
                    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                    OdbcCom.ExecuteNonQuery();
                }
            }
        }
        else
        {
            sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf poli tidak tersedia di Klinik kami!!!','true)";
            OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
        }
    }
    else
    {
        sql = "insert into outbox(Coding,DestinationNumber,TextDecoded,Multipart)
values('Default_No_Compression','" + no_hp + "','Maaf ID anda tidak terdaftar!!!','true)";
        OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
    }
}
}
else if (pesan.StartsWith("JO"))
{
    string[] isiPesan = pesan.Split(separator, 3);
    sql = "select count(id_user) from t_user where id_user = '" + isiPesan[1] + "' and hak_akses =

```

```

'Direktur'';
OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
//baca data
reader = OdbcCom.ExecuteReader();
if (reader.Read())
{
    if (reader.GetInt32(0) >= 1)
    {
        sql = "select pendapatan from t_pendapatan where tanggal = '" + isiPesan[2] + "'";
        OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
        //baca data
        reader = OdbcCom.ExecuteReader();
        if (reader.Read())
        {
            omset = reader.GetInt32(0);
        }
        sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Pendapatan Klinik untuk tanggal " + isiPesan[2] +
" adalah " + omset + "','true)";
        OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
    }
    else
    {
        sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf Anda tidak Berhak melihat pendapatan
klinik !!!','true)";
        OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
    }
}
}
else
{
    //MessageBox.Show("Format Pesan Salah");
    sql = "insert into outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf format pesan anda salah!!!','true)";
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
}
}
//hapus data di inbox n stop ngirim pesan balasan
qry = "Delete from inbox";
OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}

```

## A.2.5 Spesifikasi Report

Tidak ada

## A.3 Spesifikasi Fungsi/Proses F3.1

Identifikasi>Nama : Validasi No\_Pendaftaran  
 Deskripsi Isi : Pesan Validasi No\_Pendaftaran ke Pasien  
 Jenis : Proses tanpa layar

### A.3.1 Spesifikasi Layar Utama

Tidak ada

### A.3.2 Spesifikasi Objek-Objek pada layar

Tidak ada

### A.3.3 Spesifikasi layar pesan

Tidak Ada

### A.3.4 Spesifikasi proses/algoritma

**A.3.4.1. F3.1** : Validasi No\_Pendaftaran

**Objek terkait** : -

**Event** : select

**Initial State (IS):-** REG, id\_pasien, poli

**Final State (FS) :-**no\_pendaftaran  
-simpan data no\_pendaftaran ke tabel t\_registrasi  
-kirim pesan validasi dan invalid

**Spesifikasi Proses/algoritma:**

Open koneksi;

```
if (pesan.StartsWith("REG"))
{
string[] isiPesan = pesan.Split(separator, 3);
select count(id_pasien) from t_pasien where id_pasien = '"' +
isiPesan[1] + '"';

// jika data terbaca
if (reader.Read())
{
if (reader.GetInt32(0) == 1)
{
//jalankan query
qry = "select count(poli) from t_dokter where
poli like '%" + isiPesan[2] + "%'";

//simpan data pendaftaran ke tabel t_registrasi
SqlQry = "insert into t_registrasi
(id_pasien, poli) value('" + isiPesan[1] +
"', '" + isiPesan[2] + "')";

//ambil no_pendaftaran dari t_registrasi
qry = "select MAX(no_pendaftaran) from t_registrasi where
id_pasien = '"' + isiPesan[1] + '"';

//balas pesan ke pasien
SqlQry = "select count(*) from outbox where
DestinationNumber = '" + no_hp + "' and TextDecoded = '"
+ message + "' and waktu = now()";
```

```

    }

else
    {
        SqlQry = "select count(*) from outbox where
        DestinationNumber = '" + no_hp + "' and TextDecoded =
        '" + message + "' and waktu = now()";

        if (reader.GetInt32(0) == 0)
            {
                //Pesan Invalid apabila poli yang di input
                //pasien tidak tersedia di klinik
                sql = "insert into outbox(waktu,Coding,
                DestinationNumber,TextDecoded,Multipart)
                values(now(),'Default_No_Compression','" +
                no_hp + "','Maaf poli tidak tersedia di rumah
                sakit kami!!!','true')";
            }
        }
    }

else
    {
        SqlQry = "select count(*) from outbox where DestinationNumber =
        '" + no_hp + "' and TextDecoded = '" + message + "' and waktu =
        now()";

        if (reader.Read())
            {
                if (reader.GetInt32(0) == 0)
                    {
                        //PESAN INVALID JIKA ID_PASIEN TIDAK TERDAFTAR
                        sql = "insert into outbox(waktu,Coding,
                        DestinationNumber, TextDecoded,Multipart)
                        values(now(),'Default_No_Compression','" + no_hp +
                        "','Maaf ID anda tidak terdaftar!!!','true')";
                    }
            }
    }

close koneksi;

```

#### A.4 Spesifikasi Fungsi/Proses F3.2

Identifikasi>Nama : Validasi Nama Dokter  
 Deskripsi Isi : Pesan validasi ke Pasien  
 Jenis : Proses tanpa layar

#### A.4.1 Spesifikasi Layar Utama

Tidak ada

#### A.4.2 Spesifikasi Objek-Objek pada layar

Tidak ada

#### A.4.3 Spesifikasi layar pesan

Tidak Ada

#### A.4.4 Spesifikasi proses/algorithm

**A.4.4.1. F3.2** : Validasi Nama Dokter

**Objek terkait** : -

**Event** : select

<b>Initial State (IS):-mendapatkan nama dokter</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi;  <pre>if (pesan.StartsWith("ND")) {     string[] isiPesan = pesan.Split(separator, 4);     if (isiPesan[1].StartsWith("P"))     {         sql = "select count(id_pasien) from t_pasien             where id_pasien = '" + isiPesan[1] + "'";     }      OdbcCom.ExecuteNonQuery();     //baca data sms     reader = OdbcCom.ExecuteReader();     if (reader.Read())     {         if (reader.GetInt32(0) == 1)         {             qry = "select count(poli) from t_dokter                 where poli like '%" + isiPesan[2] + "%'";              //baca data             reader = OdbcCom.ExecuteReader();             if (reader.Read())             {                 if (reader.GetInt32(0) &gt;= 1)                 {                     SqlQry = "select t_dokter.nama_dokter from t_dokter,t_praktek where t_praktek.hari = '" + isiPesan[3] + "' and t_dokter.poli = '" + isiPesan[2] + "' and t_dokter.id_dokter = t_praktek.id_dokter";  OdbcCom.ExecuteNonQuery();</pre>

```

//baca data
reader = OdbcCom.ExecuteReader();
nama_dokter = "";

while (reader.Read())
{
nama_dokter = "- " + nama_dokter + "\n" + reader.GetString(0);

//PESAN VALIDASI NAMA DOKTER
}
sql = "insert intooutbox(waktu,Coding,DestinationNumber,
TextDecoded,Multipart) values(now(),'Default_No_Compression','
" + no_hp + "','Nama Dokter yang praktek adalah : " +
nama_dokter + "','true')";

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}
}

else
{
//Pesan Invalid apabila poli yang di input
//USER tidak tersedia di klinik

sql = "insert into outbox(waktu,Coding,DestinationNumber,
TextDecoded, Multipart)values(now(),'Default_No_Compression
',' " + no_hp + "','Maaf poli tidak tersedia di rumah sakit
kami!!!','true')";

}
}

else
{
//Pesan Invalid apabila dokter yang di input
//tidak terdaftar

sql = "insert into outbox(waktu,Coding,DestinationNumber,
TextDecoded,Multipart) values(now(),'Default_No_
Compression','" + no_hp + "','Maaf ID anda tidak
terdaftar!!!','true')";

}
}

else
{
//("PESAN JIKA Format Pesan Salah");
sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf format
pesan anda salah!!!','true')";
OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}
}

koneksi close;

```

## A.4.5 Spesifikasi Report

Tidak ada

## A.5 Spesifikasi Fungsi/Proses F3.3

Identifikasi>Nama : Validasi Jadwal Dokter  
Deskripsi Isi : Pesan validasi ke Pasien  
Jenis : Proses tanpa layar

### A.5.1 Spesifikasi Layar Utama

Tidak ada

### A.5.2 Spesifikasi Objek-Objek pada layar

Tidak ada

### A.5.3 Spesifikasi layar pesan

Tidak Ada

### A.5.4 Spesifikasi proses/algorithm

**A.5.4.1. F3.3** : Validasi Jadwal Dokter  
**Objek terkait** : -  
**Event** : select

<b>Initial State (IS):-mendapatkan jadwal dokter</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi; <pre>if (pesan.StartsWith("JD")) {     string[] isiPesan = pesan.Split(separator, 5);     if (isiPesan[1].StartsWith("P"))     {         sql = "select count(id_pasien) from t_pasien             where id_pasien = '" + isiPesan[1] + "'";     }     OdbcCom.ExecuteNonQuery();     //baca data sms     reader = OdbcCom.ExecuteReader();     if (reader.Read())     {         if (reader.GetInt32(0) == 1)</pre>

```

        {
            qry = "select count(poli) from t_dokter
            where poli like '%" + isiPesan[2] + "%'";

            //baca data
            reader = OdbcCom.ExecuteReader();
            if (reader.Read())
            {
                if (reader.GetInt32(0) >= 1)
                {
                    SqlQry = "select t_praktek.waktu
                    from t_praktek,t_dokter where t_dokter.poli = '" + isiPesan[2] + "'
                    and t_praktek.hari = '" + isiPesan[3] + "' and t_dokter.nama_dokter =
                    '" + isiPesan[4] + "' and t_dokter.id_dokter = t_praktek.id_dokter";

                    OdbcCom.ExecuteNonQuery();

                    //baca data
                    reader = OdbcCom.ExecuteReader();
                    while (reader.Read())
                    {
                        jam_praktek = reader.GetString(0);
                    }

                    //PESAN VALIDASI JADWAL DOKTER / JAM BEROBAT
                    sql = "insert into outbox(waktu,Coding,DestinationNumber,
                    TextDecoded,Multipart) values(now(),'Default_No
                    _Compression','" + no_hp + "','Dokter " + isiPesan[4] + "
                    sepesialis " + isiPesan[2] + " praktek " + jam_praktek +
                    "','true')";
                }

                else
                {
                    //Pesan Invalid apabila poli yang di input
                    //USER tidak tersedia di klinik

                    sql = "insert into outbox(waktu,Coding,DestinationNumber,
                    TextDecoded, Multipart)values(now(),'Default_No_Compression
                    ',' " + no_hp + "','Maaf poli tidak tersedia di rumah sakit
                    kami!!!','true')";
                }
            }

            else
            {
                //Pesan Invalid apabila ID yang di input
                //tidak terdaftar

                sql = "insert into outbox(waktu,Coding,DestinationNumber,
                TextDecoded,Multipart) values(now(),'Default_No_
                Compression','" + no_hp + "','Maaf ID anda tidak
                terdaftar!!!','true')";
            }
        }

        else
        {

```

```

        //("PESAN JIKA Format Pesan Salah");
        sql = "insert into outbox(waktu,Coding,DestinationNumber,
        TextDecoded,Multipart) values(now(),'Default_No_Compression
        ','" + no_hp + "','Maaf format pesan anda salah!!!','true')";

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
    }

koneksi close;

```

### A.5.5 Spesifikasi Report

Tidak ada

## A.6 Spesifikasi Fungsi/Proses F3.4

Identifikasi>Nama : Validasi Jumlah Pasien  
 Deskripsi Isi : Pesan validasi ke Pasien  
 Jenis : Proses tanpa layar

### A.6.1 Spesifikasi Layar Utama

Tidak ada

### A.6.2 Spesifikasi Objek-Objek pada layar

Tidak ada

### A.6.3 Spesifikasi layar pesan

Tidak Ada

### A.6.4 Spesifikasi proses/algorithm

**A.6.4.1. F3.4** : Validasi Jumlah Pasien  
**Objek terkait** : -  
**Event** : select

<b>Initial State (IS):-mendapatkan jumlah pasien yang mendaftar</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi;  <pre> if (pesan.StartsWith("JP"))     {         string[] isiPesan = pesan.Split(separator, 5); </pre>

```

        if (isiPesan[1].StartsWith("P"))
        {
            sql = "select count(id_pasien) from t_pasien
where id_pasien = '" + isiPesan[1] + "'";
        }

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();

//baca data
reader = OdbcCom.ExecuteReader();
        if (reader.Read())
        {
            if (reader.GetInt32(0) == 1)
            {
                qry = "select count(poli) from t_dokter
where poli like '%" + isiPesan[2] + "%'";
                OdbcCom = new
System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
                //baca data
                reader = OdbcCom.ExecuteReader();
                if (reader.Read())
                {
                    if (reader.GetInt32(0) >= 1)
                    {
                        SqlQry = "select count(*) from
t_registrasi where poli = '" + isiPesan[2] + "' and waktu like '%" +
isiPesan[3] + "%'";
                        OdbcCom = new
System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
                        OdbcCom.ExecuteNonQuery();
                        //baca data
                        reader = OdbcCom.ExecuteReader();
                        while (reader.Read())
                        {
                            jumlah = reader.GetInt32(0);
                        }
                        sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','jumlah pasien
yang telah melakukan registrasi pada poli " + isiPesan[2] + " adalah
" + jumlah + "','true')";
                        OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                        OdbcCom.ExecuteNonQuery();
                    }
                }
            }
            else
            {
                sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf poli
tidak tersedia di rumah sakit kami!!!','true')";
                OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
            }
        }
    }
}
else

```

```

        {
//PESAN JIKA ID TIDAK TERDAFTAR
            sql = "insert into
outbox(Coding, DestinationNumber, TextDecoded, Multipart)
values('Default_No_Compression', '"' + no_hp + "', 'Maaf ID anda tidak
terdaftar!!!', 'true')";
            OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
        }
    }
}

else
{
    //("PESAN JIKA Format Pesan Salah");
    sql = "insert into outbox(waktu, Coding, DestinationNumber,
TextDecoded, Multipart) values(now(), 'Default_No_Compression
', '"' + no_hp + "', 'Maaf format pesan anda salah!!!', 'true')";

    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();
}

koneksi close;

```

## A.6.5 Spesifikasi Report

*Tidak ada*

## A.7 Spesifikasi Fungsi/Proses F4.1

Identifikasi>Nama : Validasi Jumlah Omzet  
 Deskripsi Isi : Pesan validasi ke Direktur  
 Jenis : Proses tanpa layar

### A.7.1 Spesifikasi Layar Utama

*Tidak ada*

### A.7.2 Spesifikasi Objek-Objek pada layar

*Tidak ada*

### A.7.3 Spesifikasi layar pesan

*Tidak Ada*

### A.7.4 Spesifikasi proses/algorithm

**A.7.4.1. F4.1** : Validasi Jumlah Omzet

**Objek terkait** : -

**Event**

: select

<b>Initial State (IS):-mendapatkan jumlah omzet</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi;  <pre>if (isiPesan[1].StartsWith("U"))     {  sql = "select count(id_user) from t_user where id_user = '" + isiPesan[1] + "' and no_hp_user = '" + no_hp + "'";     }  OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon); OdbcCom.ExecuteNonQuery();  //baca data sms     reader = OdbcCom.ExecuteReader();     if (reader.Read())     {         if (reader.GetInt32(0) == 1)         {             qry = "select count(pendapatan) from t_pendapatan where tanggal like '%" + isiPesan[2] + "%'";              //baca data             reader = OdbcCom.ExecuteReader();             if (reader.Read())             {                 if (reader.GetInt32(0) &gt;= 1)                 {                     SqlQry = "select t_user.id_user, t_pendapatan.tanggal from t_user t_pendapatan where t_user.id_user = '" + isiPesan[2] + "'";  OdbcCom.ExecuteNonQuery();  //baca data reader = OdbcCom.ExecuteReader(); while (reader.Read())     {         pendapatan = reader.GetString(2);     }  //PESAN VALIDASI JUMLAH OMZET sql = "insert into outbox(waktu,Coding, DestinationNumber, TextDecoded,Multipart) values(now(), 'Default_No _Compression', '" + no_hp + "', 'Total Pendapatan " + isiPesan[2] + ', 'true')";  }  else     {         //Pesan Invalid apabila ID yang di input</pre>

```

//tidak terdaftar

sql = "insert into outbox(waktu,Coding,DestinationNumber,
      TextDecoded,Multipart) values(now(),'Default_No_
      Compression','" + no_hp + "','Maaf ID anda tidak
      terdaftar!!!','true')";

}

}

else
{
    //("PESAN JIKA Format Pesan Salah");
    sql = "insert into outbox(waktu,Coding,DestinationNumber,
          TextDecoded,Multipart) values(now(),'Default_No_Compression
          ','" + no_hp + "','Maaf format pesan anda salah!!!','true')";

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}

koneksi close;

```

## A.7.5 Spesifikasi Report

*Tidak ada*

## A.8 Spesifikasi Fungsi/Proses F4.2

Identifikasi>Nama : Validasi Nama Dokter  
 Deskripsi Isi : Pesan validasi ke Direktur  
 Jenis : Proses tanpa layar

### A.8.1 Spesifikasi Layar Utama

*Tidak ada*

### A.8.2 Spesifikasi Objek-Objek pada layar

*Tidak ada*

### A.8.3 Spesifikasi layar pesan

*Tidak Ada*

### A.8.4 Spesifikasi proses/algoritma

A.8.4.1. F4.2 : Validasi Nama Dokter  
 Objek terkait : -

**Event**

: select

<b>Initial State (IS):-mendapatkan nama dokter</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi;  <pre>if (pesan.StartsWith("ND")) {     string[] isiPesan = pesan.Split(separator, 4);     if (isiPesan[1].StartsWith("U"))     {         sql = "select count(id_user) from t_user         where id_user = '" + isiPesan[1] + "' and         no_hp_user = '" + no_hp + "'";     }      OdbcCom.ExecuteNonQuery();     //baca data sms     reader = OdbcCom.ExecuteReader();     if (reader.Read())     {         if (reader.GetInt32(0) == 1)         {             qry = "select count(poli) from t_dokter             where poli like '%" + isiPesan[2] + "%'";              //baca data             reader = OdbcCom.ExecuteReader();             if (reader.Read())             {                 if (reader.GetInt32(0) &gt;= 1)                 {                     SqlQry = "select                     t_dokter.nama_dokter from t_dokter,t_praktek where t_praktek.hari =                     '" + isiPesan[3] + "' and t_dokter.poli = '" + isiPesan[2] + "' and                     t_dokter.id_dokter = t_praktek.id_dokter";                      OdbcCom.ExecuteNonQuery();                      //baca data                     reader = OdbcCom.ExecuteReader();                     nama_dokter = "";                      while (reader.Read())                     {                         nama_dokter = "- " + nama_dokter + "\n" + reader.GetString(0);                     }                      //PESAN VALIDASI NAMA DOKTER                     sql = "insert intooutbox(waktu,Coding, DestinationNumber,                     TextDecoded,Multipart) values(now(), 'Default_No_Compression',                     '" + no_hp + "', 'Nama Dokter yang praktek adalah : " +                     nama_dokter + "', 'true')";                      OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);                     OdbcCom.ExecuteNonQuery();                 }             }         }     } }</pre>

```

else
{
//Pesan Invalid apabila poli yang di input
//USER tidak tersedia di klinik

sql = "insert into outbox(waktu,Coding,DestinationNumber,
TextDecoded, Multipart) values (now(),'Default_No_Compression
',' " + no_hp + "','Maaf poli tidak tersedia di rumah sakit
kami!!!','true')";

}
}

else
{
//Pesan Invalid apabila dokter yang di input
//tidak terdaftar

sql = "insert into outbox(waktu,Coding,DestinationNumber,
TextDecoded,Multipart) values (now(),'Default_No_
Compression',' " + no_hp + "','Maaf ID anda tidak
terdaftar!!!','true')";

}
}

else
{
//("PESAN JIKA Format Pesan Salah");
sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values (now(),'Default_No_Compression',' " + no_hp + "','Maaf format
pesan anda salah!!!','true')";
OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}

koneksi close;

```

### A.8.5 Spesifikasi Report

*Tidak ada*

### A.9 Spesifikasi Fungsi/Proses F4.3

Identifikasi>Nama : Validasi Jadwal Dokter  
 Deskripsi Isi : Pesan validasi ke Direktur  
 Jenis : Proses tanpa layar

#### A.9.1 Spesifikasi Layar Utama

*Tidak ada*

## A.9.2 Spesifikasi Objek-Objek pada layar

Tidak ada

## A.9.3 Spesifikasi layar pesan

Tidak Ada

## A.9.4 Spesifikasi proses/algoritma

**A.9.4.1. F4.3** : Validasi Jadwal Dokter

**Objek terkait** : -

**Event** : select

<b>Initial State (IS):-mendapatkan jadwal dokter</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algoritma:</b> Open koneksi; <pre>if (pesan.StartsWith("JD")) {     string[] isiPesan = pesan.Split(separator, 5);     if (isiPesan[1].StartsWith("U"))     {         sql = "select count(id_user) from t_user         where id_user = '" + isiPesan[1] + "' and         no_hp_user = '" + no_hp + "'";     }     OdbcCom.ExecuteNonQuery();     //baca data sms     reader = OdbcCom.ExecuteReader();     if (reader.Read())     {         if (reader.GetInt32(0) == 1)         {             qry = "select count(poli) from t_dokter             where poli like '%" + isiPesan[2] + "%'";              //baca data             reader = OdbcCom.ExecuteReader();             if (reader.Read())             {                 if (reader.GetInt32(0) &gt;= 1)                 {                     SqlQry = "select t_praktek.waktu                     from t_praktek,t_dokter where t_dokter.poli = '" + isiPesan[2] + "'                     and t_praktek.hari = '" + isiPesan[3] + "' and t_dokter.nama_dokter =                     '" + isiPesan[4] + "' and t_dokter.id_dokter = t_praktek.id_dokter";                      OdbcCom.ExecuteNonQuery();                      //baca data                     reader = OdbcCom.ExecuteReader();                     while (reader.Read())</pre>

```

    {
        jam_praktek = reader.GetString(0);
    }

//PESAN VALIDASI JADWAL DOKTER / JAM BEROBAT
sql = "insert into outbox(waktu,Coding,DestinationNumber,
    TextDecoded,Multipart) values(now(),'Default_No
    _Compression','" + no_hp + "','Dokter " + isiPesan[4] + "
    sepesialis " + isiPesan[2] + " praktek " + jam_praktek +
    "','true')";
}

    else
    {
        //Pesan Invalid apabila poli yang di input
        //USER tidak tersedia di klinik

        sql = "insert into outbox(waktu,Coding,DestinationNumber,
            TextDecoded, Multipart) values(now(),'Default_No_Compression
            ',' " + no_hp + "','Maaf poli tidak tersedia di rumah sakit
            kami!!!','true')";

    }
}

else
{
    //Pesan Invalid apabila ID yang di input
    //tidak terdaftar

    sql = "insert into outbox(waktu,Coding,DestinationNumber,
        TextDecoded,Multipart) values(now(),'Default_No_
        Compression','" + no_hp + "','Maaf ID anda tidak
        terdaftar!!!','true')";

}

}

else
{
    //("PESAN JIKA Format Pesan Salah");
    sql = "insert into outbox(waktu,Coding,DestinationNumber,
        TextDecoded,Multipart) values(now(),'Default_No_Compression
        ','" + no_hp + "','Maaf format pesan anda salah!!!','true')";

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}

koneksi close;

```

## A.9.5 Spesifikasi Report

Tidak ada

## A.10 Spesifikasi Fungsi/Proses F4.4

Identifikasi>Nama : Validasi Jumlah Pasien  
Deskripsi Isi : Pesan validasi ke Direktur  
Jenis : Proses tanpa layar

### A.10.1 Spesifikasi Layar Utama

Tidak ada

### A.10.2 Spesifikasi Objek-Objek pada layar

Tidak ada

### A.10.3 Spesifikasi layar pesan

Tidak Ada

### A.10.4 Spesifikasi proses/algorithm

**A.10.4.1. F4.4** : Validasi Jumlah Pasien  
**Objek terkait** : -  
**Event** : select

<b>Initial State (IS):-mendapatkan jumlah pasien yang mendaftar</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi;  <pre>if (pesan.StartsWith("JP")) {     string[] isiPesan = pesan.Split(separator, 5);     if (isiPesan[1].StartsWith("U"))     {         sql = "select count(id_user) from t_user where id_user = '" + isiPesan[1] + "' and no_hp_user = '" + no_hp + "'";     }  OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon); OdbcCom.ExecuteNonQuery();  //baca data reader = OdbcCom.ExecuteReader();     if (reader.Read())     {         if (reader.GetInt32(0) == 1)         {             qry = "select count(poli) from t_dokter where poli like '%" + isiPesan[2] + "%'";             OdbcCom = new System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);</pre>

```

        OdbcCom.ExecuteNonQuery();
        //baca data
        reader = OdbcCom.ExecuteReader();
        if (reader.Read())
        {
            if (reader.GetInt32(0) >= 1)
            {
                SqlQry = "select count(*) from
t_registrasi where poli = '" + isiPesan[2] + "' and waktu like '%" +
isiPesan[3] + "%'";
                OdbcCom = new
System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
                //baca data
                reader = OdbcCom.ExecuteReader();
                while (reader.Read())
                {
                    jumlah = reader.GetInt32(0);
                }
                sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','jumlah pasien
yang telah melakukan registrasi pada poli " + isiPesan[2] + " adalah
" + jumlah + "','true')";
                OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                OdbcCom.ExecuteNonQuery();
            }
        }
        else
        {
            sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf poli
tidak tersedia di rumah sakit kami!!!','true')";
            OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
        }
    }
    else
    {
        //("PESAN INVALID JIKA ID TIDAK TERDAFTAR
        sql = "insert into
outbox(Coding,DestinationNumber,TextDecoded,Multipart)
values('Default_No_Compression','" + no_hp + "','Maaf ID anda tidak
terdaftar!!!','true')";
        OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
    }
}
}

else
{
    //("PESAN JIKA Format Pesan Salah");
    sql = "insert into outbox(waktu,Coding,Destin,
TextDecoded,Multipart) values(now(),'Default_No_Compression
','" + no_hp + "','Maaf format pesan anda salah!!!','true')";
}
}
}

```

```
OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);  
OdbcCom.ExecuteNonQuery();  
    }  
  
koneksi close;
```

## **A.10.5 Spesifikasi Report**

*Tidak ada*

## **A.11 Spesifikasi Fungsi/Proses F5.1**

Identifikasi>Nama : Validasi Jumlah Pasien  
Deskripsi Isi : Pesan validasi ke Dokter  
Jenis : Proses tanpa layar

### **A.11.1 Spesifikasi Layar Utama**

*Tidak ada*

### **A.11.2 Spesifikasi Objek-Objek pada layar**

*Tidak ada*

### **A.11.3 Spesifikasi layar pesan**

*Tidak Ada*

### **A.11.4 Spesifikasi proses/algorithm**

**A.11.4.1. F5.1** : Validasi Jumlah Pasien  
**Objek terkait** : -  
**Event** : select

**Initial State (IS):-mendapatkan jumlah pasien yang mendaftar**

**Final State (FS) :-kirim sms validasi atau invalid**

**Spesifikasi Proses/algorithm:**

Open koneksi;

```
if (pesan.StartsWith("JP"))
{
    string[] isiPesan = pesan.Split(separator, 5);
    if (isiPesan[1].StartsWith("D"))
    {
        sql = "select count(id_dokter) from t_dokter
where id_dokter = '" + isiPesan[1] + "'";
    }
    OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
    OdbcCom.ExecuteNonQuery();

    //baca data
    reader = OdbcCom.ExecuteReader();
    if (reader.Read())
    {
        if (reader.GetInt32(0) == 1)
        {
            qry = "select count(poli) from t_dokter
where poli like '%" + isiPesan[2] + "%'";
            OdbcCom = new
System.Data.Odbc.OdbcCommand(this.qry, this.OdbcCon);
            OdbcCom.ExecuteNonQuery();
            //baca data
            reader = OdbcCom.ExecuteReader();
            if (reader.Read())
            {
                if (reader.GetInt32(0) >= 1)
                {
                    SqlQry = "select count(*) from
t_registrasi where poli = '" + isiPesan[2] + "' and waktu like '%" +
isiPesan[3] + "%'";
                    OdbcCom = new
System.Data.Odbc.OdbcCommand(this.SqlQry, this.OdbcCon);
                    OdbcCom.ExecuteNonQuery();
                    //baca data
                    reader = OdbcCom.ExecuteReader();
                    while (reader.Read())
                    {
                        jumlah = reader.GetInt32(0);
                    }
                    sql = "insert into
outbox(waktu,Coding, DestinationNumber, TextDecoded, Multipart)
values(now(), 'Default_No_Compression', '" + no_hp + "', 'jumlah pasien
yang telah melakukan registrasi pada poli " + isiPesan[2] + " adalah
" + jumlah + "', 'true')";
                    OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
                    OdbcCom.ExecuteNonQuery();
                }
            }
            else
            {
                sql = "insert into
outbox(waktu,Coding, DestinationNumber, TextDecoded, Multipart)
```

```

values(now(),'Default_No_Compression','" + no_hp + "', 'Maaf poli
tidak tersedia di rumah sakit kami!!!','true');"
        OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
    }
}
else
{
//PESAN INVALID JIKA ID TIDAK TERDAFTAR
        sql = "insert into
outbox(Coding, DestinationNumber, TextDecoded, Multipart)
values('Default_No_Compression', '" + no_hp + "', 'Maaf ID anda tidak
terdaftar!!!','true');"
        OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
        OdbcCom.ExecuteNonQuery();
    }
}
}

else
{
    //("PESAN JIKA Format Pesan Salah");
    sql = "insert into outbox(waktu,Coding, DestinationNumber,
TextDecoded, Multipart) values(now(), 'Default_No_Compression
', '" + no_hp + "', 'Maaf format pesan anda salah!!!','true');"

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}

koneksi close;

```

### A.11.5 Spesifikasi Report

Tidak ada

### A.12 Spesifikasi Fungsi/Proses F5.2

Identifikasi>Nama : Validasi Nama Dokter  
 Deskripsi Isi : Pesan validasi ke Dokter  
 Jenis : Proses tanpa layar

#### A.12.1 Spesifikasi Layar Utama

Tidak ada

#### A.12.2 Spesifikasi Objek-Objek pada layar

Tidak ada

### A.12.3 Spesifikasi layar pesan

Tidak Ada

### A.12.4 Spesifikasi proses/algorithm

**A.12.4.1. F5.2** : Validasi Nama Dokter  
**Objek terkait** : -  
**Event** : select

<b>Initial State (IS):-mendapatkan nama dokter</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi;  <pre>if (pesan.StartsWith("ND")) {     string[] isiPesan = pesan.Split(separator, 4);     if (isiPesan[1].StartsWith("D"))     {         sql = "select count(id_dokter) from t_dokter where id_dokter = '" + isiPesan[1] + "'";     }     OdbcCom.ExecuteNonQuery();     //baca data sms     reader = OdbcCom.ExecuteReader();     if (reader.Read())     {         if (reader.GetInt32(0) == 1)         {             qry = "select count(poli) from t_dokter where poli like '%" + isiPesan[2] + "%'";              //baca data             reader = OdbcCom.ExecuteReader();             if (reader.Read())             {                 if (reader.GetInt32(0) &gt;= 1)                 {                     SqlQry = "select t_dokter.nama_dokter from t_dokter,t_praktek where t_praktek.hari = '" + isiPesan[3] + "' and t_dokter.poli = '" + isiPesan[2] + "' and t_dokter.id_dokter = t_praktek.id_dokter";                      OdbcCom.ExecuteNonQuery();                      //baca data                     reader = OdbcCom.ExecuteReader();                     nama_dokter = "";                      while (reader.Read())                     {                         nama_dokter = "- " + nama_dokter + "\n" + reader.GetString(0);                     }                 }             }         }     } }</pre>

```

//PESAN VALIDASI NAMA DOKTER
}
sql = "insert intooutbox(waktu,Coding,DestinationNumber,
TextDecoded,Multipart) values(now(),'Default_No_Compression','
" + no_hp + "','Nama Dokter yang praktek adalah : " +
nama_dokter + "','true')";

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}
}

else
{
//Pesan Invalid apabila poli yang di input
//USER tidak tersedia di klinik

sql = "insert into outbox(waktu,Coding,DestinationNumber,
TextDecoded, Multipart) values(now(),'Default_No_Compression
',' " + no_hp + "','Maaf poli tidak tersedia di rumah sakit
kami!!!','true')";

}
}

else
{
//Pesan Invalid apabila dokter yang di input
//tidak terdaftar

sql = "insert into outbox(waktu,Coding,DestinationNumber,
TextDecoded,Multipart) values(now(),'Default_No_
Compression','" + no_hp + "','Maaf ID anda tidak
terdaftar!!!','true')";

}
}

else
{
//("PESAN JIKA Format Pesan Salah");
sql = "insert into
outbox(waktu,Coding,DestinationNumber,TextDecoded,Multipart)
values(now(),'Default_No_Compression','" + no_hp + "','Maaf format
pesan anda salah!!!','true')";
OdbcCom = new
System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}

koneksi close;

```

## A.12.5 Spesifikasi Report

Tidak ada

## A.13 Spesifikasi Fungsi/Proses F5.3

Identifikasi>Nama : Validasi Jadwal Dokter  
Deskripsi Isi : Pesan validasi ke Dokter  
Jenis : Proses tanpa layar

### A.13.1 Spesifikasi Layar Utama

Tidak ada

### A.13.2 Spesifikasi Objek-Objek pada layar

Tidak ada

### A.13.3 Spesifikasi layar pesan

Tidak Ada

### A.13.4 Spesifikasi proses/algorithm

**A.13.4.1. F5.3** : Validasi Jadwal Dokter  
**Objek terkait** : -  
**Event** : select

<b>Initial State (IS):-mendapatkan jadwal dokter</b>
<b>Final State (FS) :-kirim sms validasi atau invalid</b>
<b>Spesifikasi Proses/algorithm:</b> Open koneksi; <pre>if (pesan.StartsWith("JD")) {     string[] isiPesan = pesan.Split(separator, 5);     if (isiPesan[1].StartsWith("D"))     {         sql = "select count(id_dokter) from t_dokter             where id_dokter = '" + isiPesan[1] + "'";         OdbcCom.ExecuteNonQuery();         //baca data sms         reader = OdbcCom.ExecuteReader();         if (reader.Read())         {             if (reader.GetInt32(0) == 1)             {                 qry = "select count(poli) from t_dokter                     where poli like '%" + isiPesan[2] + "%'";                 //baca data                 reader = OdbcCom.ExecuteReader();                 if (reader.Read())                 {                     if (reader.GetInt32(0) &gt;= 1)                     {                         SqlQry = "select t_praktek.waktu                             from t_praktek,t_dokter where t_dokter.poli = '" + isiPesan[2] + "'                             and t_praktek.hari = '" + isiPesan[3] + "' and t_dokter.nama_dokter = "</pre>

```

" + isiPesan[4] + "' and t_dokter.id_dokter = t_praktek.id_dokter";

OdbcCom.ExecuteNonQuery();

//baca data
reader = OdbcCom.ExecuteReader();
while (reader.Read())
{
    jam_praktek = reader.GetString(0);
}

//PESAN VALIDASI JADWAL DOKTER / JAM BEROBAT
sql = "insert into outbox(waktu,Coding,DestinationNumber,
    TextDecoded,Multipart) values(now(),'Default_No
    _Compression','" + no_hp + "','Dokter " + isiPesan[4] + "
    sepesialis " + isiPesan[2] + " praktek " + jam_praktek +
    "','true')";
}

else
{
    //Pesan Invalid apabila poli yang di input
    //USER tidak tersedia di klinik

    sql = "insert into outbox(waktu,Coding,DestinationNumber,
    TextDecoded, Multipart)values(now(),'Default_No_Compression
    ','" + no_hp + "','Maaf poli tidak tersedia di rumah sakit
    kami!!!','true')";

}

else
{
    //Pesan Invalid apabila ID yang di input
    //tidak terdaftar

    sql = "insert into outbox(waktu,Coding,DestinationNumber,
    TextDecoded,Multipart) values(now(),'Default_No_
    Compression','" + no_hp + "','Maaf ID anda tidak
    terdaftar!!!','true')";

}

else
{
    //("PESAN JIKA Format Pesan Salah");
    sql = "insert into outbox(waktu,Coding,DestinationNumber,
    TextDecoded,Multipart) values(now(),'Default_No_Compression
    ','" + no_hp + "','Maaf format pesan anda salah!!!','true')";

OdbcCom = new System.Data.Odbc.OdbcCommand(this.sql, this.OdbcCon);
OdbcCom.ExecuteNonQuery();
}

koneksi close;

```

### **A.13.5 Spesifikasi Report**

*Tidak ada*

## Lampiran B Uraian Rinci Library

### **B.1. Spesifikasi Library Gammu for Windows**

Identifikasi>Nama : Gammu for Windows

Deskripsi Isi : SMS Gateway tool dengan menggunakan koneksi Sony Usb cable data

#### **B.1.1. Spesifikasi Fungsi**

Identifikasi>Nama : gammu -identify

Penggunaan : Mengidentifikasi jenis handphone, no IMEY handphone, koneksi antara Gammu dengan handphone dan database.

#### **B.1.2. Spesifikasi Fungsi <2>**

Identifikasi>Nama : gammu --smsd MYSQL smsdrc

Penggunaan : Berfungsi untuk menjalankan Gammu.

## Lampiran C DAFTAR RINCI FILE DAN DATA

### C.1. Struktur direktori

Objek	Struktur Direktori
Source	D:\TA_sukses\Aplikasi_pendaftaran_klinik
ExeFiles	D:\TA_sukses\Aplikasi_pendaftaran_klinik\Aplikasi_pendaftaran_klinik\bin\Debug
Data	D:\TA_sukses\Aplikasi_pendaftaran_klinik
Dokumentasi	D:\TA_sukses\TA_Iyut Northon

### C.1.2. Direktori pengembangan

Objek	Struktur Direktori
Source	D:\TA_sukses\Aplikasi_pendaftaran_klinik
ExeFiles	D:\TA_sukses\Aplikasi_pendaftaran_klinik\Aplikasi_pendaftaran_klinik\bin\Debug
Data	D:\TA_sukses\Aplikasi_pendaftaran_klinik
Dokumentasi	D:\TA_sukses\TA_Iyut Northon

### C.1.3. Direktori operasional

Objek	Struktur Direktori
Source	D:\TA_sukses\Aplikasi_pendaftaran_klinik
ExeFiles	D:\TA_sukses\Aplikasi_pendaftaran_klinik\Aplikasi_pendaftaran_klinik\bin\Debug
Data	D:\TA_sukses\Aplikasi_pendaftaran_klinik

## C.2. <Isi Direktori 1>

### C.2.1. Isi Subdirektori : D:\TA\_sukses\Aplikasi\_pendaftaran\_klinik

Volume in drive D is Data

Volume Serial Number is 54A3-99E0

Directory of D:\TA\_sukses\Aplikasi\_pendaftaran\_klinik

```
04/08/2008  08:58 AM  <DIR>      .
04/08/2008  08:58 AM  <DIR>      ..
04/08/2008  08:29 AM  <DIR>      Aplikasi_pendaftaran_klinik
04/04/2008  11:37 AM                970 Aplikasi_pendaftaran_klinik.sln
04/08/2008  08:58 AM  <DIR>      Database
04/08/2008  08:56 AM  <DIR>      doc
04/08/2008  08:58 AM  <DIR>      gammu
04/08/2008  08:56 AM  <DIR>      image
04/08/2008  08:56 AM  <DIR>      laporan
04/08/2008  08:56 AM  <DIR>      logbook
04/08/2008  08:56 AM  <DIR>      slide

          1 File(s)                970 bytes
        10 Dir(s)  14,102,323,200 bytes free
```

## C.3. File Instalasi

## LAMPIRAN D DOKUMEN RINCI TESTING

### A.1 Tim Penguji

1. Pak Riwinoto (RW)
2. M.Irwan(MI)

### A.2 Hasil Rinci Pengujian

No .	No. Fungsi	Deskripsi Fungsional	Kelompok Uji	Prosedur & Kasus uji	Hasil yang diharap	Hasil Tester	Tes ter	Tanggal testing	Ket
1	F1	Proses Login	Normal	Masukkan Username dan Password valid	Login Sukses, Tampil Menu Utama	diterima	RW MI	03-04- 2008	
			Tidak normal	Masukkan Username atau Password salah	Pesan Invalid, Menu Utama tidak tampil	diterima	RW MI		
2	F2	Cek_sms	Normal	Pengecekan format sms user	Pengecekan format sms user diketahui	diterima	RW MI	03-04- 2008	
			Tidak normal	Pengecekan format sms user gagal	Pengecekan format sms user tidak diketahui	diterima	RW MI		
4	F3.1	Validasi Pendaftaran	Normal	Id_pasien dan format sms benar	Kirim pesan validasi ke pasien	diterima	RW MI	03-04- 2008	
			Tidak normal	Id_pasien dan format sms salah	Kirim pesan invalid	diterima	RW MI		
5	F3.2	Validasi Nama Dokter (ket: untuk pasien )	Normal	Id_pasien dan format sms benar	Kirim pesan validasi ke pasien	diterima	RW MI	03-04- 2008	
			Tidak normal	Id_pasien dan format sms salah	Kirim pesan invalid	diterima	RW MI		
6	F3.3	Validasi Jadwal Dokter (ket: untuk pasien	Normal	Id_pasien dan format sms benar	Kirim pesan validasi ke pasien	diterima	RW MI	03-04- 2008	
			Tidak normal	Id_pasien dan format sms salah	Kirim pesan invalid	diterima	RW MI		
7	F3.4	Validasi Jumlah Pasien (ket: untuk pasien)	Normal	Id_pasien dan format sms benar	Kirim pesan validasi ke pasien	diterima	RW MI	03-04- 2008	
			Tidak normal	Id_pasien dan format sms salah	Kirim pesan invalid	diterima	RW MI		

8	F4.1	Validasi Omzet (ket: untuk direktur)	Normal	Id_user(direktur) dan format sms benar	Kirim pesan validasi ke pasien	Tidak diterima	RW MI	03-04- 2008	
			Tidak normal	Id_user(direktur) dan format sms salah	Kirim pesan invalid	Tidak diterima	RW MI		
9	F4.2	Validasi Nama Dokter (ket: untuk direktur)	Normal	Id_user(direktur) dan format sms benar	Kirim pesan validasi ke pasien	Diterima	RW MI	03-04- 2008	
			Tidak normal	Id_user(direktur) dan format sms salah	Kirim pesan invalid	Diterima	RW MI		
10	F4.3	Validasi Jadwal Dokter (ket: untuk direktur)	Normal	Id_user(direktur) dan format sms benar	Kirim pesan validasi ke pasien	Diterima	RW MI	03-04- 2008	
			Tidak normal	Id_user(direktur) dan format sms salah	Kirim pesan invalid	Diterima	RW MI		
11	F4.4	Validasi Jumlah Pasien (ket: untuk direktur)	Normal	Id_user(direktur) dan format sms benar	Kirim pesan validasi ke pasien	Diterima	RW MI	03-04- 2008	
			Tidak normal	Id_user(direktur) dan format sms salah	Kirim pesan invalid	Diterima	RW MI		
12	F5.1	Validasi Jumlah Pasien (ket: untuk dokter)	Normal	Id_dokter dan format sms benar	Kirim pesan validasi ke pasien	Diterima	RW MI	03-04- 2008	
			Tidak normal	Id_dokter dan format sms salah	Kirim pesan invalid	Diterima	RW MI		
13	F5.2	Validasi Nama Dokter (ket: untuk dokter)	Normal	Id_dokter dan format sms benar	Kirim pesan validasi ke pasien	Diterima	RW MI	03-04- 2008	
			Tidak normal	Id_dokter dan format sms salah	Kirim pesan invalid	Diterima	RW MI		

14	F5.3	Validasi Jadwal Dokter (ket: untuk dokter)	Normal Tidak normal	Id_dokter dan format sms benar Id_dokter dan format sms salah	Kirim pesan validasi ke pasien Kirim pesan invalid	Diterima Diterima	RW MI RW MI	03-04- 2008	
----	------	--	------------------------	--	---	----------------------	----------------------	-------------	--



## LAMPIRAN F LOGBOOK

<b>Minggu</b>	<b>Periode</b>	<b>Ada/ Tidak Ada</b>
1 - 2	22 Februari s.d. 29 Februari 2008	Ada
3 - 6	7,14,21,28 Maret 2008	Ada
7	4 April 2008	Ada
8	7 April s.d 11 April 2008	Ada
9 – 10	18 April s.d 25 April 2008	Ada
11	28 April s.d 2 Mei 2008	Ada
12 – 13	5 Mei s.d 16 Mei 2008	Ada
14	19 Mei s.d 23 Mei 2008	Ada

## LAMPIRAN G DAFTAR PUSTAKA

1. Eric Gunnerson: A programmer's introduction to c#, aPress, Desember 2007
2. Kadir, Abdul: Pemrograman C++, Andi. Yogyakarta, Desember 2007
3. Lagod, Junald. MAGIC PAYROLL VERSION 1.5, [www.planet-sourcecode.com](http://www.planet-sourcecode.com), Desember 2007
4. Shekhar, Raj: Login System in C# : [www.planet-source-code.com](http://www.planet-source-code.com), Desember 2007
5. <http://www.muhadkly.net/sms-gateway-menggunakan-gammu>